Development of Network Architecture and Protocol Stack for Mobile Wireless Sensor Networks

THESIS

Submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

SREEJITH V.

Under the Supervision of

Prof. K. R. ANUPAMA

and

Co-supervision of

Dr. LUCY J. GUDINO

BITS Pilani
Pilani|Dubai|Goa|Hyderabad

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

2016

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

Certificate

This is to certify that the thesis entitled **'Development of Network Architecture and Protocol Stack for Mobile Wireless Sensor Networks'** and submitted by **SREEJITH V.**, ID.No. **2011PHXF021G** for award of Ph.D. of the Institute embodies original work done by him under our supervision.

Signature of the Supervisor : K.R. Anupama
Name in capital letters : **Prof. K. R. ANUPAMA**
Designation : **PROFESSOR**
Date : 27/12/2016

Signature of the Co-supervisor :
Name in capital letters : **Dr. LUCY J. GUDINO**
Designation : **ASSISTANT PROFESSOR**
Date : 27/12/2016

Declaration

I, Sreejith V., declare that this thesis titled, ‘Development of Network Architecture and Protocol Stack for Mobile Wireless Sensor Networks’ submitted by me under the supervision of Prof. K. R. Anupama and Dr. Lucy J. Gudino is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for award of any degree.

Signature of the student :

Name of the student : **SREEJITH V.**

ID number of the student : **2011PHXF021G**

Date :

Abstract

Recent technological advances led to the development of low-cost, low-power small size devices. One such device is sensor nodes, which form Wireless Sensor Network (WSN) which are capable of sensing, computation and communication. There exist many application scenarios, ranging from monitoring to military applications, where mobility is also required. In mobile WSN, the mobility is classified as controlled and uncontrolled (random) mobility. In case of controlled mobility, introducing mobility can enhance the performance of existing WSN in terms of data rate, reliability and energy efficiency. This thesis aims to improve the services and protocols of both controlled and uncontrolled mobility based WSN. In controlled mobility, thesis aims to improve the performance of services such as event monitoring, area monitoring and improving the performance of existing terrestrial network using mobile sensor nodes. Network architecture and algorithms are proposed for controlled mobility to improve sensing resolution, network coverage and connectivity.

For event monitoring applications, incorrectly captured data from sensor nodes can lead to false alarms, which are equally undesirable as a missing event. So, we considered the use of mobile WSN equipped with multimedia sensors for event monitoring. In our proposed architecture, these mobile nodes are arranged between event location and the base-station. The thesis focuses on optimal selection with minimum number of mobile nodes to establish a communication between the event and the base-station in minimum time. Our proposed method uses Hungarian algorithm to minimize the maximum time taken for the simultaneous movement of mobile nodes from their current position. Further, we extend our approach to address multi-event monitoring using Minimum Steiner Tree (MST).

Area monitoring applications in WSN are used for a multitude of applications ranging from data collection to network maintenance. In this thesis, we proposed deterministic (Max-Gain) and non-deterministic (Hybrid) algorithms for area monitoring application using mobile nodes. Max-Gain algorithm moves mobile nodes to a distant unexplored position rather than directing them to a near frontier, while Hybrid approach uses the principles of random direction mobility model and frontier algorithm to monitor a given area. The algorithms proposed for area monitoring and multi-event connectivity were used to discover partitions in a network and to restore connectivity.

Mobile WSN is used to improve the QoS of the existing terrestrial WSN. A multipath node disjoint routing algorithm is proposed in order to establish multiple paths from source to base-station with the help of static and mobile nodes. For interference free communication, a frequency hopping based multi-channel MAC protocol has also been proposed.

This thesis also looks at the need for energy efficient communication protocols for mobile WSN. We discuss different scenarios of routing depending on the mobility of an element such as source, relay node and sink. We proposed dynamic Steiner tree based routing algorithm with the support of Elastic routing for different scenarios when sink is mobile. An area based opportunistic routing algorithm that predicts the life time of links with neighbors is also proposed for routing via mobile relays. A schedule based MAC protocol that uses dynamic TDMA is presented in this thesis. This protocol is designed for a mixed scenario consisting of mobile and static nodes.

We used both simulations and test-bed for demonstrating the effectiveness of the proposed protocols. Simulations were done using Castalia, an OMNeT++ platform which has got a realistic wireless channel and radio models. The test-bed was implemented using Berkely motes. A custom designed mobile robot named *B-Bot* embedded with TelosB mote was used as mobile bot.

Simulation results indicate that our proposed algorithms for event monitoring and area monitoring perform better compared to similar methods. Similarly, simulation results indicate that the proposed routing and MAC protocol performs better compared to similar approaches. Results of simulation have been studied in terms of parameters of network performance such as Packet Delivery Ratio(PDR), throughput, control overhead etc. Feature of energy efficiency of the proposed protocol stack has been demonstrated using the simulation tools. Proof of concept of event monitoring service application with the proposed routing and MAC were verified using our test-bed. The results confirm the efficacy of our proposed algorithms.

Acknowledgements

It is my pleasure to acknowledge several individuals who has played key roles for the completion of my Ph.D thesis.

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof.K.R.Anupama, for her constant guidance and motivation in doing my research. She has extended all necessary facilities and help right from the commencement to the conclusion of this research work. Without her constant support and inspiration, this thesis would not have been possible.

I take this opportunity to express my deep and profound gratitude to my co-supervisor, Dr. Lucy J.Gudino, for her guidance and support. Her valuable guidance and support in completing my thesis is appreciable.

My sincere thanks to members of the Doctoral Advisory Committee, Prof. Bharat Deshpande and Prof. Neena Goveas, for their critical suggestions and comments.

I express my thanks to Prof. Souvik Bhattacharyya, Vice-Chancellor, BITS-Pilani, Prof. G Raghurama, Director, BITS-Pilani, K K Birla Goa Campus, Prof. Sasikumar Punnekkat, former Director, BITS-Pilani, K K Birla Goa Campus, Prof. S. K. Verma, Dean, Academic Research Division, BITS-Pilani, Prof. P. K. Das, Associate Dean, Academic Research Division, BITS-Pilani K K Birla Goa Campus and members of the Doctoral Research Committee of Dept. of CS&IS, BITS-Pilani, K K Birla Goa Campus for providing administrative support and facilities to carry out my research at the institute.

With great pleasure, I express my appreciation to my co-authors, Mr. Suriyadeepan R., Mr. Prateek Khandelwal, and Mr. Vaibhav Gupta. I thank all of them for helping me conduct numerous experiments on hardware testbed in laboratory as well as on simulation platforms.

I express sincere thanks to my colleagues Dr. Biju Raveendran, Dr. Baskar A., Dr. Anoop Sasikumar, Dr Rayson Alex, Mr. Bijil Prakash and Mr. Shamanth N., for their immense help, constructive comments, discussions and moral support. I thank Dr. Shalu M A, Dept. of Mathematics, IIITDM, Kancheepuram, Chennai for providing help at different levels of the work.

I thank all my teachers, colleagues, friends and relatives who have helped me either directly or indirectly at different stages of my research work.

I remain indebted to my parents, A. Vidhyadharan Nair and Usha Kumari Amma, for their blessings, moral support and motivation.

From the bottom of my heart, I thank my wife, Meera and my son Gautham for their kind co-operation, patience and understanding throughout the course of this work.

I am thankful to God Almighty for bestowing his blessings on me.

SREEJITH V.

Dedicated To My Parents . . .

Contents

List of Figures

List of Tables

Abbreviations

WSN	**W**ireless **S**ensor **N**etwork
MEMS	**M**icro-**E**lectro-**M**echanical **S**ystems
PDR	**P**acket **D**elivery **R**atio
TDMA	**T**ime **D**ivision **M**ultiple **A**ccess
GPS	**G**lobal **P**ositioning **S**ystems
RPGM	**R**eference **P**oint **G**roup **M**odel
OTA	**O**ver-**T**he-**A**ir Programming
MST	**M**inimum **S**panning **T**ree
SMT	**S**teiner **M**inimum **T**ree
VFF	**V**irtual **F**orce **F**ield
CV	**C**overage **V**alue
PI	**P**artition **I**nterface
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
CH	**C**luster **H**ead
FDM	**F**requency **D**ivision **M**ultiplexing
TPSN	**T**ime synchronization **P**rotocol for **S**ensor **N**etwork
CSMA	**C**arrier **S**ense **M**ultiple **A**ccess
REAR	**R**eliable **E**nergy **A**ware **R**outing
EECA	**E**nergy **E**fficient **C**ollision **A**ware
UART	**U**niversal **A**synchronous **R**eceiver/**T**ransmitter
OR	**O**pportunistic **R**outing
TTDD	**T**wo-**T**ier **D**ata **D**issemination
LBDD	**L**ine **B**ased **D**ata **D**issemination
EG	**E**xpect **G**rids

EZ	**E**xpect **Z**one
TR	**T**raffic **R**e-director
SL	**S**ink **L**ocation
GPSR	**G**reedy **P**erimeter **S**tateless **R**outing
H-MAC	**H**ybrid **M**edium **A**ccess **C**ontrol
LEACH	**L**ow-**E**nergy **A**daptive **C**lustering **H**ierarchy
AODV	**A**d-Hoc **O**n-demand **D**istance **V**ector
PWM	**P**ulse **W**idth **M**odulation
PAN	**P**ersonal **A**rea **N**etworks

Chapter 1

Introduction

With the recent developments in embedded systems and communication devices, the design and development of low-cost, low-power, multi-functional devices that are small in size have become both technologically and commercially feasible and their applications are many fold. Wireless network offers the flexibility in terms of design and applications. Advances in Micro-Electro-Mechanical Systems (MEMS) technology has triggered the evolution of low-power wireless sensor nodes [1]. The Wireless Sensor Network is a dominant, flexible and scalable paradigm which consists of a large number of sensing nodes, that can gather information from the environment, and coordinate with each other in order to relay the information to a central base-station. These sensor nodes can sense, process and relay data to the base-station by setting up a collaborative network. The concept of WSN was first proposed in 1999 by Joseph et al., where sensor nodes were deployed anywhere in the environment to collaboratively solve the problem [2].

A WSN consists of many sensor nodes that communicate over wireless links. A variety of thermal, mechanical, magnetic and optical sensors may be attached to the sensor node to measure environmental conditions. These sensors can sense and measure the information from the environment and transfer the data to the microcontroller [3]. Sensor nodes have limited memory and are generally deployed in a remote location where they sense, collect, process and transmit data to the base-station. Sensor nodes are mainly powered by battery units. In some cases, secondary power sources such as solar panels may be added to increase the life time of a node [4]. The type of processor, radio and sensors used in the network depends on the application.

WSNs are used in several applications [5] [6] such as surveillance, target tracking, structural monitoring, etc. It can also be used for forecasting natural disasters by sensing seismic events, or by monitoring volcanic activity before they occur. Applications of WSN are also being increasingly used in the health care system for patient monitoring, where the body vitals are sensed using sensor nodes and are communicated to doctors who may be in a remote location.

Jennifer et al., classifies WSNs based on the application and deployment scenarios [7]. Depending on the environment, a sensor network faces different challenges and constraints. WSNs are classified as

Terrestrial WSN: Terrestrial WSN consists of sensor nodes deployed in structured or unstructured pattern, on the ground. In this network, energy conservation is a major challenge. This is addressed either by the use of low duty cycle operations, optimal communication protocols or by reducing the number of communication overhead. The sensors used for such WSNs are usually less complex.

Multimedia WSN: It consists of sensor nodes embedded with acoustic sensors, cameras or both. Monitoring or tracking is done using images, audio or video [8]. Multimedia sensor nodes can be a part of terrestrial WSNs depending on the type of applications [9]. The major challenges in multimedia WSN are the processing and high bandwidth requirements by low profile nodes in an energy and processing power constrained environment.

Underwater WSN: In underwater WSN, sensor nodes are deployed on the seabed and along the ocean column. Autonomous Underwater Vehicles (AUVs) may be used for gathering data. The underwater sensor nodes use acoustic links to communicate with surface station. Acoustic links have high latency, hence protocols developed for Radio Frequency(RF) based systems cannot be used without major modifications. The protocol development for underwater WSN should also be energy efficient, as the battery cannot be recharged or replaced [10].

Underground WSN: Sensor nodes in underground WSNs are buried or deployed in caves or mines. Underground WSN incurs high cost in terms of equipment, deployment and maintenance. Once deployed, accessibility will not be easy. Hence, the protocol designed for such a scenario should be energy aware and fault tolerant [11]. The high attenuation and signal loss, make the design of communication protocol in underground WSNs challenging.

Mobile WSN : In this category, the network consists of sensor nodes that have the ability to move within the network. Mobile sensor nodes can be a part of other types of WSN also. For example, the mobile nodes can be used to improve data gathering in terrestrial or in multimedia WSN [12] [13] [14]. Mobile multimedia sensor network consists of node with mobile and multimedia functionality which can be controlled by contextual information collected by other systems to enable interactive multimedia services [15] [16]. Yu-chee et al., proposed a mobile surveillance system called iMouse for detecting and analysing unusual events [17]. John and Mehmet proposed an architecture consisting of mobile nodes that harvest data from stationary underground nodes [18].

Mobility can be achieved by interfacing a node with robotic vehicle or attaching it to a moving object like vehicle, animal and humans [19]. In certain scenarios, sensor node may move due to the environment in which they are placed [20]. Mobile WSNs have better and improved coverage, better energy efficiency, superior channel capacity, etc., when compared to their static counterparts [21]. This thesis presents the network architecture and protocol stack for mobile WSN.

Mobility in WSNs can also be classified according to the type of movement of the moving entity. Based on the mobility pattern, mobility in WSN can be classified as controlled or uncontrolled (random) mobility [22]. In controlled mobility, mobile nodes are programmed to move in a pre-determined pattern. In certain applications, mobility is deliberately introduced in order to improve coverage and connectivity and hence to improve the network performance. In random mobility, sensor node movement cannot be pre-determined or controlled.

1.1 Applications of Mobile WSNs

In this section, the application areas where mobile WSN can be used are presented. The applications presented here are based on the mobility aspect of mobile WSN which can be either controlled or random mobility. Mobility can also be passive as in the case where mobile nodes move with wind or water. Mobile WSN with the controlled mobility can be used in the following applications:

- To improve the network services [23] [24]: Applications such as network maintenance, mobile anchor node based static node localization and area exploration are some examples of network services.
- To detect and repair the coverage holes in the network [25] [26]: If the deployed area is remote or hostile, manual intervention in the sensor field is difficult. In such a scenario, mobile nodes can be used to add connections and restore network connectivity [27] [28]. In several WSN network applications, it may not be possible to determine an optimal deployment pattern until the nodes start sensing and transmitting data towards the base-station. Redeployment in such situation is not feasible in remote areas. This problem can be resolved by the use of mobile nodes, which can reorganize themselves as per the ideal deployment pattern. It is shown by Basagni et al., that integrating mobile nodes to the existing sensor network improves the coverage and connectivity and hence the performance [29].
- Partition discovery and recovery: In terrestrial WSN, when nodes are deployed randomly, certain parts of the target area may not have enough density of sensor nodes. This leads to network partitioning. Partitions may also occur due to reasons such as software bugs , atmospheric corrosion of sensor nodes, poor protocol design, battery exhaustion etc., [30] [31] [32].
- Adaptable sensing applications [33]: Mobile nodes can be used to monitor the sensing field depending upon the position of the event or to follow a moving target. Controlled mobility can also be used for coordinating other mobile and static sensor nodes to provide updated information to the base-station. In applications

such as forest fire monitoring; mobile nodes can maintain a safe distance to monitor and update the information regarding an event to the remote control station.

- Other applications: It includes data muling, Over The Air (OTA) programming, etc., [23] [7]. OTA refers to various methods of distributing new softwares,configuration etc., to devices via secure wireless communication. For WSN, OTA is applies to sensor nodes via Zigbee or IEEE 802.15.4. If the deployment area is large, mobile nodes can be used to update the static nodes.

Mobile WSN with the random mobility can be used in the following applications:

- Monitoring and tracking: Sensors that are attached to animals, humans and vehicles introduce mobility into WSNs. Applications where random mobility is inherent are wildlife monitoring, underwater networks and body area networks etc. In the case of wildlife monitoring, sensor nodes are placed on the body of the animal. The sensor nodes gather the data and transmit it to a base-station via static nodes or via other mobile nodes [34].
- Military applications: Nodes may be mounted on vehicles or worn by soldiers [35]. In this scenario, the sink and the relay node themselves may be mobile.
- Patient-monitoring: Vital signals collected from the patient need to be communicated to a remote doctor.
- Mobile base-station: The nodes in the neighborhood of static base-station will die earlier due to being on multiple paths to the base-station. Mobile base-station is used in terrestrial WSN to prolong the life time of the network [19]. In applications where data mules are used, the network life time is extended by reducing the number of hops for data communication to the base-station. In real life applications, these mobile base-stations can be carried by animal, human or by vehicle and hence follow a random mobility model.

1.2 Advantages and Challenges of Mobile WSN

In many application scenarios, a mobile WSN out performs static WSN [36], [37]. Some of the advantages of using mobile nodes with existing static WSN are listed below.

- In case of static sensor network; for better quality sensing, a dense deployment is preferred. This results in increase in the cost of the network. To overcome this, a sparse deployment can be achieved with the help of mobile WSN.
- Mobile nodes can rearrange themselves for better sensing and targeted monitoring.
- Mobile nodes can relocate themselves for improving connectivity and coverage of a sparse WSN.
- Mobility of sensor nodes can be exploited to reduce the energy consumption in a multi-hop network by using them as relays or as data mules. Mobile nodes can be used for increasing the network life time via mobile sinks(mobile base-station)

While mobile WSNs have several advantages, introducing mobility is challenging in terms of network protocol stack design. Conventional routing protocols use routing table to route the data to the destination. Frequent routing table update is inevitable in case of mobile WSN due to dynamic topology. The network needs to perform route discovery frequently while optimizing energy consumption, time and bandwidth. Dynamic topology also leads to frequent link breakages, thereby affecting the on going communication. Other issues include channel fading, shadowing and interference, etc.

In a static sensor network, the node position is fixed while in a mobile node, the location information needs to be frequently updated. This frequent localization is a challenge for low-energy devices in-terms of energy consumption and the unavailability of rapid localization services.

Communication in mobile as well as static WSNs consumes the maximum energy [7]. Mobile WSN needs frequent neighbor discovery and fast localization, hence the amount of control overhead is high, as compared to static WSNs. Moreover, mobile WSNs need additional power for mobility, and hence are often equipped with additional power sources [4].

1.3 Motivation

There are several protocols available in the literature [7] which addresses the issues pertaining to static WSN such as localization, connectivity, coverage, communication protocols, etc. These protocols cannot be applied directly to mobile WSN. Applications using mobile WSN have application-specific characteristics and requirements. These application-specific characteristics and requirements coupled with today's technology lead to the development of robotic vehicle and also in the design of new algorithms and communication protocols.

Recent advancement in distributed robotics and low-power embedded systems have allowed designers to come up with low cost wireless robots [38]. These robots can be programmed to move in the sensor field to rearrange themselves for improving the network performance. Introducing mobility in WSN helps to improve connectivity, coverage [39], reliability [40], reduction in deployment cost [41], energy efficiency [42] [43], etc. Many protocols are proposed in literature that employ mobile nodes for improving the network performance. Although the performance of these protocols is promising in terms of energy efficiency, further research would be needed to address issues such as QoS, latency and throughput for real-time applications.

In time critical application such as event monitoring application, the need for sensing the entire area with stationary nodes may not be efficient and economical. Also, such application requires special sensors such as multimedia sensors, using it in all static node seems unrealistic. The alternate way is to monitor the area with several static sensors and few mobile sensors that can collaborate in order to improve the network performance to detect or to perform certain action as fast as possible.

In time critical application such as event monitoring application, the need for sensing the entire area with stationary nodes may not be efficient and economical. Also, such application requires special sensors such as multimedia sensors, using it in all static node seems unrealistic. The alternate way is to monitor the area with several static sensors and few mobile sensors that can collaborate in order to improve the network performance to detect or to perform certain action as fast as possible.

Many routing protocols have been proposed in literature for terrestrial and multimedia sensor network [23]. An important consideration for these routing protocols are energy efficiency and traffic flow [7]. Two main categories of routing protocols are location based and cluster based. Node location is used for routing data in the location based routing, where as in cluster based routing, cluster head is used to collect the data and then to route it to the base-station. Applications such as event tracking using mobile nodes may use media sensors, which generate high bandwidth data needs special attention while designing the routing protocol, to address the issues such as improving QoS and energy efficiency. Also, the protocols designed should ensure the interference free communication in-order to enhance the data rate. There is little research done in QoS routing [23]. The new communication protocol needs to be designed which can handle high-bandwidth data while ensuring QoS such as an end to end delay while providing security.

In random mobility, depending on the application, the mobility can be at different levels. A source(s) can be mobile, sink node can be mobile or relay node(s) can be mobile. Frequent topological changes result in link failure, which make the routing a real challenge [22]. Since nodes are mobile, it is impossible to have an addressing scheme as it leads to large network overhead. The node position needs to be calculated (or predicted) for efficient data communication. Frequent location update can lead to excess use of energy and can increase collision in the network as well as excess energy drains [44]. Hence, the communication protocols designed should handle frequent topological changes and incorporate the location discovery. Also, communication protocol needs careful design to address limited energy and storage capacity of sensor node to ensure energy efficient reliable data delivery.

This thesis address the issues of both controlled and random mobility. Issues pertaining to applications with the help of controlled mobility and the corresponding protocol design are discussed. Thesis also discuss the protocol design issues with random mobility with respect to different mobility scenario in the later part.

Currently, a very few literature are/is available for routing protocol which requires further study to provide reliable, energy efficient and minimize latency in the protocol design. Different mobile scenarios need to be investigated separately to identify the design issues. Most of the current protocols available in the literature work with the assumption that neighbor node will remain stable. In case of mobile WSN, since nodes are dynamic and neighbors keep on changing, further studies need to be done to incorporate topology control into protocol design. Also, protocols need to be carefully designed to reduce the communication overhead, which may otherwise lead to an increase in traffic that may result in data collision and increase in interference which affect the QoS, reliability and latency. The protocol designed should minimize the frequent localization.

An attempt is made in this thesis to design protocols that address the issues of both controlled and random mobility, thereby providing energy efficient, reliable and QoS compliant data delivery in mobile WSN.

1.4 Objectives

The following objectives have been achieved in this thesis.

Objective 1: *To develop algorithms to improve the performance of applications such as event monitoring that are time critical. To develop an energy efficient area monitoring algorithm using mobile WSN.*

Objective 2: *To improve the performance of data communication in static WSN by using mobile nodes. To develop a protocol for ensuring interference free communication in sensor network in-order to enhance the data rate.*

Objective 3: *To develop an energy efficient routing protocol for mobile WSN by considering different mobility scenarios. The protocol development has to take into consideration the sleep-wake pattern of nodes and communication overhead of sensor network.*

Objective 4: *To develop an improved MAC protocol for mobile WSNs by considering topology changes of mobile nodes in the network.*

1.5 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 presents a background study of mobile WSN that further justifies the significance of the proposed work, particularly with respect to the research objectives.

Chapter 3 presents a methodology to improve event monitoring using mobile WSNs. This chapter suggests a node selection and placement algorithm to establish a communication path between any sensed event and the base-station. The proposed approach is further extended to monitor multiple events occurring in the sensor field. Chapter 3 also suggests an energy efficient area exploration algorithm that requires less communication and coverage overhead. A deterministic and a non-deterministic approach is proposed for exploring a given area with multiple mobile nodes. The same algorithm is used to discover partitions in a network for restoring network connectivity. Chapter 3 address the first objective proposed in this thesis.

Chapter 4 addresses the second objective. It describes a methodology to improve QoS in a static WSN by the use of mobile WSN to establish multiple paths from source (mobile node) to base-station via static sensors. For interference free communication, a frequency hopping based multi-channel MAC protocol has also been proposed.

Chapter 5 explains the need for an improved energy efficient routing that takes into account of the mobility element. The mobile node can act as a source, relay node, sink or can support a combination of these functions. A dynamic Steiner tree based algorithm for routing data from multiple sources to sink is proposed. An area based opportunistic routing algorithm that predicts the life time of links with neighbor is also proposed. Third objective is addressed in Chapter 5

Chapter 6 presents a MAC protocol for WSNs that have both static and mobile nodes. In this chapter, a contention based multichannel MAC protocol that uses mobility vector for neighbor selection is proposed. The MAC protocol presented here uses topology control for energy efficiency. A schedule based MAC protocol that uses dynamic TDMA is presented. Dynamic slots are assigned to a mobile node, based on the node mobility. Objective four is addressed in this chapter.

Chapter 7 concludes the thesis by summarizing the results and future directions of the work.

Chapter 2

Background of Mobile WSNs

Mobile WSN is a network of multiple nodes, where all or some of the nodes are mobile. Nodes in mobile WSN, sense, process and transmit the data towards the base-station or sink. Due to the mobility of the nodes, topology changes are inevitable and hence add complexity in designing the network protocol stacks. For example, in monitoring and tracking applications, the mobile object may be animal, vehicle or a person with a sensor node attached to it. This node senses data and transmits it to the base-station. Since the source node itself is mobile with no control on mobility, it adds new challenges in the design of WSN protocol stack. Also, the protocols designed for MWSN should be energy efficient, since WSNs are usually deployed for long periods.

The node's mobility also increases the network maintenance overhead since there is an increase in the number of control messages between nodes. As the topology becomes more dynamic, more number of control messages need to be exchanged to maintain connectivity in the network.

In mobile WSN, mobility might be involved at different network components. Source nodes(source) may be mobile [45], sinks (base-station) may be mobile [46], intermediate relay nodes, that route data from sensor nodes to the base-station can be mobile or it can be a combination of the above [22]. In general, a mobile WSN is defined as a network, where at least one or more nodes are mobile.

2.1 Mobile WSN Classification

As mentioned in chapter 1, mobile nodes are classified into *controlled* and *uncontrolled* mobility based on whether nodes move in a pre-determined pattern or the motion is unpredictable [22]. The node architecture and deployment pattern are different for both the types.

2.1.1 Controlled Mobility

In controlled mobility, mobile robots equipped with sensor nodes are used. These robots are programmed to move, collect data and route information in the sensor network.

In a self-organizing network, mobile nodes can be used for data collection or for improving the performance of the network by bridging any network partition, route optimization, static node localization, etc. In such a network, mobility can be controlled centrally or in a distributed manner. Since there is control over speed and trajectory, mobile nodes can be effectively used to provide better connectivity and coverage.

Mario et al., classifies mobile nodes as relocation nodes, mobile data collectors, and mobile peers, based on their role in the WSN by [22]. Relocation nodes are nodes that move to a certain strategic position so as to improve the performance of the network. Relocation node acts as a bridge between the static sensor nodes and the base-station [47]. These nodes move to a predetermined location. They remain static at that location and forward the data collected from the static nodes or from the sensor field to the base-station via multiple hops. Literatures are available that propose the use of relocation nodes for improving the connectivity in the network [48] [28].

Mobile data collectors (data mules) are nodes that move across the network to collect data sensed by static nodes [49]. Data mules improve the packet delivery ratio and reliability of the network. According to Francesco et al., there are two types of mobile data collectors: mobile sinks and mobile relays [22]. Energy consumption of the nodes that are close to the gateway/sink/base-station is always very high and results in partitions [50]. This is termed as funneling effect [51]. Funneling effect can be reduced by employing one or more mobile sink. Mobile sinks helps in uniform load distribution [22].

Mobile relays acting as data mules collect data from sensor nodes, store it and carry the data, move in the direction of the sink and deliver the data to it. Data mules are employed to gather data from static nodes and deliver it to the sink node using single hop communication thereby guaranteeing reliability and reduce energy consumption. Since data mules ultimately return to the sink, they can be recharged even if they are depleted of energy.

Mobile peers are sensor nodes, that sense as well as relay data. These nodes not only sense data, but also store and carry forward data from other nodes to the base-station. One of the primary uses of nodes, whose mobility can be controlled, is to improve the overall performance of the network.

In this thesis, one of the major focus is to improve the performance of WSN applications using controlled mobility. Though controlled mobility offers a wide variety of applications, here three main application areas are selected for discussion. Applications are selected based on its order of importance. Following applications are presented and analysed :

1. Event monitoring
2. Area Exploration
3. Improving QoS

Algorithms are proposed to improve the performance of event monitoring, area monitoring and QoS while considering the protocol design. Event monitoring with static sensor nodes possess several challenges. Two of the principal challenges are: bandwidth limitation and recording of incorrect data. Erroneous data from sensor nodes can lead to false alarms, which are equally undesirable as a missed event. Adding more sensor nodes or using multimedia sensors for monitoring will overcome the above issues. Both approaches are not economically feasible and require complex processing and large bandwidth for transferring the data. The main objective of the proposed event monitoring service is to develop an efficient system that employs stationary nodes, which will collaborate with a small set of mobile nodes. The main strength of this collaborative architecture is that it can monitor multiple events with minimum number of mobile nodes for time critical applications, while optimizing the cost.

Some of the applications that require area monitoring are: network maintenance, firmware updating, data collection, node localization, etc. For time critical applications, the aim is to explore the given area in minimum time and with minimum communication overhead to maintain energy efficiency. In monitoring applications, mobile nodes are directed across the sensor network to move from point to point in a predetermined and co-ordinated pattern so as to cover the entire area in terms of both communication and/or sensing coverage. For features such as OTA and static node localization, mobile nodes are programmed to visit every static node deployed in an area to either program them or map their position. The proposed approach can be used to cover the entire area efficiently in minimum time. We propose a deterministic as well as non-deterministic methodology of area monitoring with mobile robots.

One of the major applications of mobile WSN is to use them to improve the network performance and to improve the QoS of terrestrial WSN. Using controlled mobility, mobile nodes are positioned or trajectories are calculated so as to improve the performance of high bandwidth data communication.

Chapter 3 and 4, presents literature survey and methods that improves the performance of static WSN using random mobile nodes are discussed.

2.1.2 Random Mobility

Sometimes sensor node movement cannot be pre-determined or controlled and is completely random. According to Subir et.al., random mobility can be classified as deterministic or random pattern [52]. When mobility is deterministic, the future position of the mobile nodes can be predicted [53]. It is the most simplistic of all mobility models. In this model, a sensor node is interfaced to an object that follows a predictable path.

An example of this is a sensor node attached to a vehicle that is moving towards a particular destination. Vehicle usually moves at a certain constant speed and in a particular direction. The position of the mobile node at any point of time can be easily predicted. In random pattern, sensor nodes move randomly without any restrictions. In such a scenario, node discovery need to be done at regular intervals so as to maintain network connectivity. Mobility vector parameters such as speed, direction and location can be used to reduce the frequency of neighbor discovery without loss in connectivity. An example for such scenario is the use of sensor node in habitat monitoring [54]. WSNs have multiple constraints such as energy, transmission range, processing power, limited bandwidth and memory. These issues are compounded by the introduction of mobility especially if the node mobility cannot be controlled.

Introduction of random mobility adds to the challenges of designing a network protocol stack for WSN. Channel fading, shadowing, interference, node failure are some of the challenges. Also re-computation of routes is not feasible in a dynamic topology.

Due to energy constraints and processing constrains, sensor nodes cannot run complex routing algorithms. Some of the major sources of energy wastage in the network are

- Collisions
- Packet re-transmission

In the case of static WSNs, as the topology is known in advance, energy optimization can be easily implemented. Energy consumption can be reduced by controlling transmission, power levels, reducing control overhead or by using a sleep wake cycles. Nodes with low residual energy can be avoided. Due to dynamic topology, in mobile networks, implementing energy optimization schemes into the routing protocol becomes complex.

Routing can be classified as direct, flat and hierarchical based on the network architecture and organization [23]. In direct routing, all nodes are in transmission range of the base-station and hence send data directly. Since all nodes need to communicate directly with the base-station, energy consumed will be high. Collisions while transmission will also be high as multiple nodes may attempt to send data simultaneously to the base-station .

Flat architecture is used in case of homogeneous WSNs [55]. Here all nodes have similar architecture and features. Data is transfered from the sensor nodes to the base-station via multi-hop peer to peer network. Opportunistic Routing (OR) is used in such a network architecture [56] [57]. OR is usually geographical routing. OR also uses sleep-wake cycle to reduce energy consumption; this may not be feasible if some nodes are mobile. In a hierarchical routing, nodes are organized into multiple-levels of clusters. As the topology is dynamic, cluster reformation needs to be done frequently.

Routing algorithms can also be classified as topology based and location based [23]. In topology based routing, the network layout is used to form an optimal route from source to destination. In topology based routing all or some nodes need to have complete details of each and every node in the network. This routing technique is not feasible with dynamic topology, as topological changes are frequent. The destination may be a single node or target zone. In geographical routing, the forwarding node is selected based on its proximity to the destination. For this every node has to maintain neighbor information. If the topology is dynamic, frequent updates of neighbor information will be required.

Mobility might be involved at the different network components. For instance, nodes may be mobile and sinks might be static, or vice versa. Network protocol design can vary with the type of mobility. This thesis aims to investigate different scenarios of mobility for protocol development, considering energy efficiency, packet delivery ratio, connectivity etc.

In Chapter 5 and 6, we have proposed routing and MAC protocols that can be used for mobile WSNs. In this thesis, we have considered varying mobility scenarios and proposed routing and MAC protocols that can be used.

2.2 Mobile Node Architecture

A WSN node (termed as mote) has an on-board microcontroller/DSP processor, memory, sensors and their interfacing circuits, communication module and an energy source. The communication module usually use the low range Zigbee protocol, IEEE 802.15.4 developed for Personal Area Networks(PAN). The physical layer of IEEE 802.15.4 supports 868/915MHz as well as the 2.4GHz frequency bands [58]. The physical layer of IEEE 802.15.4 includes low cost deployment, low complexity and availability of low power nodes to reduce power consumption. Mobile node architecture is shown in Figure 2.1

A radio transceiver can be in any of the following states: transmit, receive, idle and sleep. During transmit and receive state, node's transceiver transmits or receives packets. In idle state, the transceivers continuously listens to the media. In sleep state, the radio is switched off and this reduces power consumption. In this state, the node does not send or receive any packet.

Nodes in WSNs are usually battery-powered and some nodes die due to battery failure. Hence any protocol designed must be energy efficient. The power consumption in mobile WSN is higher than in static WSNs. In the case of mobile WSN, apart from sensing, processing and communication, power is also needed to support mobility. However, a mobility module in mobile WSN is often integrated with additional power source or may use energy harvesting techniques such as solar and inductive recharging [4]. In case of scenarios where the sensor node is mounted on a vehicle, animal or human, the energy need not be expended to obtain mobility. Despite this, since mobility is uncontrolled, a large number of control messages need to be exchanged to maintain connectivity in the network and this increases energy consumption.

2.2.1 Controlled Vs Random Mobility

The hardware architecture of the mobile node is different for controlled and random mobility. In controlled mobility based WSN nodes, the sensor node architecture is similar to a static sensor node with an addition of separate mobility module. The mobility module is used to move the sensor node from one point to another within the sensor field. The on board microcontroller does this via the driver circuit. For better mobility control, the mobile node may contain additional sensors such as orientation sensors and magnetometers to keep track of the position. In scenarios where mobile anchor nodes are used for static node localization, localization unit such as a GPS module is added to obtain the node location information. In certain applications where the mobile nodes handle high bandwidth data such as multimedia data requires the support for multiple communication protocols must be added. An application where mobile node is used as base-station requires high power microcontroller for complex and fast computation for handling all network traffic [59] [60]. A mobile base-station may also require multi-communication protocol support when the data needs to be sent to a remote location for post processing [61]. Mobile nodes used as relocation nodes or data mules may not have a sensor subsystem. In case of controlled mobility, mobile robots are equipped with separate power supply for powering the motor [62] [63] [64].

In random mobility, the node architecture is similar to that of a static sensor node except that the node is mounted on a moving object such as animal, person or a transport vehicle. Here the physical mobility is due to the object movement. In certain applications where the sensor nodes are attached to animals or vehicles to act as data mules or mobile base-stations, the sensing module may not be needed. When sensor nodes are attached to vehicles, that can be used to power up the sensor nodes, there is no need for a separate power source.

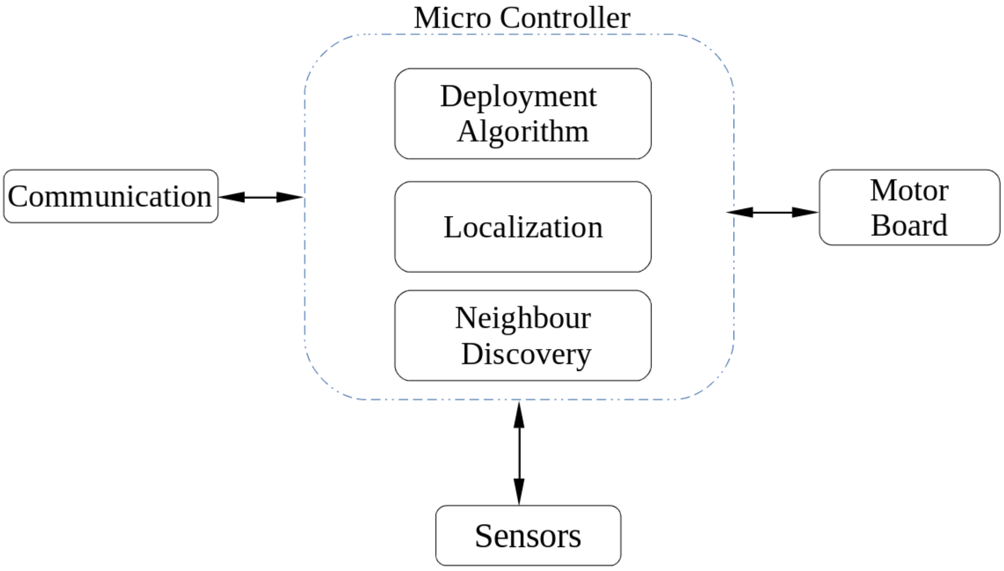

FIGURE 2.1: Mobile node Architecture

2.3 Node Deployment

2.3.1 Controlled Mobility

The deployment of WSN nodes may be structured or unstructured depending on the application. Deploying sensor nodes in a grid structure is an example of structured deployment. Unstructured or Random deployment is done by scattering the sensor nodes over an area of interest. Such deployment strategies need to address issues such as coverage and connectivity. These issues can be resolved by making use of mobile sensor nodes. The general deployment algorithm is shown in Figure 2.2. Deployment algorithm consists of three parts: neighbor discovery, deployment objective calculation and mobility.

During neighbor discovery phase, the mobile node broadcasts information regarding position, energy level, etc., to its neighbors and in turn receives similar information from its neighbor nodes. The data obtained via neighbor discovery is then used by the deployment objective module to frame the trajectory for the mobile nodes to reach a pre-determined destination by satisfying constraints such as connectivity, energy consumption etc. Deployment objective module employs different mathematical models, based on either probabilistic or geometrical techniques, that are used to place the mobile node at optimal location. Finally, the mobility module executes commands to move the mobile nodes towards selected destination.

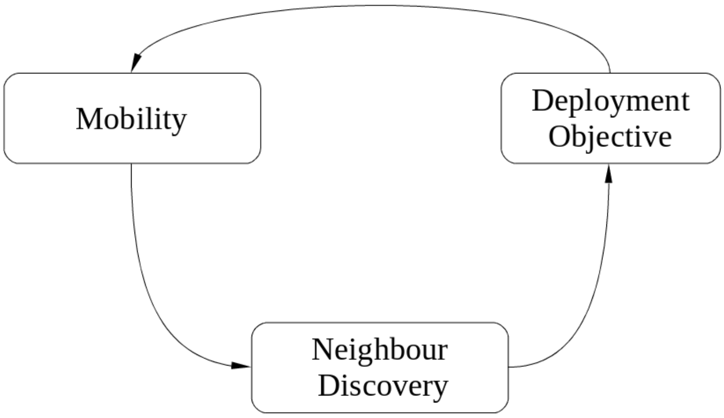

Figure 2.2: Block diagram of general deployment algorithm

Sensor deployment based on controlled mobility can be classified as: centralized and distributed [65]. In the centralized scheme, the base-station has all the required information such as node's position, velocity and residual energy of a node. Using this information, it calculates the optimal placement of mobile nodes in the network and directs the movement of the mobile nodes to their future position. This approach gives optimal results, as the optimization algorithm can be employed directly for the position calculation. The major challenges in this scheme include acquiring the information from all the nodes in the network. Another major issue is the scalability that in turn may increase energy consumption. Increasing the size of the network or the number of nodes increases communication overhead. Also, if the topology is highly dynamic in nature, the centralized algorithm may fail due to the need of frequent topology updates and resulting control overhead.

In the distributed scheme, decisions are made locally. Hence the distributed scheme is more scalable. Other advantages include reduction of latency in decision making and reduction in control overhead. The main disadvantage of the distributed system is that the node's position computed may not be optimal as the entire network topology and characteristic are not known. Obtaining such information locally is challenging in-terms of communication overhead and bandwidth constraints.

2.3.2 Random Mobility

In case of random mobility, deployment objective module is not required. The mobility depends on the movement of the object to which the node is attached to. Since the node movement is random in nature, the network topology changes are unpredictable. Hence there is a need for frequent neighbor discovery before data be communicated.

Since there is no control over mobility- mathematical models are used to represent nodes mobility pattern. The mobility pattern is modelled in terms of the location, velocity and change in the acceleration of a mobile node with respect to time. The mobility models are used for simulating the performance of a network. The deployment algorithm uses the mobility model to predict the future position of the mobile node. Mobility models are classified as *random models, models with temporal dependencies, models with spatial dependencies* and *models with geographic restriction.*

In the case of random mobility, the change in speed and direction is application specific. For example, the probability that an animal carrying a sensor node moves in a straight line at constant speed for a long duration is less compared to a sensor node attached to a vehicle. Mobility models should incorporate such scenarios in their mathematical model. To emulate the real-life random mobility pattern, different mobility models have been described in the literature [66]. Random mobility models are classified as *random-waypoint model, random-direction model* and *random-walk model.*

Random-waypoint mobility model: Random-waypoint mobility model is the preferred mobility model as it covers a wide range of applications [66]. In random-waypoint, each mobile node randomly selects a destination location and then travels towards it with constant velocity. A node will choose a velocity between $[0, V_{max}]$ every time before it decides to move. The velocity V_{max} is the maximum velocity allowed. A node after reaching the destination will pause for a time Δt_{wait}, where $\Delta t_{wait} \geq 0$. If we assume that Δt_{wait} time is zero, then the average nodal speed is $\frac{V_{max}}{2}$. But in practice

$\Delta t_{wait} > 0$, so the average nodal speed may not be the correct method to calculate the node speed. The trajectory of a mobile node moving random-waypoint mobility model is shown in Figure 2.3.

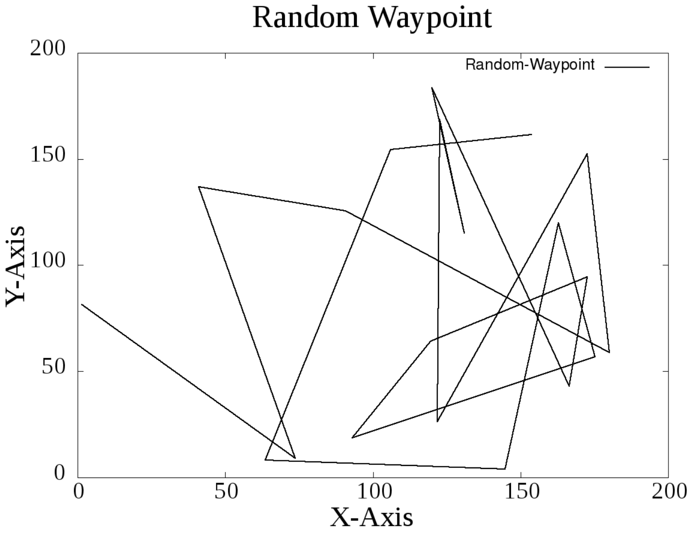

FIGURE 2.3: Random-waypoint mobility model

Bai et.al., [66] proposed a mobility metric to capture the node speed using the relative motion of the nodes. If $l(n, t)$ is the position of node n at time t, the relative velocity $v(x, y, t)$ between node x and y at time t and at speed, R_S, given as

$$v(x, y, t) = \frac{d}{dt}(l(x, t) - l(y, t)). \tag{2.1}$$

The mobility metric M_{xy} of any pair of nodes, defined as the absolute relative speed, is given by the equation

$$M_{xy} = \frac{1}{T} \int_{t_0 \leq t \leq (t_o + T)} |v(x, y, t)| dt \tag{2.2}$$

For calculating the total mobility metrics of the scenario M, the value of M_{xy} given by equation 2.2 is averaged over all the mobile node pairs and is given by the equation

$$M = \frac{2}{n(n-1)} \sum_{x=1}^{n} \sum_{y=x+1}^{n} M_{xy} \tag{2.3}$$

The mobility metric helps to classify the different mobility scenarios.

Random-direction mobility model: In random-waypoint algorithm, the mobile nodes tend to cluster at the centre of simulation area. This problem is known as border effect which can be overcome by using random-direction model. In random-direction mobility model, nodes choose a random angle θ and move to the boundary of the simulation area. Once it reaches the boundary, the node may pause for a small interval of time Δt_{wait} (where $\Delta t_{wait} \geq 0$) and then repeats the process. The trajectory of a node moving random-direction mobility model is shown in Figure 2.4.

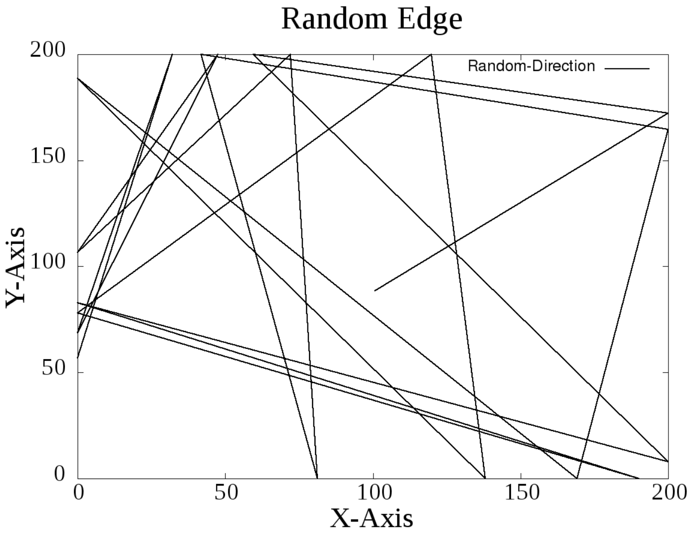

Figure 2.4: Random-direction mobility model

Random-walk mobility model: Random-walk model, also known as *Brownian motion,* is used to simulate the unpredictable movement of the mobile nodes. At each interval, the node chooses a velocity randomly between $[0, V_{max}]$ and an angle θ between $[0, 2\pi]$ for its movement. Unlike the above mentioned models, the random-walk model has zero pause time. This model is Markovian, since the next movement of the node is not based on history. Such a mobility model can be rarely seen in real life applications. Figure 2.5 shows the trajectory of node movement using random-walk.

Mobility models with *temporal dependencies* can be classified as *Gauss Markov model* and *Smooth random mobility model.* In *Gauss Markov model*, the velocity of the mobile node is assumed to be correlated over time and modelled as a Gauss-Markov stochastic process. In this model, when the node goes beyond the boundaries of the simulation area, the direction of movement is forced to flip 180 degrees to keep the nodes inside

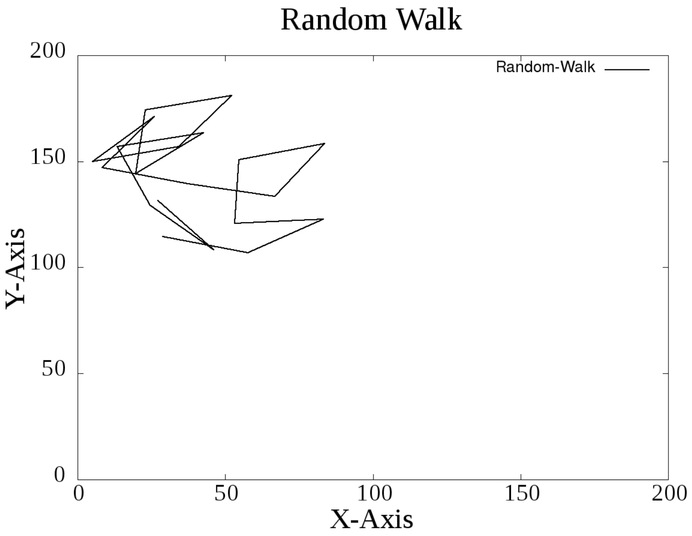

FIGURE 2.5: Random-walk mobility model

the boundary. In the *smooth random model*, the frequency at which the node speed varies is assumed to be a Poisson process. The mobile node will adjust its current speed to the targeted new speed by acceleration or deceleration.

Mobility model with *spatial dependencies* are classified as *Reference Point Group Model (RPGM)* and *set of correlated model.* RPGM model, deals with group mobility of mobile nodes. In RPGM model, each group has a centre, which is either a logical centre or a group leader node. The movement of the group leader determines the mobility behaviour of the entire group. Set of *Spatially correlated models* include *Column mobility model, Pursue mobility model* and *Nomadic Mobility model.* These models are expected to exhibit strong spatial dependency between neighboring nodes. *Column mobility model* is used for the mobile nodes that moves in a certain fixed direction. *Pursue mobility model*, is used for the scenario where multiple mobile nodes attempt to follow a single mobile node's movement. The *nomadic mobility model* is used for modeling the scenario where a group of nodes move from one location to another randomly.

In *geographical restriction based mobility model*, the movement of nodes is bounded by streets, freeways or obstacles. Two categories of geographical restriction based mobility models are *pathway mobility model* and *obstacle mobility model.* In *pathway based mobility model*, the map is predefined in simulation. In this model the mobile nodes are only allowed to travel on the pathways. In *obstacle based mobility model*, an obstacle in the form of rectangular boxes is randomly placed within the simulation area. The mobile nodes need to select a trajectory that avoids colliding with such obstacles. Also, if an obstacle is in-between two mobile nodes, the transmission link will be considered to be disconnected.

2.4 Summary

Controlled mobility is used for improving monitoring applications and network performance. Mobile nodes are deliberately added to the network to remove network partitions and to improve reliability of data delivery. If mobility is inherent in the environment, whether the mobility is controlled and un-controlled, the network protocol stack must be re-designed to support dynamic topology. The changes in the protocol stack are primarily in the MAC and network layer.

Chapter 3

Mobile WSN Deployment Applications

In this chapter, the problem of event monitoring with mobile WSN is presented in Section 3.1. Section 3.2 describes the algorithm used for area monitoring using mobile nodes. We adopt the approach used in Section 3.1 and Section 3.2 for partition discovery and connectivity restoration. This is presented in Section 3.3.

3.1 Better Event/Environment Monitoring

The main challenge in a time-critical event monitoring system is to transfer the mission critical data to the base-station with minimum delay, while minimizing energy consumption. In real-time applications such as battlefield reconnaissance, fire detection in forests, air quality monitoring, intruder detection, etc., sensor nodes are deployed over large areas [67]. A large number of sensor nodes are used for detecting various events and reporting them to a base-station.

In case of event detection, mobile nodes can be send to sense the event and hence to report to the base-station. This can be done dynamically by rearranging the nodes after detecting an event. Using mobile nodes in such scenario can help in increasing the network life time. With special communication modules and sensors, this approach can provide better QoS when compared to their static counter part. In this section, we propose the use of mobile nodes for event monitoring application. Using mobile nodes for event monitoring is not novel and several literature are available [68] [69]. The aim of the proposed approach is to improve the QoS of time critical event monitoring application. Highlight of the proposed approach includes

- Optimal mobile node selection with minimal number of mobile nodes
- Fast connectivity establishment between event and the base-station
- Handling multiple events
- Mobile node architecture with multi-communication protocol.

The proposed approach uses minimum number of mobile sensor nodes for forwarding details of multiple events to the base-station using the shortest path possible in minimum time.

The approach suggested uses controlled mobility, where the placement of the mobile node is done by the base-station after detecting an event. These mobile nodes are further used to route the data between the event location and the base-station. The main objective is to minimize the time taken for the movement of multiple mobile nodes from their current position to a new one. These mobile nodes are supported by multiple-communication protocol, so as to increase the connectivity via multi-hop paths between the base-station and the event location. The mobile node architecture proposed here uses IEEE 802.11 and IEEE 802.15.4 for communication. A modification of Hungarian Algorithm has been used for determining the optimal positions for multiple mobile nodes in the sensor field. This algorithm has been further extended to monitor multiple events in the sensor field using mobile nodes. We propose to use the Steiner Minimum Tree (SMT) to determine the route via which information about the events is relayed to the base-station. We intend to use a modified version of an approximation algorithm proposed in [70] to solve the SMT problem.

3.1.1 Proposed Algorithm

Problem Statement and Assumption

The sensor field is represented as a rectangular area A, where $A = R_l \times R_b$, R_l and R_b represent the length and the breadth of the rectangle, respectively. Static sensor nodes are deployed in the area A and are represented using a set S. A set of mobile nodes is randomly placed at position $\{p_1, p_2, ...p_m\}$, where p_i represents the coordinates of m mobile nodes in A.

If an event is detected in the sensor field at co-ordinate (X_E, Y_E), the base-station selects n nodes from a set of m mobile nodes and places them optimally between the location of the event (X_E, Y_E) and the base-station in minimum time. We extend this approach to transfer information about multiple events to the base-station using minimum number of mobile nodes. We have used the concept of Steiner tree along with a modified Hungarian algorithm for mobile node selection and placement. It is assumed that the mobile sensor nodes identity their locations through GPS or through other localization methods [71].

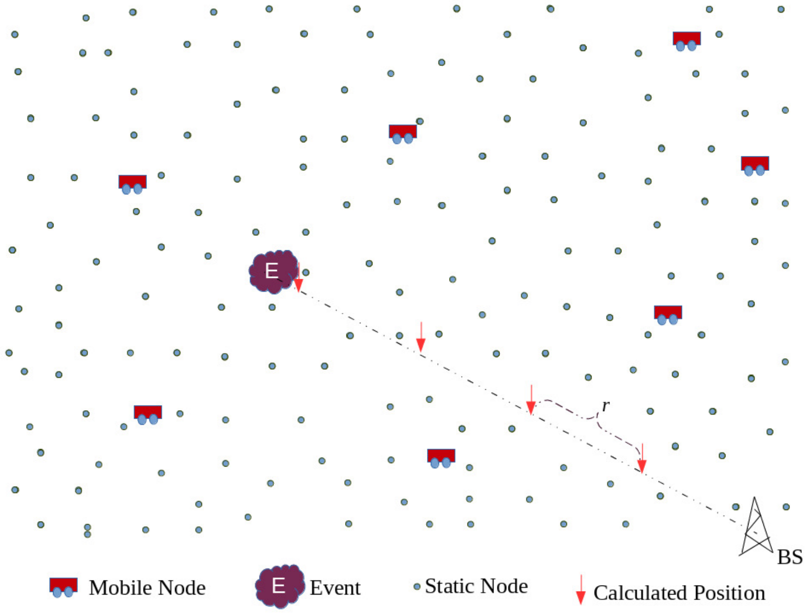

FIGURE 3.1: Initial deployment of sensor motes and mobile nodes on the field

Since an event can occur at any point in time, it is important that the base-station should be regularly updated with the location of the mobile nodes. Mobile nodes update their location information to the base-station through a static neighbor node. The scenario of the initial deployment of mobile nodes is shown in the Figure 3.1. When an event occurs, static nodes cooperatively determine the event location and an alert pertaining to the event is sent to the base-station. The base-station, then calculates the coordinates at which the mobile nodes must be placed in the sensor field. The base-station needs to select a set of mobile nodes n from a set of m nodes where $n \leq m$, to establish a line of connectivity in minimum time and begin transferring data from the event location.

The proposed approach has two phases.
Phase I: Calculate the coordinates at which the mobile nodes are to be placed.
Phase II: From the set of m mobile nodes, calculate the most optimal solution of mobile nodes with the calculated coordinates, so that the total time required to establish the connection is minimal.

Phase I: Algorithm to find the mobile node future position

When a sensor node detects the presence of an event, it updates this information to the base-station via static nodes. The base-station on receiving the location of the event, calculates the Euclidean distance between the event and the base-station. The parameters used to model this are listed in Table 3.1. Let d be the distance between the event location and the base-station. The distance d is given by

$$d = \sqrt{(X_B - X_E)^2 + (Y_B - Y_E)^2} \tag{3.1}$$

The maximum number of mobile nodes required in the sensor field for a single event monitoring is

$$m = D/r \tag{3.2}$$

where D is the maximum distance between the base-station and any other point in the sensor field and r is the *safe communication range* [72]. The safe communication range is the maximum distance between two mobile nodes where 99.99% packet delivery ratio is guaranteed. The actual number of mobile nodes required is given by the equation,

TABLE 3.1: Parameters used: To find the mobile node future position

Parameter	Description
(X_B,Y_B)	Base-station position in terms of co-ordinates.
(X_E,Y_E)	Event position
m	Total number of mobile nodes needed in the sensor field
n	Number of mobile nodes needed for the communication between the event and the base-station where $n \leq m$
V	speed of mobile nodes in the sensor field
D	Maximum distance between the base-station to any other point in the sensor field
r	Safe communication range of mobile nodes

$$n = \lceil (d/r) \rceil \leq m \tag{3.3}$$

Algorithm 3.1 Algorithm to find mobile nodes future position

```
procedure FUTURE_POSITION($(X_B, Y_B)$,$(X_E, Y_E)$, $r$, $D$, $V$ )
    Calculate $d = \sqrt{(X_B - X_E)^2 + (Y_B - Y_E)^2} \leq D$
    Calculate $n = \lceil \frac{d}{r} \rceil \leq m$, where $r$ is the safe communication distance.
    Let $(X_0, Y_0)$=$(X_B, Y_B)$ and let $\theta = \tan^{-1} \left\{ \frac{Y_E - Y_B}{X_E - X_B} \right\}$
    for all $i$ : 1 to $n$ do
        $X_i = X_{i-1} + r\cos\theta$
        $Y_i = Y_{i-1} + r\sin\theta$
    end for
    Construct a distance matrix by computing the Euclidean distance from current positions to future positions of mobile nodes.
    Convert the distance matrix into time matrix using the equation, time = $\frac{\text{distance}}{\text{speed}}$
end procedure
```

Algorithm 3.1 computes the coordinates where the mobile nodes are to be placed. In Algorithm 3.1, steps 5-8 calculates the new position of the mobile node, where

$\{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ are the points in the straight line joining (X_B, Y_B) and (X_E, Y_E). The Euclidean distance between (X_{i-1}, Y_{i-1}) and (X_i, Y_i) for $i = 1, \ldots, n$ is r. and,

$$\sqrt{(X_B - X_n)^2 + (Y_B - Y_n)^2} = nr \approx d \tag{3.4}$$

The above polynomial time complex algorithm generates the coordinates for the mobile node placement between the base-station and the event location. It also computes the time matrix for each mobile node with respect to the calculated coordinates. The time matrix is computed using the distance between the mobile node and the calculated coordinates based on the speed of the mobile node using the equation $t' = d/V$, where d is the distance and V is the speed.

Phase II: Algorithm for assigning the mobile node

Assigning a set of mobile nodes to the calculated coordinates could be solved using standard assignment algorithms. The Hungarian algorithm is an optimization based method that solves the assignment problem in polynomial time [73], but it does not provide an optimal solution for the given scenario. Consider the scenario where there are two mobile nodes m_1 and m_2 (shown in the Figure 3.2). Let $\{Q_1, Q_2\}$ be the calculated coordinates for placing the mobile nodes. The aim is to assign mobile nodes $\{m_1, m_2\}$ to $\{Q_1, Q_2\}$ in minimal time. The distance between the calculated coordinates and the mobile nodes is as shown in Figure 3.2 and the amount of time required for moving a mobile node to the co-ordinates is proportional to the corresponding distance. The Hungarian algorithm uses a *minimum matching* assignment [73] and matches $m_1 \rightarrow Q_1$ and $m_2 \rightarrow Q_2$. This assignment brings a reduction in the overall distance the mobile nodes need to travel. Since the mobile nodes move simultaneously, the total time for the mobile node to align itself between the event and base-station is proportional to $\max(10, 40)$. On the other hand, if we assign $m_1 \rightarrow Q_2$ and $m_2 \rightarrow Q_1$, the overall time will be reduced to 30. Hence, there is a need for a new assignment algorithm which can reduce the overall time.

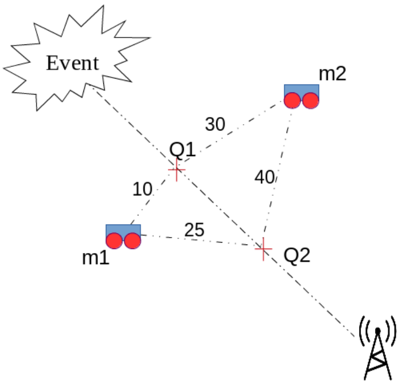

FIGURE 3.2: Scenario of optimal node assignment

Let P_1, P_2, ..., P_m be the current location of mobile nodes and let Q_1, Q_2, ..., Q_n be the new locations. $a_{i,j}$ denotes the time required for the mobile node at P_i to reach Q_j. The objective is to compute the minimum time required to complete the placement of all mobile nodes. The relocation of the mobile node from P_i to Q_j is denoted by a pair (P_i, Q_j) or simply as (i, j). Considering the movement of mobile nodes from their current locations to new locations : $X = \{(P_1, Q_{\alpha_1}), (P_2, Q_{\alpha_2}), \dots, (P_m, Q_{\alpha_n})\}$, where $\alpha_i \in \{1, 2, \dots, n\}$ and $\alpha_i \neq \alpha_j$ for $i \neq j$. The time required to complete the relocation to X is indicated in the equation 3.5.

$$f(X) = \max\{a_{i,\alpha_i} \ : \ 1 \leq i \leq n\} \tag{3.5}$$

The objective is to find *a relocation point X such that $f(X)$ is minimum.* Let us define the relocation problem on time matrix $A = [a_{i,j}]_{n \times n}$ precisely.

Relocation Problem on $A = [a_{i,j}]_{n \times n}$

Minimize $f(X)$, where

$$f(X) = \max_{\beta, X} \sum_{i=1}^{n} \left(\beta_i \sum_{j=1}^{n} a_{i,j} x_{i,j} \right), \tag{3.6}$$

subject to the constraints

$$\sum_{i=1}^{n} x_{i,j} = 1, \sum_{j=1}^{n} x_{i,j} = 1, \sum_{i=1}^{n} \beta_i = 1$$

$$x_{i,j}, \beta_i \in \{0, 1\}, \text{ for } 1 \leq i, j \leq n$$

$a_{i,j}$ = travel time between i^{th} and j^th positions. $X = [x_{i,j}]$ is a binary matrix (relocation matrix) and every assignment, column sum of X = row sum of X = 1 and $x_{i,j} = 1$ if i^{th} vehicle moves to j^{th} position, else 0. $\beta = (\beta_1, \beta_2, ..., \beta_n)$ a binary vector. Presence of β_i enforces that $f(X)$ computes overall maximum time (not total time) required to complete the relocation. Then we minimise $f(X)$ over all possible relocation.

(X) computes the maximum over all time to complete the relocation X and then we minimise $f(X)$ over all possible X(relocations).

It can be noted that constraints imposed on $x_{i,j}$ ensure that the mobile node at P_i moves exactly to its designated new location. Also, $\sum_{i=1}^{n} \beta_i = 1$ guarantees that $f(X)$ is the time required to complete the placement of all mobile nodes in X. The relocation problem on A has certain characteristics similar to a well known assignment problem [74].

Assignment Problem on $A = [a_{i,j}]_{n \times n}$

Minimize $g(X)$, where

$$g(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} x_{i,j} \tag{3.7}$$

subject to the constraints,

$$\sum_{i=1}^{n} x_{i,j} = 1, \sum_{j=1}^{n} x_{i,j} = 1$$

$$x_{i,j}, \in \{0, 1\}, \text{ for } 1 \leq i, j \leq n$$

The complexity of the Hungarian algorithm [74] to compute an optimal solution for this assignment problem is $O(n^3)$.

Relation between assignment and relocation problem

The following observation shows that there is no direct link between the relocation problem and the assignment problem.

Observation: An optimal solution to the assignment problem on $A = [a_{i,j}]$ may not be an optimal solution for the relocation problem on A and vice versa.

Let us consider the following matrix to verify the above observation.

$$A = [a_{i,j}] = \begin{pmatrix} 6 & 7 & 21 \\ 20 & 5 & 4 \\ 4 & 22 & 5 \end{pmatrix}$$

Consider the following assignments on A:

(i) $U =: \{(1,2),(2,3),(3,1)\}$, $a_{1,2} = 7, a_{2,3} = 4, a_{3,1} = 4$

(ii) $V =: \{(1,1),(2,2),(3,3)\}$, $a_{1,1} = 6, a_{2,2} = 5, a_{3,3} = 5$,So

(a) $g(U) = 7 + 4 + 4 = 15$ and $g(V) = 6 + 5 + 5 = 16$.

(b) U is an optimum solution for assignment problem on A and $g(U) = 15$,

(c) $f(U) = \max\{7,4,4\} = 7$, $f(V) = \max\{6,5,5\} = 6$,

(d) V is an optimal solution of the relocation problem on A and $f(V) = 6$.

It may be noted that (i) U is an optimal solution of the assignment problem on A but not an optimal solution of the relocation problem on A and (ii) V is an optimal solution of the relocation problem on A but not an optimal solution of the assignment problem on A.

The algorithm proposed in Algorithm 3.2 is a simple technique to compute an optimal solution for the relocation problem using the Hungarian method that is used to compute an optimal solution for an assignment problem.

Applying the Hungarian algorithm on matrix A in the example to compute a minimum

Algorithm 3.2 Algorithm to solve the relocation problem

```
procedure RELOCATION_PROBLEM($A = [a_{i,j}]_{n\times n}, n \geq 2, a_{i,j} > 0, integer, M := \sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j}, L = 0$)
    Find an optimal assignment $U$ on $A$ using the Hungarian algorithm and let $L = f(U) = \max\{a_{i,j} : (i,j) \in U\}$. Let $R := U$
    Modify the matrix $A$ by
        $a_{i,j} = \begin{cases} M, & \text{if } a_{i,j} \geq L \\ a_{i,j}, & \text{if } a_{i,j} < L \end{cases}$
    Find an optimal assignment $V$ on $A$ by using the Hungarian algorithm and $f(V) := \max\{a_{i,j} : (i,j) \in V\}$
    if $f(V) < M$ then
        $L := f(V), R := V$
        goto Step 3
    else
        Return $R$
    end if
end procedure
```

assignment.

$$A = \begin{pmatrix} 6 & \mathbf{7} & 21 \\ 20 & 5 & \mathbf{4} \\ \mathbf{4} & 22 & 5 \end{pmatrix}$$

Clearly $U = \{(1,2),(2,3),(3,1)\}$ is an optimal assignment of assignment problem and $f(U) = 7$.

Let $L = f(U) = 7$ and $M = \sum_{i=1}^{3}\sum_{j=1}^{3} a_{i,j} = 94$

Modifying the matrix A by

$$a_{i,j} = \begin{cases} M, & \text{if } a_{i,j} \geq L \\ a_{i,j}, & \text{if } a_{i,j} < L \end{cases}$$

Hence the modified $A_1 = \begin{pmatrix} 6 & 94 & 94 \\ 94 & 5 & 4 \\ 4 & 94 & 5 \end{pmatrix}$.

Reapplying the Hungarian algorithm on A_1.

$$A_1 = \begin{pmatrix} \mathbf{6} & 94 & 94 \\ 94 & \mathbf{5} & 4 \\ 4 & 94 & \mathbf{5} \end{pmatrix}$$

Clearly $V = \{(1,1),(2,2),(3,3)\}$ is an optimal solution of assignment problem on A_1 and $f(V) = 6$.

Note that $6 = f(V) < L = 7$. So update the value of $L = f(V) = 6$ and modify the matrix A_1 as above.

$$A_2 = \begin{pmatrix} 94 & 94 & 94 \\ 94 & 5 & 4 \\ 4 & 94 & 5 \end{pmatrix}.$$

Therefore, $W = \{(1,2),(2,3),(3,1)\}$ is an optimal assignment on A_2 and $f(W) = 94 = M$. Note that $f(W) = M = 94 > L = 6$. Hence stop. So V is an optimal assignment for the relocation problem on A.

The idea is to ignore expensive assignments in each iteration and look for an optimal solution for the assignment problem for the original A. We stop this process at a stage where we need to use a modified assignment; i.e. we have to move out of original A to find an optimal assignment in modified A. Since the complexity of the Hungarian algorithm is $O(n^3)$, the complexity of the modified algorithm is $O(Mn^3)$=$O(n^3)$, where $M := \sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j}$

Proof of correctness: Note that f and g denote the objective functions of the relocation problem and the assignment problem, respectively.

Let A_0(same as A)$, A_1, \ldots, A_l$ be the sequence of modifications on A during the execution of the algorithm and let V_0(same as U)$, V_1, \ldots, V_{l-1}$(same as R^*)$, V_l$ be the corresponding sequence of assignments produced by Step 1 and 2, where R^* be the output of the algorithm. Note that A_l is the last modification on A and $f(V_{l-1}) = f(R^*) < M$. Due to Step 2(i) and 2(iii) of the relocation algorithm, (a) $f(U) = f(V_0) > f(V_1) > \ldots, > f(V_{l-1}) = f(R^*)$ and (b) $f(V_l) = M$. Moreover, $g(V_l) > M$ on A_l due to $n \geq 2$, $a_{i,j} > 0$, and $f(V_l) = M$.

Lemma: The output R^* of the algorithm is an optimal solution of the relocation problem on A.

Let N be an optimal solution of the relocation problem on A. Hence $f(N) \leq f(R^*) < M$. Hence, it is proved that $f(N) = f(R^*)$.

If $f(N) < f(R^*)$, then we arrive at a contradiction on minimality of V_l on A_l. Since $f(N) < f(R^*) < M$, for any $(i, j) \in N$, the value $a_{i,j}$ is never updated during the execution of the algorithm and kept intact in A_l. Hence $g(N) < M$ on A_l. But $g(V_l) > M$ on A_l, which contradicts the minimality of V_l on A_l in Step 2(ii). This completes the proof. At the end of Phase II, the mobile nodes are moved to their corresponding coordinates as shown in Figure 3.3. In Algorithm 3.2, we can create dummy AGV or locations if the number of AGV's are not equal to the co-ordinate locations. Giving minimum value to the corresponding dummy variable will solve the problem. Since f(X) go for over all maximum time required, dummy variable does't affect the solution.

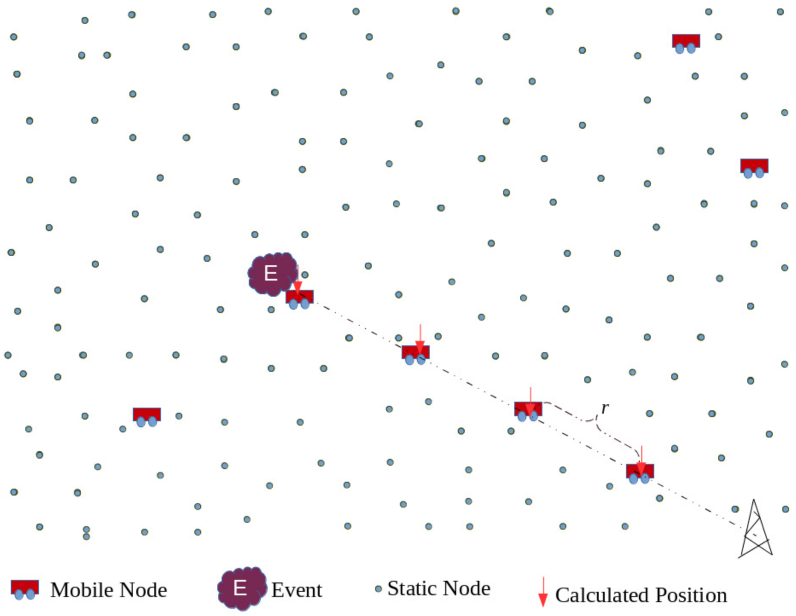

FIGURE 3.3: Final placement of Mobile Nodes

3.1.2 Multi-event Monitoring

We extend our proposed approach to address the assignment of mobile nodes to connect multiple event locations in the sensor field. The aim is to assign mobile nodes in the sensor field, so that the event occurrences can be communicated to the base-station in minimum time. The extended algorithm uses a set of Steiner points to place the mobile nodes. Steiner points are points that are introduced in a graph to connect a set of terminal vertices, such that the sum of the edge weights of the resulting tree is minimum. The resultant tree is known as Steiner Minimum Tree (SMT). The Steiner tree algorithm is similar to Minimum Spanning Tree (MST), except that with extra non-terminal vertices, the total length of the tree is reduced. Since, finding an optimal Steiner tree is a NP-hard problem [75], we have adopted approximation techniques to calculate a SMT, which though, not an optimal solution, is a feasible one.

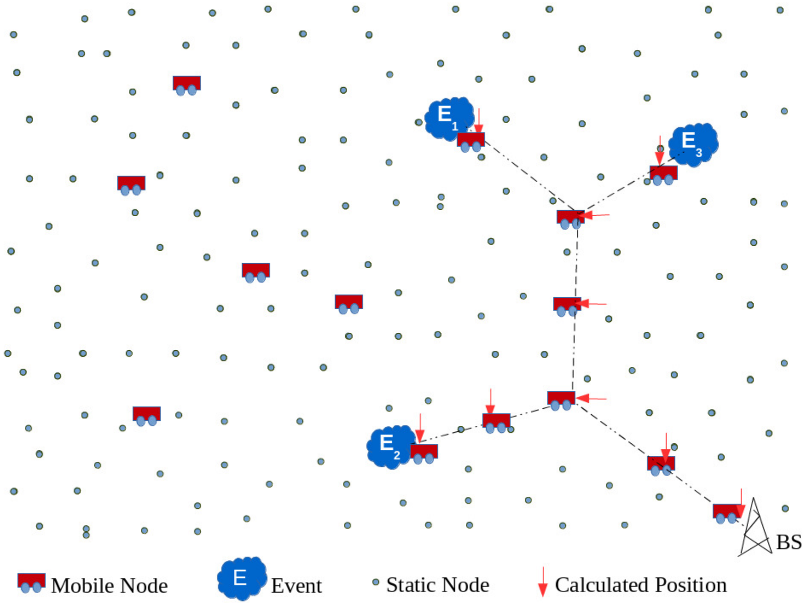

FIGURE 3.4: Mobile node relocation for connecting multiple events

Once the events are identified, the base-station must calculate the positions to place the mobile nodes, so that the connection between event locations in the field is established while keeping the number of mobile nodes required minimum. The aim is to interconnect the set of vertices (events and base-station) by a tree with shortest length. We propose to use an approximation algorithm for Steiner tree to solve this problem. In comparison with the MST problem, Steiner tree algorithm calculates an extra set of vertices called Steiner points, which when added, will result in a tree of minimum length. We have modified the Iterative-1 SMT approach proposed by Kahng and Robins for finding the Steiner set S [70].

Steiner tree aims to connect a given set of vertices ($\mathbb{V}$) in a graph (G) using the smallest length. SMT algorithm will calculate extra vertices called Steiner points (S) whose cost (c) is given as,

$$c(SMT(\mathbb{V})) = c(MST(\mathbb{V} \cup S))$$

where

$$c(MST(\mathbb{V} \cup S)) \leq c(MST(\mathbb{V}))$$

In our approach, the vertices $\mathbb{V}$ is a set containing the location of events and the location of base-station, BS and is given by,

$$\mathbb{V} = (E_i \cup BS),$$

where E_i denotes the set of k event locations $\{E_1, E_2, ...E_k\}$. Our approach utilizes a modified version of iterative-1 Steiner tree approximation algorithm [70] for calculating the Steiner points and it then routes the data using these points. For a given graph with $\mathbb{V}$ vertices, an optimal Steiner tree has at the most $|\mathbb{V}| - 2$ Steiner points [76]. The resultant number of Steiner points in iterative-1 approach may contain more than $|\mathbb{V} - 2|$ points, which can be eliminated later, based on the degree of the node [70].

Problem Statement

For a given set $\mathbb{V}$ with $k+1$ points, the aim is to determine a set of S points, such that the Minimum Spanning Tree over $(\mathbb{V}, S)$ points is minimum. Number of vertices in $\mathbb{V}$ is given as $|E_i| + |BS|$, where E_i denotes the set of events $E_1, E_2, E_3, ...E_k$ and BS indicates the location of the base-station. The aim is to find the optimal set of Steiner points over $\mathbb{V}$ denoted as $SMT(\mathbb{V})$, which will reduce the overall cost in routing the data from multiple event locations to the base-station. Once the Steiner point set (S) is identified, the corresponding coordinates are calculated with respect to each edge of the resulting tree (as explained in phase I). We follow phase II for assigning mobile nodes to connect multiple event locations with the base-station. Simulation results given in Section 3.1.3 show that modified iterative-1 approach will reduce the number of iterations and hence increase the speed of computation.

Given two point sets $\mathbb{V}$ and S, saving in-terms of cost can be shown as

$$\Delta MST(\mathbb{V}, S) = c(MST(\mathbb{V})) - c(MST(\mathbb{V} \cup S))$$

3.1.2.1 Steiner set Calculation

The proposed approach uses the set $\mathbb{V}$ to calculate the set of Steiner points S with the help of two candidate sets $\mathbb{P}$ and $\mathbb{P}'$. We assume that area A is virtually divided into a mesh with horizontal and vertical lines separated by a distance γ. Let $\mathbb{P}$ denote the candidate sets containing the intersecting points of line crossing the area. For an element in $\mathbb{P}$, a Steiner point at which $x \in \mathbb{P}$ that maximizes $\Delta(\mathbb{V}, \{x\}) > 0$. The aim is to find an element x from the candidate set which maximizes $\Delta MST(\mathbb{V} \cup S, \{x\}) > 0$. For calculating the near optimal solution, it is better to iterate all possible points in the area A, by dividing it into smaller unit area. By doing so, the computational complexity or the number of iterations is inversely proportional to the value of γ. For a small value of γ, the number of iterations will be large. In the proposed approach, we initially use a large value of γ to find the candidate set $\mathbb{P}$ and hence to find the element x, that maximizes the $\Delta MST(\mathbb{V} \cup S, \{x\}) > 0$, where initially $S = \emptyset$. Once the element x is identified, the process is repeated for a small value of γ (denoted as γ'), for an area ΔA to find the candidate set $\mathbb{P}'$. ΔA denotes a small region around the location x (whose co-ordinate is given as (p, q)). The area of ΔA is represented using the coordinates connecting $(p-\gamma, q-\gamma)$ and $(p+\gamma, q+\gamma)$. From the set $\mathbb{P}'$, an element x' is chosen which maximizes $\Delta MST(\mathbb{V} \cup S, \{x'\}) > 0$. The Steiner set S is then updated to $S = S \cup x'$. Repeat the above steps for $|\mathbb{V} - 2|$ times. The proposed algorithm is given in Algorithm 3.3. Figure 3.4 shows the scenario when mobile nodes are rearranged to connect multiple events to the base-station.

Algorithm 3.3 Algorithm for calculating SMT($\mathbb{V}$)

```
1: procedure SMT_POINTS($\mathbb{V} = \{E_1, E_2, ...E_k, BS\}, A, count = 0, S = \emptyset$
2:     Calculate candidate_set $|\mathbb{P}| = Cal\_points(A, \gamma)$
3:     Find $x$ from $|\mathbb{P}|$ which maximizes $\Delta(\mathbb{V} \cup S, \{x\}) > 0$
4:     Calculate candidate_set $|\mathbb{P}'| = cal\_points(\Delta A, \gamma')$, where $\gamma\prime << \gamma$
5:     Find $x'$ from $|\mathbb{P}'|$ which maximizes $\Delta(\mathbb{V} \cup S, \{x'\}) > 0$
6:     Update the Steiner set $S = S \cup \{x'\}$
7:     Repeat the steps 2 to 6 till $|S| = |\mathbb{V} - 2|$
8:     Return $S$
9: end procedure
```

3.1.3 Implementation

Test-bed implementation was done to check the performance empirically. colorblack The proposed event monitoring system uses multiple-communication protocol IEEE 802.11 for multimedia data communication, IEEE 802.15.4 for non multimedia communication which is not suitable for multimedia communication due to the low data rate, which is evident from Figure 3.5. Throughput results are shown with reference to a test-bed consisting of single event sending at a rate of 200 pps in the presence of other data sources. The static nodes used in the test-bed are micaZ motes. With a single channel, throughput decreases with increase in the number of sources, but by increasing the size of the payload in IEEE 802.15.4 a higher throughput can be achieved. When multiple sources are transmitted using orthogonal channels, the data rate achieved is almost constant. Figure 3.6 shows the throughput analysis for varying hop counts. It can be seen from the graph that even though the hop count increases, the PDR remains constant. From the figure, it is clear that the maximum data rate achieved is less than $140 Kbps$, which is in sufficient for transmitting multimedia data even if there is an increase in the number of hops.

Energy consumed in the network is proportional to the number of nodes in routing and the amount of communication overhead in the network. Simulations were done in Castalia to compare SMT approach with the other approaches such as MST and direct routing (shortest path routing), by varying the number of sources and the size of the deployment area to calculate the number of hops required for data communication. Simulations were done for an area of $1km \times 1km$ with multiple sources and relay nodes

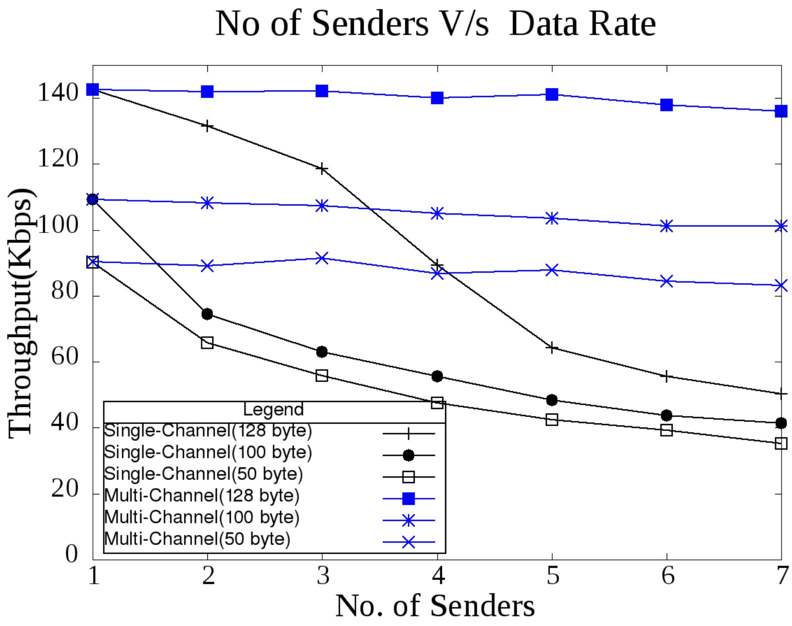

Figure 3.5: Performance of Single sender in the presence of other senders (Single Channel)

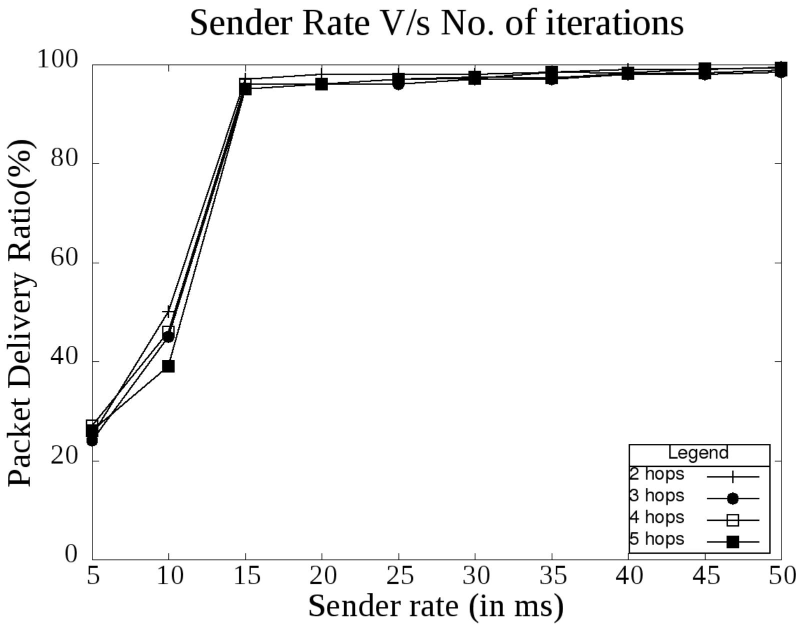

Figure 3.6: Throughput Analysis with single sender: Multiple Channel

deployed randomly. Figure 3.7 plots the number of sources vs hop count with 50 nodes in the sensor field. SMT requires lesser number of hops when compared to MST and direct routing. This is due to the presence of Steiner points, which in turn minimizes the requirement of extra hops. Figure 3.8 shows the hop count vs number of nodes in

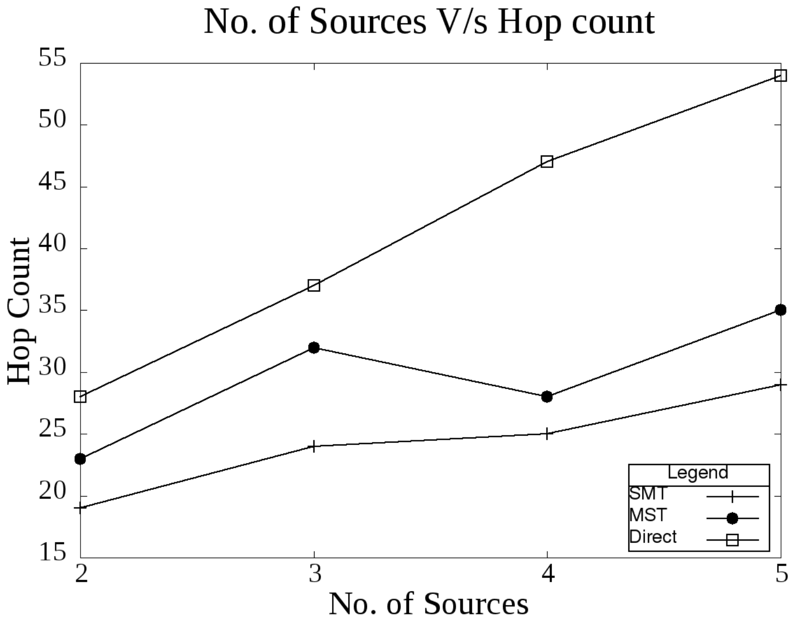

FIGURE 3.7: Comparison of MST, SMT and Direct method: Hop count Vs No.of Sources

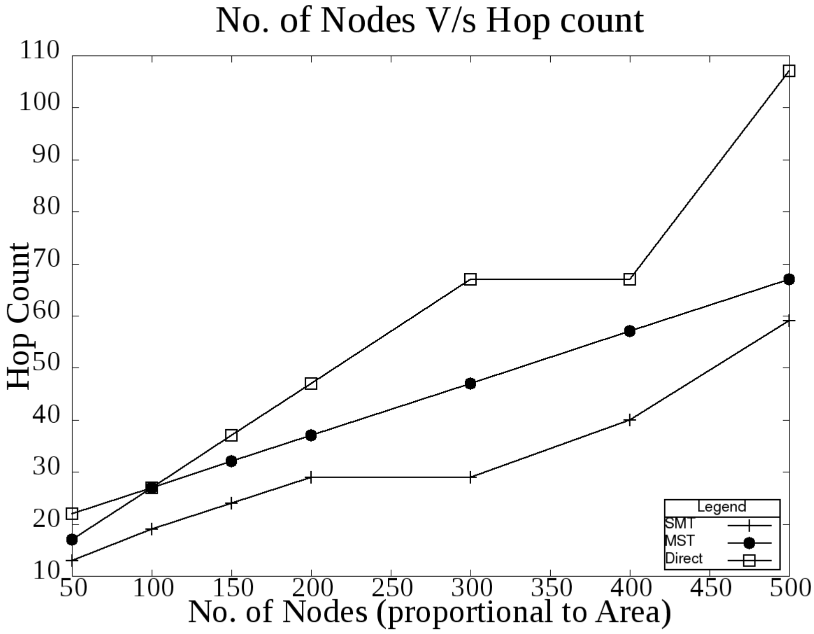

FIGURE 3.8: Comparison of MST, SMT and Direct method: Hop count Vs No.of Nodes

the field with three sources simultaneously sending data. In this scenario also, SMT performs better than its counter part.

Simulations were also done to check the performance of Modified-Iterative SMT (MI-SMT) with Iterative-1 SMT approach. For comparing the result, three sources were

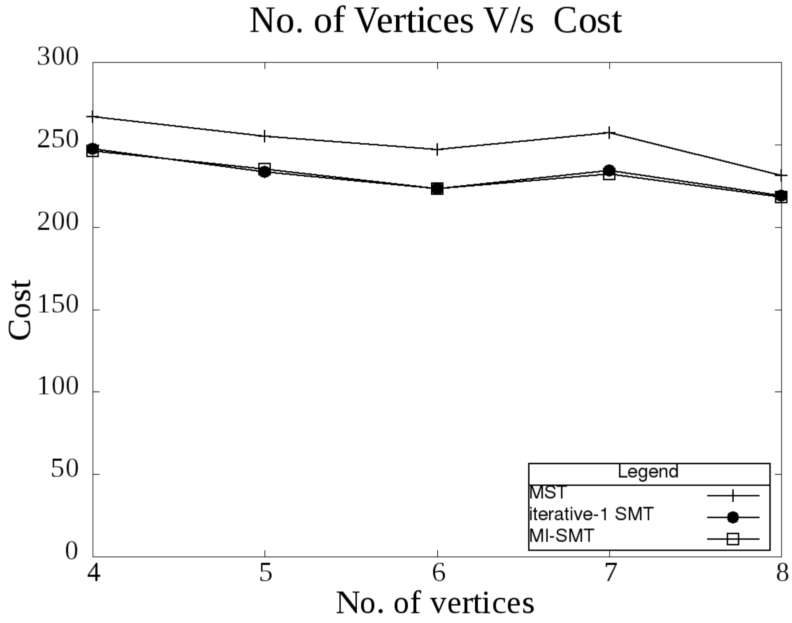

Figure 3.9: Cost Vs No. of vertices

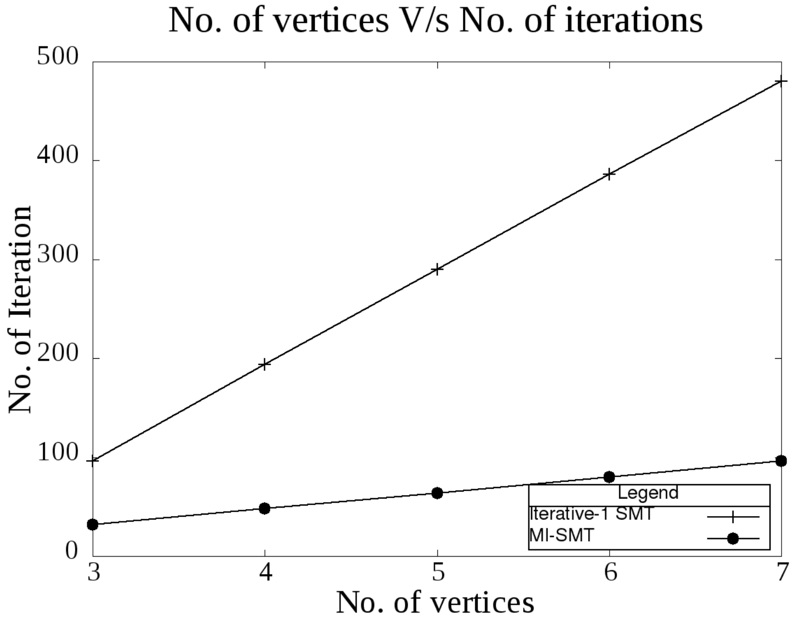

Figure 3.10: No. of Iteration Vs vertices

programmed to send the data to the base-station. Node deployment followed for simulation was random. Results shown in Figures 3.9 and 3.10 indicate that the number of iterations needed for MI-SMT is less as compared to Iterative-1 SMT algorithm, without compromising on the SMT calculation. From the above results, it is clear that SMT guarantees the use of minimum number of mobile nodes with reduced hop count as compared to MST and direct routing. Modified iterative-1 algorithm reduces the number of comparisons without compromising on the Steiner point calculation.

Test-bed: For further proof of concept of the proposed algorithm, a WSN test-bed consisting of Micaz static sensor nodes and custom made mobile robots termed as B-Bot are used. The details of B-Bot architecture are provided in Appendix A.

Static node deployment pattern: A grid based deployment was used for static nodes. All static nodes were pre-coded with their locations corresponding to Cartesian coordinates in a 2-D plane. The role of static sensor node includes

- Sensing an event and reporting it to the base-station.
- Routing control messages between mobile nodes and base-station.

The accuracy of the mobile node movement is dependent on the localization algorithm used. In our implementation, Received Signal Strength Indicator (RSSI) based weighted trilateration was used to calculate the current position of the B-Bot. Events have been emulated events using TelosB mote placed in the test-bed, which will broadcast a packet every 5 seconds. Neighboring static nodes after receiving the emulated event packet trilaterate the location of the event. We have followed the approach proposed in Appendix B for trilateration. The location of the event is then communicated to the base-station. On receiving the event location, the base-station does the following

- Calculates the Euclidean distance between the base-station and the event location
- Determines the coordinates between the event location and the base-station to place the mobile nodes (Phase I)
- Executes the modified Hungarian algorithm for optimal placement of mobile nodes (Phase II)

In case of multiple events, the base-station calculates the Steiner tree connecting all the event locations and then uses the modified Hungarian algorithm for optimal placement of mobile nodes.

Experimental Set-up: Telosb static nodes were deployed on a grid of size $5nodes \times 5nodes$ separated by a distance of 2.5 $meters$. Nodes were programmed to communicate at minimum power levels using $setpower()$ function available in TinyOS.

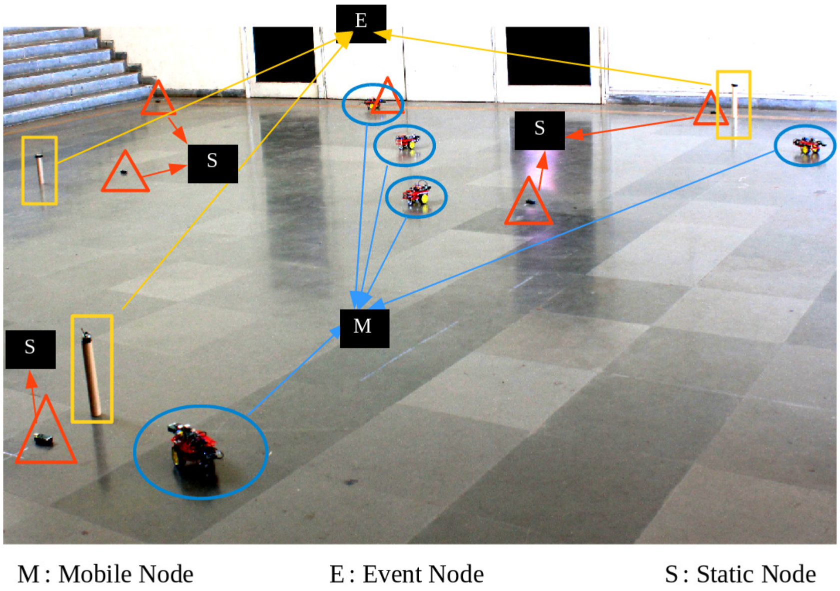

Figure 3.11: Test-bed: Multi-Event Monitoring

B-Bots were placed in random positions in the sensor network. Event nodes (three micaZ motes) were placed in the testbed at random locations $((X_E, Y_E))$, were $((X_E, Y_E))$ denote the set of location of event nodes. Static nodes were used to route the position of mobile nodes as well as the event location to the base-station. The algorithms propounded in Section 3.1.2 were implemented at the base-station. The base-station instructs the selected mobile nodes to move to the calculated location. Experiments were conducted for different values of (X_E, Y_E), with varying random initial position of mobile nodes with the same grid size. A picture of the deployment is shown in Figure 3.11. As show in 3.11, **M** stands for mobile nodes, **S** stands for static nodes and **E** represents events. Average time for the mobile nodes to move from the current location to the new location after getting instruction from the base-station is around 7.5 minutes.

3.2 Area Monitoring

Monitoring a given area is termed as area monitoring in this thesis. Area monitoring in WSN is used for a multitude of applications from data collection to network maintenance functions such as network diagnostics and health monitoring, localization, residual energy scanning, partition discovery, topology discovery and global reprogramming [77]. Applications such as In-Application Programming (IAP) to reprogram these sensor nodes using a multi-hop route from the base-station is not a feasible solution due to bandwidth limitations. One possible method to achieve this is by using mobile nodes to explore the network area. These mobile nodes can further establish a direct connection with the static neighbor nodes while they are mobile.

The methodology of using mobile nodes to explore an area is not novel. Data mules are employed to gather data from static nodes and deliver it to the base-station [49]. Use of one or more mobile sinks to collect the data from static nodes was proposed in [78] [79]. Mobile anchor nodes *(mobile beacon)* were extensively used in static network localization [80]. In most literature available, it has been assumed that the topology of the network is known in advance and hence proper path planning for mobile nodes could be made [25].

The objective in using mobile nodes to explore the sensor field is to ensure that every node is visited, with maximum energy efficiency and in minimum time. In the case of random deployment or node displacement due to environmental calamities, nodes need to be localized with their changed position. A node localization scheme is proposed in [80], where mobile anchor nodes should reach each static node in the area and localize it. Similarly, partition detection using mobile nodes was proposed in [31], to detect partition in a segmented network.

Since most of the applications are real time, there is a need for fast exploration. Compared to a single mobile node based exploration, multiple mobile node based exploration provides better reliability, efficiency and flexibility. Multiple mobile nodes can be programmed to collaborate and monitor previously unmapped areas quickly and simultaneously. Cooperation between mobile nodes require exchanging of location information between each other in order to plan an optimal exploration of a given area. Advanced path planning for mobile nodes is not feasible for an unknown area with obstacles.

3.2.1 Background

The objective of using mobile nodes in area monitoring is to have mobile nodes touch every region of the sensor network at least once. The movement is directed in a manner such that energy consumption and latency are minimum.

The main focus of area monitoring using mobile nodes is the mobility pattern and inter-node communication. In practice, communication may be limited or unavailable. Arkin and Diaz assume that the mobile nodes are in line-of-sight [81]. According to Matthew et.al., [82], mobile nodes choose their direction, in a way that one node is always in line-of-sight. Communication range limitations are addressed in [83] [84], where a mobile node is always in the range of another.

Out of several cooperative strategies for determining the mobility pattern, the most widely accepted frontier-based [85], market-driven [83] and role-based approaches [84]. In frontier-based method, mobile nodes move towards the boundary between discovered and unknown area. Frontiers are regions on the boundary between discovered and unknown area. In [86], multiple mobile nodes (relays) choose frontiers based on proximity and then move towards it.

The main advantage of this method is that it can be used in both large open spaces and narrow cluttered spaces. This approach groups adjacent cells into frontier areas. Since there is no explicit coordination between mobile nodes, the nodes may end up covering the same area and may even physically collide with one another. [85] solves the problem of exploring the same area by considering a trade-off between the cost of reaching a particular frontier and the gain in estimated spatial information. This is achieved by adjusting the utility value of each frontier cell based on the location of frontier cells previously assigned to the other relay nodes.

In the market-driven method, mobile nodes place bids on submission, based on traveling costs and expected information gain. The Role-based model proposed in [84], categorizes mobile nodes into explorer nodes and relay nodes in order to establish a multi-hop communication with the base-station. The speed at which area coverage is achieved in frontier based approach is higher than market-driven and role-based exploration strategies.

In this section, we present two types of sensor field area coverage, *i) Hybrid Method which is non-deterministic and ii) Max-Gain Method which is deterministic*

Problem definition with Assumption: The modeling of area coverage using mobile nodes is as follows: Sensor field is represented as a convex polygon with an area A. Static sensor nodes deployed in the field are represented by using the set k. A set of N mobile nodes are deployed in the area A. It is assumed that all the mobile nodes can determine their location and can navigate freely in the area. The aim is to cover the area A with N mobile nodes efficiently in minimum time and with minimum communication overhead.

Communication: We have assumed that the base-station can reach all mobile nodes and vice-versa using a long range antenna and wireless communication module.

There are many technologies available for long range communication in literature such as: usage of satellite transceiver [87], long range IEEE 802.11 transceiver [15], cellular based communication module [88] and directional antennas [80]. We propose to use multiple-communication protocols described in Section 3.1. The mobile nodes activate long range transceivers only when they need to communicate with the base-station.

3.2.2 Hybrid Method

3.2.2.1 Motivation

Simulations were done to study the comparison of frontier approach with other random mobility models. Simulations were done openCV (Open Source Computer Vision) for an area of 750 × 750 with 6 mobile nodes which were initially placed in the center of the area. We compared the simulation time with the percentage of area covered for simulation. A comparison of random exploration techniques using frontier-based method is shown in Figure 3.12. It can be observed from Figure 3.12 that the initial area coverage of *random-direction* model is higher when compared to other methods. This is because in the random direction model, the mobile nodes choose a direction and move in that direction until they reach the boundary, which results in higher spatial gain during the initial stages of coverage. It is also clear from the Figure 3.12 that the *frontier-based* model converges faster than other models as it is deterministic.

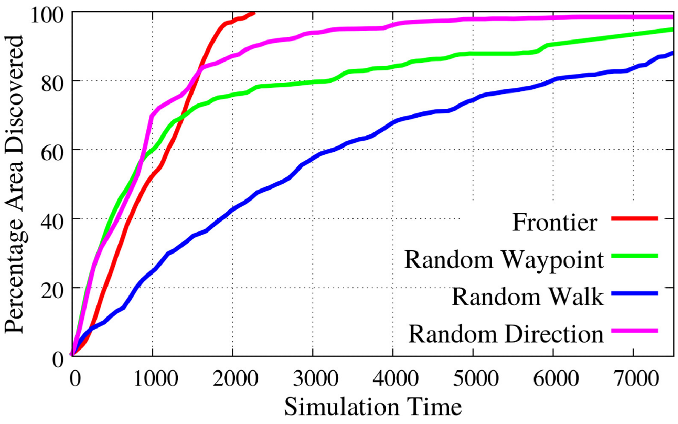

FIGURE 3.12: Comparison between Random mobility models Vs Frontier approach

A hybrid approach which makes use of *random-direction* and *frontier-based* approach for area coverage is proposed. The algorithm initially uses *random-direction* model as its initial speed at which an area is covered is higher than other methods. Once the percentage of area covered reaches the input threshold t_p, the algorithm switches to frontier based method since its convergence characteristic is better compared to other techniques. The initial exploration nature of Hybrid approach is non-deterministic because of its randomness which later switches to the frontier based method until the entire sensing area has been covered.

3.2.2.2 Proposed Algorithm

From Figure 3.12, it can be seen that the frontier-based method [85] covers the given area quickly, while the other methods take longer time to achieve coverage. It could also be seen that the initial area coverage is faster in *random-direction* model when compared to the *frontier-based* method. Therefore, we have developed a hybrid method that uses the advantage of both frontier-based exploration model and random directional model. Table 3.2 lists the notations used in our approach.

TABLE 3.2: Notations used: Hybrid Approach

Parameter	Description
N	Number of mobile nodes.
A	Area to explore.
$M = [a_{i,j}]_{l \times b}$	Map Matrix of size $l \times b$ representing an Area A with an initial value WHITE $\forall a_{i,j}$, where $a_{i,j}$ represents a pixel in the given area. $a_{i,j}$ is set as BLACK when it is explored.
t_p	Maximum threshold value to run *random-direction* approach
A_k	Area explored by the mobile node at any point of time
CV	Coverage value, ratio of A_k by A

Proposed Hybrid method is explained in Algorithm 3.4. The inputs required are: the threshold t_p, map matrix M, area that has to be covered, A and the number of mobile nodes N available. The initial value of all $a_{i,j}$ of map matrix M is set to zero, which is represented as WHITE. Depending on the location and communication range of nodes in the set A_k, the value in matrix M will be updated to BLACK. Steps 5 to 12 of the algorithm implements the *random-direction* model [66] for mobile node movement. Each mobile node N_i starts moving from its current position (X_j, Y_j) in a random direction θ_i, till it reaches the network boundary.

Algorithm 3.4 The algorithm for Area Coverage: Hybrid Approach

```
 1: procedure HYBRID_EXPLORATION($N$, $M$, $t_p$, $A$, $A_K \leftarrow 0$)
 2:     $CV \leftarrow 0$
 3:     while $CV < t_p$ do
 4:         $CV \leftarrow \frac{A_k}{A}$
 5:         for all $N_i$ do
 6:             move $N_i$ to a rand($\theta_i$)
 7:             if $N_i$ reaches the boundary of $A$ then
 8:                 update $M$ with the area covered by the $N_i$
 9:                 calculate $A_k$ from $M$
10:                 $CV \leftarrow \frac{A_k}{A}$
11:             end if
12:         end for
13:     end while
14:     while $A < A_k$ do
15:         for all $N_i$ do move $N_i$ to best frontier w.r.t $M$
16:             if $N_i$ reaches its target frontier then
17:                 update $M$ with the area covered by the $N_i$
18:                 calculate $A_k$ from $M$
19:             end if
20:         end for
21:     end while
22: end procedure
```

A mobile node updates the base-station, when it reaches the boundary. The mobile node sends its current location to the base-station using the long range transceiver module. The information communicated to the base-station includes the initial position (X_j, Y_j) and its current position. The base-station uses this information to update the map matrix M and it also updates the variable A_K. If the mobile node reaches the boundary, it communicates its last initial position to the base-station. Each mobile node then repeats steps 5 to 12 till it reaches the value of $CV < t_p$, where $CV \leftarrow \frac{A_k}{A}$. CV is computed at the base-station.

When CV exceeds the t_p value, the base-station instructs all mobile nodes to switch to *frontier-based* method. In the *frontier-based* model, base-station controls the movement of mobile nodes. It instructs the nodes to move towards the closest frontier, based on the utility value. The base-station updates the map matrix M and then instructs every mobile node to move towards a calculated frontier. Node mobility in the frontier model is decided based on the area covered and the movement of mobile relay nodes. Simulation results discussed in Section 3.2.2.3 show that the update intervals required for hybrid model is lesser than frontier model.

3.2.2.3 Simulation Results: Hybrid Method

Simulations were performed to evaluate the performance of the proposed protocol against the *frontier-based* model and *random-directional* model. The simulation was done using static nodes randomly deployed in an area of $500m \times 500m$. The base-station was located at the extreme end of the the simulated area. All the mobile nodes were initially placed at a single location. Simulations were performed by varying the range of static nodes and the number of mobile nodes. Simulation results show that the proposed approach performs better, in terms of the speed at which the area is covered and its communication overhead.

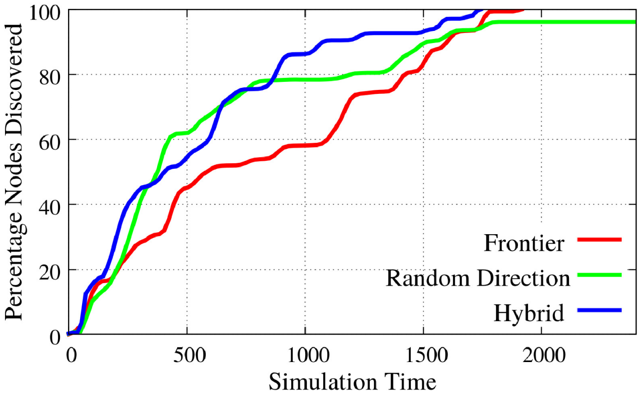

FIGURE 3.13: Area exploration with 4 mobile nodes

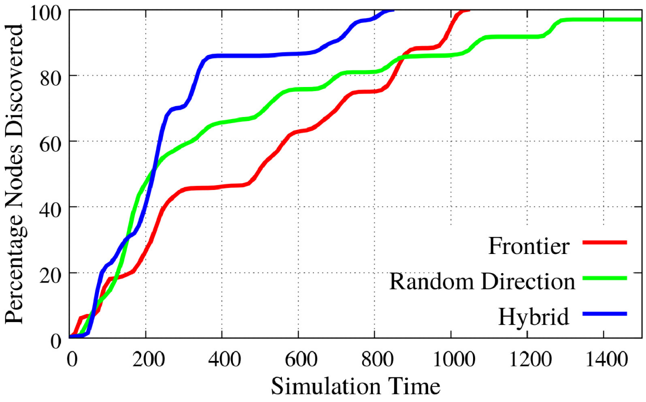

FIGURE 3.14: Area exploration with 7 mobile nodes

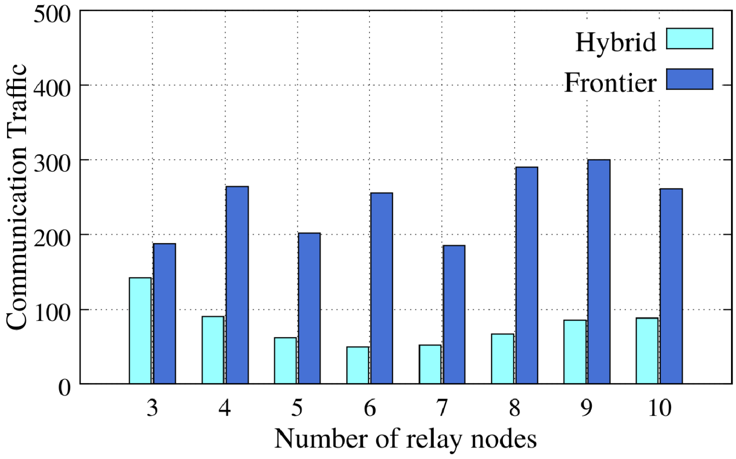

FIGURE 3.15: No. of Control packets Vs No. of mobile nodes

Figures 3.13 and 3.14 shows the node discovery with respect to the simulation time for an area of $500m \times 500m$. Transmission range of mobile nodes were set to $40m$. Initial placement of mobile nodes was in proximity of the base-station. The simulations were done upto a time at which all static nodes in the network were discovered. Comparison of the proposed method was done with *frontier-based* and *random-direction* model using four mobile relays as shown in Figure 3.13. Figure 3.14 shows the comparative performance with seven mobile nodes. The simulation was done with a threshold value of 0.8. It is clear from Figure 3.13 and Figure 3.14 that the initial performance of the proposed method was the same as the *random-direction* model. On reaching threshold, the mobile nodes switched to *frontier-based* model, that converges faster as compared to the *random-direction* model.

The communication overhead in the proposed approach was compared with *frontier-based* model and is shown in Figure 3.15. Figure 3.15 is an average over fifty iterations while varying the number of mobile nodes used for exploration with respect to the communication traffic. The comparison was made with respect to the total number of messages exchanged between the mobile nodes and the base-station. It can be concluded from Figure 3.15 that the communication overhead in the *frontier-based* model is almost twice compared to hybrid method proposed. The proposed method does a quicker area coverage when compared to *random-direction* method with less communication overhead compared to the *frontier-based* method. The communication over head is because of the frontier approach used in the proposed method. For each small movement, the node send an update message to the base-station.

3.2.3 The Max-Gain method

The Max-Gain method proposed in this section is a deterministic model for covering a given area using mobile nodes. The method proposed here considers each point in an unexplored area as a possible destination for a mobile node. For every mobile node, the most cost effective destinations were calculated based on the area already covered, the area yet to be covered and the *utility* value. The *utility* value is calculated based on the destinations previously assigned to the other mobile nodes. A map of the entire field is maintained by the base-station and is used to calculate future movement of each mobile node. The base-station determines the possible destinations considering the area covered and the area yet to be covered. It then directs the mobile node to the calculated destination. Here we assume that the boundary for area coverage is known in advance.

3.2.3.1 The Proposed Algorithm

The *Max-Gain* exploration algorithm proposed in this thesis addresses coverage issues using a trade-off between the cost of reaching an unmapped area and the utility value of the same with respect to other mobile nodes. The proposed algorithm is centralized since almost all decisions are made at the base-station. The base-station maintains a *map matrix*, M which is similar to the occupancy grid in [85] representing the area to be discovered. The *map matrix* represents the rectangular area R where the polygon P_c can be fit in. The *map matrix* M is represented as $M = [a_{i,j}]_{l \times b}$, where l and b represent the length and breadth of the rectangle respectively. The unit area $a_{i,j}$, which henceforth be referred to as *cell*, takes the value WHITE, BLACK or GREY. If a cell is unexplored then it is represented with a WHITE color in the matrix M. If the cell is already explored then it is represented in BLACK. Cells present in the path between the current mobile node position to destination are marked as GREY. As the mobile node starts moving, obstacles are discovered and this information is transmitted to the base-station. Once the mobile discovers a region, the color value in the path changes to BLACK.

TABLE 3.3: Parameters used: MAx-Gain Approach

Parameter	Description
N	Number of mobile nodes.
P_c	Represent the sensor field region.
R	Rectangular region where polygon region P_c can fit in.
M	Map Matrix of Area R represented as $M = [a_{i,j}]_{l \times b}$.
T	Total number of cells in the area R.
N_i	Represents i^{th} mobile nodes.
S'_{N_i}	Represents the set of mobile nodes except node N_i.
C_j	Represent a cell j in the region R.
C_{N_i}	Represent a cell assigned to a mobile node N_i.
$V(C_j)$	Value of a cell C_j. Value can be WHITE, BLACK or GREY.
a	Represent the area of a cell.
$\Re_i$	Communication radius of i^{th} mobile node.
C_i	Destination Cell for a mobile node N_i.
$P(x, y)$	Coverage path between mobile nodes x and y.
$dist(k, j)$	Euclidean distance between points k and j.

In this algorithm, the cost of reaching the cell is proportional to the distance between the mobile node and the destination cell. The utility of the cell depends on the density of WHITE cells between the mobile node and itself, as well as the number of other mobile nodes that are moving towards the cell or to a geographical location close to the cell. If a mobile node is assigned to a cell, then the path between the current position of the mobile node and the cell should also be considered while calculating the utility. Other mobile nodes are assigned the minimum preference for utilizing/crossing this path.

If a mobile node N_i is assigned to a cell C_j then the mobility path of the mobile node between the N_i and C_j, denoted as $P(N_i, C_j)$ is marked in GREY. This is used for assigning a low utility value to any cell in that path. The algorithm proposed differs from the *frontier-based* method in the utility function calculation. The *Max-Gain* algorithm is described below. Table 3.3 lists the notations used in the proposed algorithm.

Algorithm 3.5 explains the max-gain approach. The Algorithm 3.5 divides the entire area into cells of unit area a. Line 2 of the algorithm calculates the total number of cells

Algorithm 3.5 Max-Gain exploration algorithm

```
 1: procedure MAX_GAIN_EXPLORATION($P_c$, $N$, $R$, $a$, $\Re$)
 2:     $T \leftarrow \frac{R}{a}$
 3:     for all $N_i$ do
 4:         calculate a destination cell $C_i$
 5:         for all $C_j$ do
 6:             if $V(C_j)$ = WHITE then
 7:                 $U_w \leftarrow$ Count of WHITE Cells in $P(N_i, C_j)$
 8:                 $U_t \leftarrow \alpha.(U_w) - \sum_{k=1}^{S'_{N_i}} .U_d$
 9:                 $\kappa_{i,j} \leftarrow U_t - \beta.dist(N_i, C_j)$
10:                 $C_i \leftarrow \max(\kappa_{i,j}), \forall C_j$
11:             end if
12:         end for
13:         Mark $P(N_i, C_i)$ as GREY in $M$
14:         Set destination of $N_i$ to $C_i$
15:     end for
16: end procedure
```

T which is obtained by dividing the area R by a. Lines 3 to 10 calculates the destination cell for each mobile node N_i. For calculating the destination cell, the cost of each cell needs to be calculated which in turn depends on the cost of reaching the cell and its utility value. For each cell C_j, utility value is calculated with respect to parameters U_w and U_d, where U_w denotes the number of white cells in $P(N_i, C_j)$ and U_d is the utility in terms of the distance of other mobile nodes w.r.t to the given cell C_j.

U_d is determined using equation 3.8

$$U_d = \begin{cases} (1 - \frac{dist(C_{N_k}, C_j)}{\Re}), & \text{if } dist(C_{N_k}, C_j) < \Re \\ 0, \text{otherwise} \end{cases} \tag{3.8}$$

In equation 3.8, $dist(C_{N_k}, C_j)$ denote the distance between the mobile node N_K and the target cell. If the $dist(C_{N_k}, C_j)$ value is greater than the communication radius, then U_d will be calculated as per the equation, else the value will be zero. Line 8 of the algorithm computes the total utility value U_t of cell C_j as,

$$U_t = \alpha U_w - \sum_{k=1}^{S'_{N_i}} U_d \tag{3.9}$$

The cost of cell C_j with respect to the mobile node N_i is the difference between $dist(N_i, C_j)$ and the calculated U_t which is given by

$$\kappa_{i,j} = U_t - \beta dist(N_i, C_j) \tag{3.10}$$

where α and β are constants and $\kappa_{i,j}$ denotes the cost of cell C_j with respect to mobile node N_i. The destination cell of mobile node N_i, denoted by C_i, is determined using the following equation

$$C_i = \max(\kappa_{i,j}), \forall C_j \tag{3.11}$$

Line 13 of the algorithm marks all the cells in the path $P(N_i, C_i)$ as GREY. A control message is sent from the base-station to move the mobile node N_i towards the cell C_i. This process will continue till the entire area is covered. When a mobile node N_i reaches the destination C_i, the base-station updates the cells in the coverage path $P(N_i, C_i)$ to BLACK.

3.2.3.2 Performance Evaluation of Max-Gain algorithm

Simulations were done to evaluate the performance of the proposed model against the *frontier-based* model. The simulation was done using static nodes randomly deployed in an area of $500m \times 500m$. Mobile nodes were initially placed in a single location. Simulations were conducted by varying the range of mobile nodes and the number of mobile nodes.

Figures 3.16 and 3.17 depict the percentage of the area discovered with respect to the simulation time in an area of $500m \times 500m$. The transmission ranges of mobile nodes

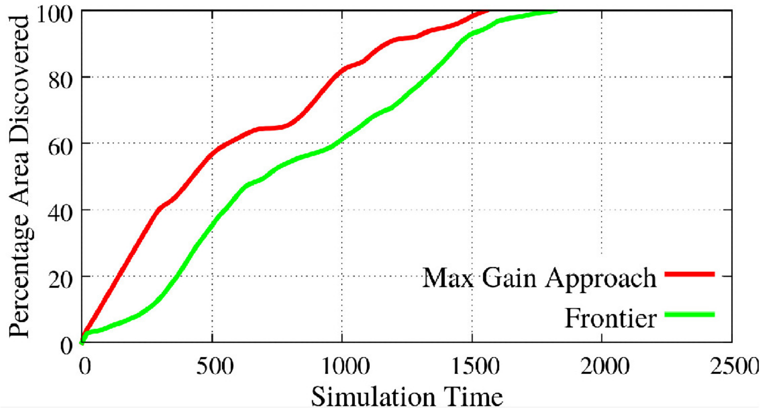

FIGURE 3.16: Area Exploration: With 4 mobile nodes

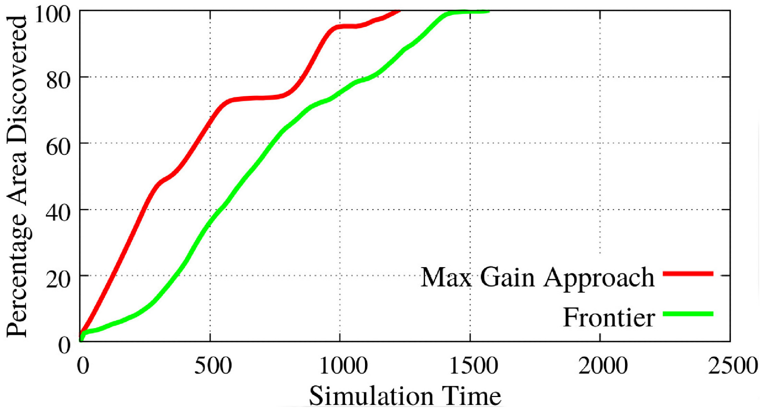

FIGURE 3.17: Area Exploration: With 6 mobile nodes

and the static sensor nodes were set to $30m$. The simulation was continued until the entire area, including all obstacles in the network were detected. A comparison of the proposed method was done with *frontier-based* method using four mobile nodes. The results of this simulation are shown in Figure 3.16. Also, Figure 3.17 shows the comparison using six mobile nodes. Simulation was done by fixing the value of α and β to 1. It is clear from Figures 3.16 and 3.17 that our approach performs better compared to *frontier-based* exploration.

We observe that the speed at which the area is covered is much higher for *Max-Gain* method when compared to *Frontier method*. This is primarily due to the fact that the

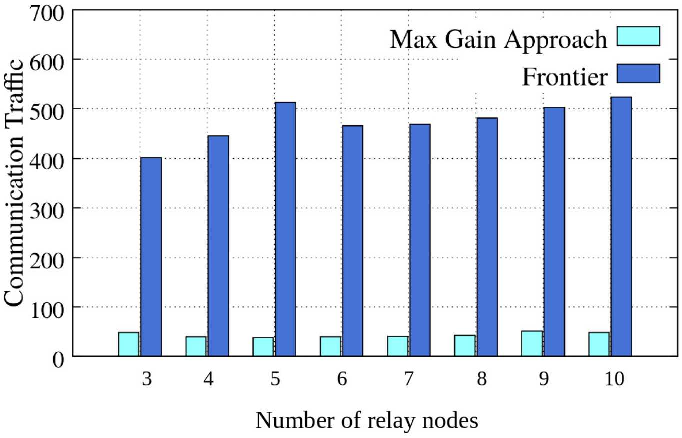

FIGURE 3.18: No.of control packets Vs No.of Mobile nodes

proposed algorithm selects a destination for each mobile node such that the spatial information gain is maximum. In the *frontier-method*, a destination cell is selected from the set of all *frontier cells*, i.e., cells on the boundary between explored and unexplored area, which has the highest utility value. Unlike frontier approach, the proposed method selects a destination cell from the set of all cells in the given area, which offers the maximum possible spatial information gain.

Communication overhead of the proposed approach was considerably less compared to *frontier-based* model. The result of the simulation is shown in Figure 3.18. The comparison was made with respect to the total number of messages exchanged between mobile nodes and the base-station during area coverage. Reduction in communication overhead in *Max-Gain* approach is due to the fact that each mobile node is directed to a destination which offers the maximum spatial gain. The time spent on covering an unexplored area during each iteration is high. This essentially decreases the number of decision-making steps at the base-station, leading to lesser number of interactions between the base-station and the mobile nodes.

It can be concluded from Figure 3.18 that the communication overhead in the proposed algorithm is less when compared to *frontier-based* approach. This reduction in communication overhead is ensured by selecting a cell which is at a longer distance in an unmapped region, rather than selecting the nearest frontier cell.

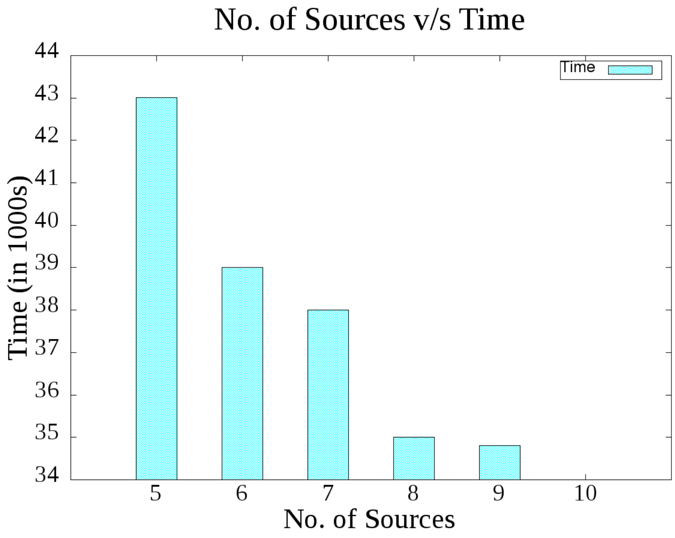

FIGURE 3.19: Effect on increase in node number

Figure 3.19 shows the effect on increase in the number of mobile node. Simulation was done in open-CV for an area of 750 × 750. Mobile nodes are programmed to move as per max-gain approach. It is clear from Figure 3.19 that with the increase in the number of mobile nodes, the time taken to explore the area reduces.From 5 mobile nodes to 10 mobile nodes, the time taken will reduce from 43K seconds to 34K seconds.

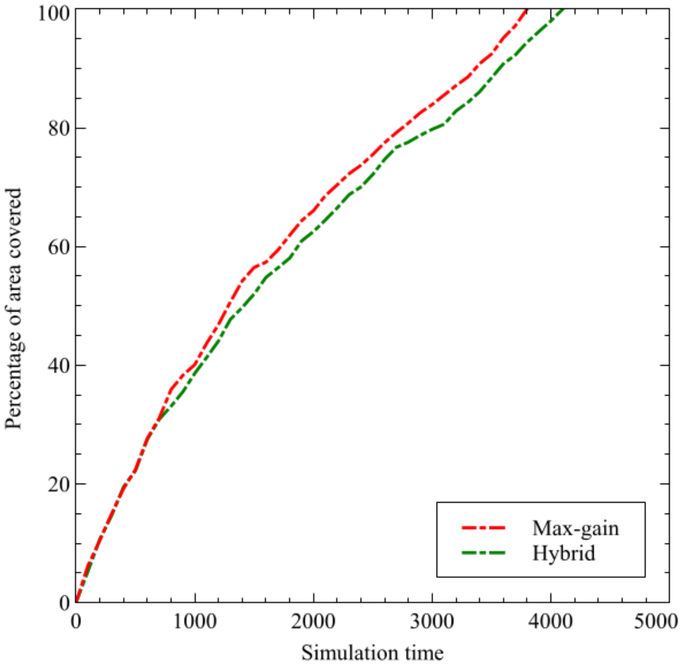

FIGURE 3.20: Comparison of Hybrid with Max-gain Approach

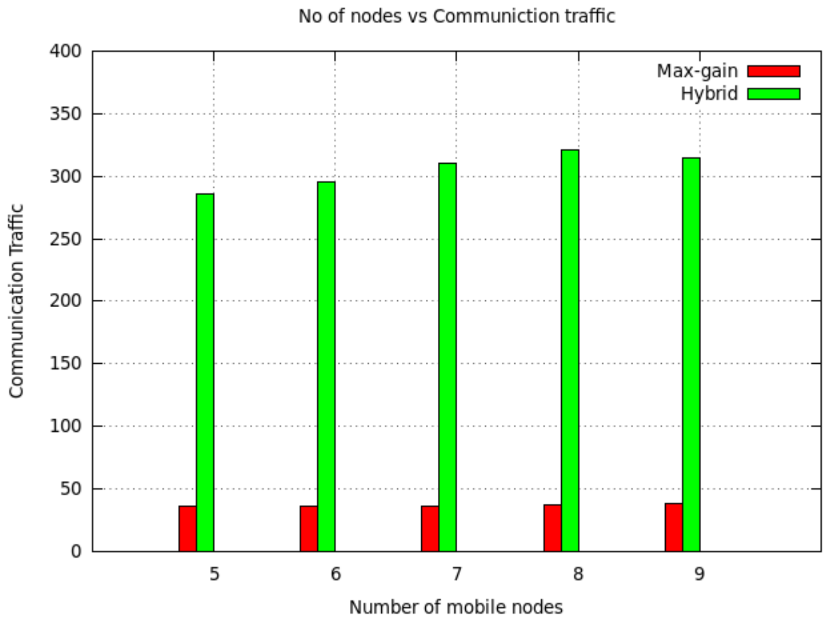

FIGURE 3.21: Comparison of Hybrid with Max-gain Approach

A comparison of hybrid with max-gain approach is done and is shown in Figure 3.20 and 3.21. From both the Figures it is clear that Max-Gain performs better when compared to hybrid approach. This is because of the deterministic nature of Max-gain approach.

Obstacle detection: One of the practical issues in the exploration with mobile robots is the detection and the avoidance of obstacles. We have used edge detection to detect the boundaries of the obstacle along with the Virtual Force Field (VFF) algorithm [89] to steer the mobile node away from the obstacle. We adopt the approach by Johann and Koren to identify the boundaries of an obstacle [90]. The mobile node transmits the data to the base-station. The base-station reconstructs the boundaries of the obstacle and marks the corresponding cell in M to BLACK. Simulations were done with obstacle (of size 10% of the total area) of polygon structure placed randomly in the simulation area.

3.3 Partition Connectivity Restoration

Every sensor node has three main features: sensing, processing and communication. Maximum energy is consumed for communication tasks. Nodes that act as a gateway get depleted of energy earlier than other nodes in the network.

Hardware failure can also occur due to corrosion or environmental calamities *(e.g., explosion, fire, landslide etc.)*. Most of the traffic in sensor network is converge-cast [91]. According to Li et.al., nodes which are closer to the base-station handle most of the traffic and hence are depleted of energy within a short period of time. This decreases the lifetime of the network due to funneling effect [42]. If a node is serving as a *cut-vertices* node, [92] failure of this node may disrupt the communication between some of the sensor nodes and the sink. Since sensor nodes are programmed to operate autonomously and collaborate with each other without direct human intervention, a node failure may affect the entire network, by creating partitions within the network. In recent literature, researchers have proposed the use of mobile sensor nodes to solve the issues pertaining to funneling, so as to improve connectivity and coverage.

A redundant deployment is one way to reduce partitioning and thereby increase the network lifetime. This approach is neither economical nor feasible if node failure is due to environmental calamities. Another approach is to replace a dead node with a new one [30]. Doing this manually is not viable as some of the application environments are harsh or remote.

Given that WSNs are used for unmanned operation, the network needs to self-configure after the partition detection and restore the connectivity in the network. A possible alternative is to use mobile WSN that can reorganize themselves for better coverage and connectivity. In most of the work it is assumed that the initial topology of the network is known in advance and hence proper path planning for mobile nodes can be made. Partition discovery in a given area using mobile relay nodes was proposed in [93]. Since sensor nodes are randomly deployed, the nodes are disconnected from the base-station making it impossible to know their location. It is required that the base-station knows the number of partitions, the geographical position of each partition and the area affected to take decision on relay node placement. Dini et.al., assumes that a Partition Detection System (PDS) runs on the base-station, which can detect the presence of partition and can also provide the rough estimate of its position on disconnectivity [31].

Most approaches for connectivity restoration do not focus on monitoring the entire area of the network. Sookyoung and Mohamed propose to deploy mobile nodes along with static node initially [93]. A magnetic repulsive force based approach is proposed in [27] to find the partition. The Partition detection methods proposed in the above literature take considerable amount of time, or assume that a large number of nodes are available in the network for discovering the partition. In practice, the network partitions need to be discovered real-time and based on the number of partition and their geographical position, the mobile nodes need to be placed optimally to reconnect the partitioned area.

In this section the focus has been on partition discovery and connectivity restoration using mobile nodes. The proposed approach is explained in two phases. In Phase I, any of the methods described in Section 3.2 for area can be used. In this Section, the partition discovery phase is done using the hybrid method proposed in Section 3.2.2.

Phase II of the algorithm computes a set of Steiner points where the mobile relays need to be placed. We have adopted the Steiner approximation approach proposed in Section 3.1.2 for calculating Steiner vertices. Once the Steiner points are computed, optimal assignment of mobile nodes to these coordinates is done. We have used the algorithm proposed in Section 3.1.1 for assigning mobile node to the corresponding coordinates.

Problem Definition with Assumptions: The connectivity restoration problem can be modeled as follows: Sensor field is represented as a convex polygon with an area A. Static sensor nodes deployed in the field are represented by using a set of k network partitions $P_1, P_2, P_3..., P_k$ where each P_i has one or more static nodes. We assume that the existing static nodes are aware of their location. For *partition discovery*, a set of m mobile nodes are deployed in the area A. It is assumed that all the mobile nodes can determine their location and can navigate freely in the area without any obstacles. The partitions discovered after exploration is modeled as convex hull which is represented as $C_1, C_2, C_3..., C_k$. Connectivity restoration of a partitioned network can be defined as:

Finding a topology to connect all the discovered partition C_i using a set of n mobile nodes.

The objective is to restore connectivity using a minimum number of mobile nodes and to determine their trajectories and final destination. Additional mobile nodes will be deployed if $n > m$. The optimal number of mobile nodes that are required and their target positions are obtained using an approximation algorithm for MST. Placement of mobile nodes is carried out by using a variant of Hungarian algorithm.

3.3.1 Proposed Algorithm for partition discovery

The algorithm is divided into multiple phases based on the sequence of operations that take place in the network, starting from the detection of network dis-connectivity to the optimal placement of mobile nodes by the base-station. The primary objective is to detect all the k partitions $P_1, P_2, P_3, ..., P_K$ in the area, (*Partition Discovery* phase). The Second phase does the reformation of the network topology by reconnecting the partitions. The final phase does the *optimal placement* of mobile nodes.

The partition discovery phase involves discovering the partitions and forwarding of this to base-station. Here, we used hybrid method for partition detection. The hybrid method was selected due to its low complexity operations. For partition discovery each static node in the network maintains a *discovery variable*, dv. Initially this value will be same for all nodes in the network. When the base-station detects a partition it broadcasts a new *discovery variable* dv_{new} into the network. Mobile nodes are also updated with dv_{new}. Sensor nodes that are connected to base-station will have the updated dv_{new} value, while the nodes which are in the partitioned area retain the old value. When a mobile node discovers a partition, it updates the dv_{new} value to all the static nodes in the partition via the *gateway node*. While traversing the sensor network, mobile nodes come in contact with some of the static nodes in the partition. Such static nodes are referred to as *gateway node*. The mobile node collects information about the partition through gateway. Propagation of dv_{new} is done to avoid repeating the same sequence of steps for other static nodes that come in contact with the mobile node in future. The message exchange between the mobile node communication and the network is shown in Figure 3.22.

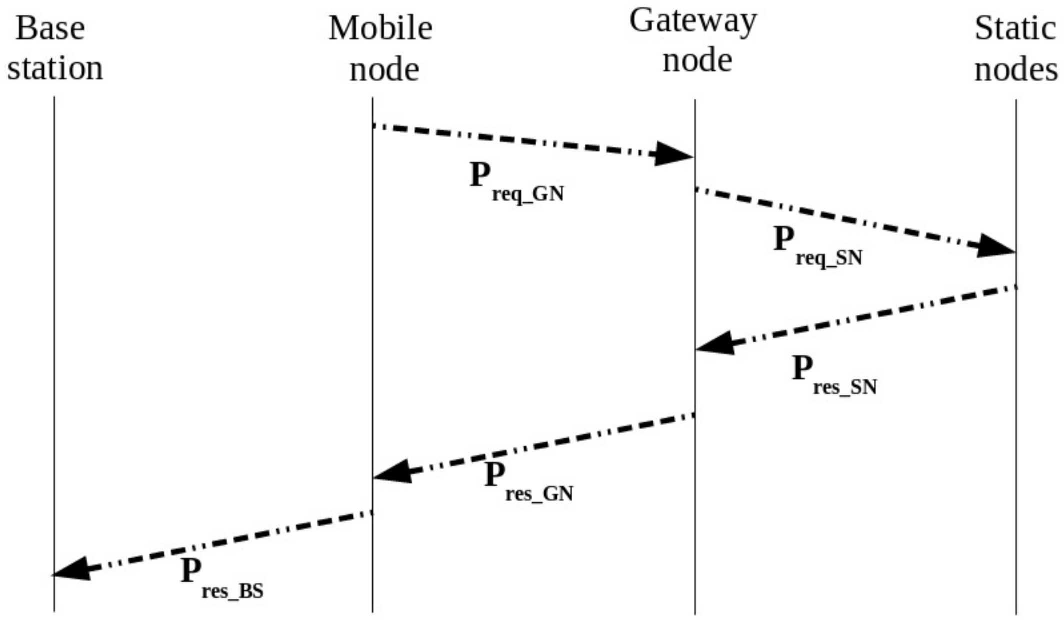

FIGURE 3.22: Message exchange between mobile node and static nodes

Mobile relay nodes use the P_{req_GN} message to request partition information from the *gateway node*. After receiving the P_{req_GN} message, the *gateway node* employs a one way broadcast P_{req_SN} message to all the nodes in the partition. Static nodes then transmit their positions, and their node ids to *gateway node* using the P_{res_SN} message. *Gateway node* in turn forwards this message to mobile relay node through P_{res_GN} message. Using the long range communication module present in mobile nodes, the information regarding the partition is transferred to the base-station via the P_{res_BS} message. The value of dv_{new} is updated in the static nodes via the P_{req_GN} and P_{req_SN} messages. Table 3.4 lists the notations used in our approach.

While exploring, if a mobile node detects a partition P_i, it collects the complete information regarding the partition via *gateway node* and updates dv_{new} value to all the static nodes in the partitioned area. The mobile node then transmits the collected information to the base-station via its long range transceiver module. The information communicated to the base-station includes starting position (X_j, Y_j), number of static

Table 3.4: Notations used: Partition Discovery

Parameter	Description
N	Number of mobile nodes.
A	Area to explore.
$M = [a_{i,j}]_{l \times b}$	Map Matrix of size $l \times b$ representing an Area A with an initial value WHITE $\forall a_{i,j}$, where $a_{i,j}$ represents a pixel in the given area. $a_{i,j}$ is set as BLACK when it is explored.
t_p	Maximum threshold value to run *random-direction* approach
P_i	i^{th} partition in the sensor field
S	Expected number of sensor nodes in the field ($\leq$ total nodes initially deployed)
S_k	Number of sensor nodes discovered by the network at any point of time
S_b	Number of sensor nodes connected to base-station. Initially $S_k = S_b$
$C(S_k)$	Area covered by S_k nodes represented using Map matrix where pixel corresponding to explored area is represented as BLACK
S_{pi}	Number of static nodes in the partition p_i
(X_{ji}^{p}, Y_{ji}^{p})	Coordinates of j^{th} node in the partition p_i
C_i	Centroid of a partition p_i
CV	Coverage value, ratio of S_k by S

nodes in partition P_i (S_{pi}) and the location of each node in the partition. The base-station uses this information to update its map matrix M and also it updates the variable $S_K \leftarrow S_K + S_{pi}$ (step 10). If the mobile node reaches the boundary, it communicates its position to the base-station. Each mobile node then repeats steps 5 to 17 till the value of $CV < t_p$, where $CV \leftarrow \frac{S_k}{S}$. CV is computed by the base-station when it receives information about the partition from the mobile nodes.

The algorithm maintains a *map matrix* representing the area to be covered. The *map matrix* is updated with an area reachable by the base-station and the area covered by the mobile nodes. The algorithm initially follows a random direction model till $\frac{S_k}{S}$ reaches t_p value. While using the *random-direction* model, if a mobile node detects a partition, it chooses a new random direction for movement after transmitting the partition information to the base-station. On reaching the threshold, the movement of every mobile node is switches to *frontier-based* model for movement. The process will continue till all partitions are discovered.

Algorithm 3.6 The algorithm for Partition Discovery

```
procedure PARTITION_DISCOVERY($N$, $M$, $t_p$, $S$, $S_b$)
    $S_k \leftarrow S_b$
    $CV \leftarrow 0$
    update $M$ with the area covered by $S_k$ nodes
    while $CV < t_p$ do
        $CV \leftarrow \frac{S_k}{S}$
        for all $N_i$ do
            move $N_i$ to a rand($\theta_i$)
            if $N_i$ detects a partition $p_i$ then
                $S_K \leftarrow S_K + S_{pi}$,
                update $M$ with the area covered by $p_i$
            end if
            if $N_i$ reaches the boundary of $A$ then
                update $M$ with the area covered by the $N_i$
            end if
        end for
    end while
    while $S < S_k$ do
        for all $N_i$ do move $N_i$ to best frontier w.r.t $M$
            if $N_i$ detect a partition $p_i$ then
                $S_K \leftarrow S_K + S_{pi}$,
                update $M$ with the area covered by $p_i$
            end if
            if $N_i$ reaches its target frontier then
                update $M$ with the area covered by the $N_i$
            end if
        end for
    end while
end procedure
```

Algorithm proposed in this Section uses a hybrid of *random-direction* and *frontier-based* methods for discovering the partitions and is given as Algorithm 3.6. Algorithm 3.6 is a modification of Hybrid area exploration approach explained in Section 3.2.2. Algorithm 3.4 is modified to explore the area with partitions. In Algorithm 3.6, the relay nodes follow a *random-direction* model, there by maximizing the probability of discovering the partitions in a short period of time. During this period we calculate the *coverage value* (CV), which is the ratio of number of nodes in the detected partition to the total number of nodes in the area. When CV exceeds the chosen threshold value, the relay nodes then follow a frontier based exploration strategy.

Restoration of connectivity in the network

After the discovery phase has been completed, the base-station has all the required information about static nodes and their positions. The next phase does the placement of minimum number of mobile nodes in order to restore connection between the partitions in the network. This phase optimizes the number of mobile nodes required. From the convex hull obtained, the Steiner point is calculated from the centroids of the partitions. We have used the steiner tree algorithm presented in Section 3.1.2.1 for connectivity restoration.

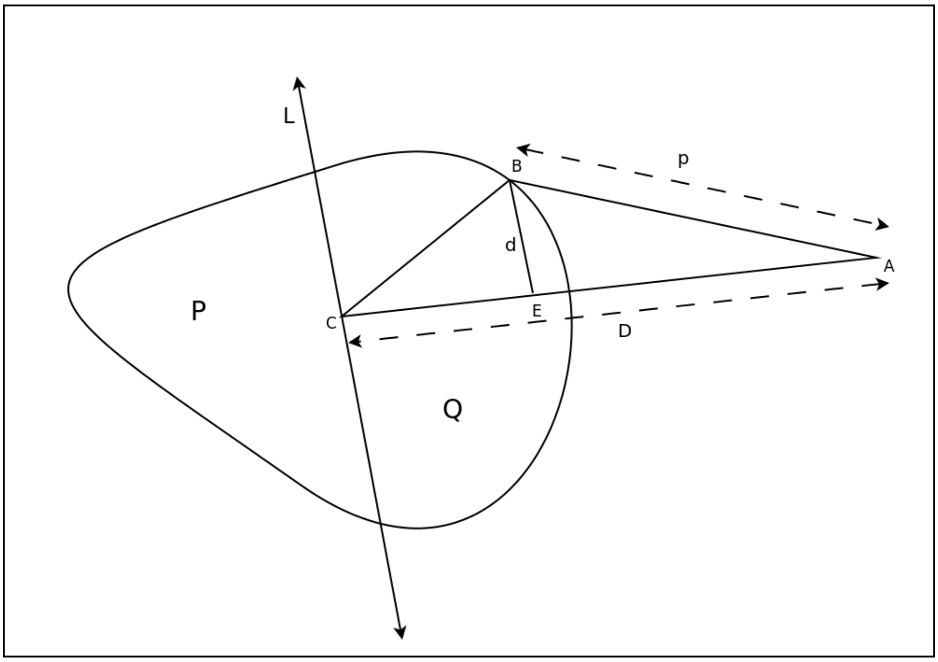

FIGURE 3.23: Initial deployment of sensor motes and AGVs on the field

After finding the Steiner points with respect to the centroid of each partition, it is essential to choose a node on the periphery of each partition to connect to the Steiner Point. This node, termed as the *Partition Interface* (*PI*) node should be selected such that it has an optimal distance from the Steiner Point. This minimizes the number of relay nodes required to be placed in order to restore connectivity. Consider the convex hull C_i from the list of discovered partitions as shown in the Figure 3.23. Steps involved in choosing the *PI* node is given below.

Steps for choosing a *PI* node

Input:

$(X_{j_i}^p, Y_{j_i}^p)$, C, A: Steiner point calculated out side the partition.

Step 1: Draw a line L through the centroid C, perpendicular to line AC.

Step 2: Divide the nodes in partition p into two parts P and Q based on their location with respect to the line L (refer Figure 3.23)

Step 3: Draw perpendicular line between every node in the coordinate $(X_{j_i}^p, Y_{j_i}^p)$ in the region Q to AC.

Step 4: Select a node Q_j such that the function if,$f(j)$=$D - d_j + p_j$ where D, d and p are the distance calculated with respect to a given node in the Q region as shown in the Figure 3.23.

then $f(j)$=$min\ f(j) \forall\ j \epsilon\ Q$

Once the *PI* nodes are determined, the mobile node positions are calculated by dividing the line connecting the partitions (Steiner point). Assuming that the above steps calculate *PI* nodes of two partitions which needs to be connected. These nodes are in the location PI_1 (X_1,Y_1) and PI_2 (X_2,Y_2). The rest of the section elaborates on the bisection algorithm used to divide the line $\overline{PI_1PI_2}$.

Let us assume that the distance between PI_1 and PI_2 is D. The slope of line $\overline{PI_1PI_2}$ is given by:

$$k = \frac{Y_2 - Y_1}{X_2 - X_1} \tag{3.12}$$

Assuming that the transmission range r of the mobile node follows a spherical pattern, we need to calculate the position to place the mobile node in the line $\overline{PI_1PI_2}$. The placement of mobile nodes is such that the successor mobile nodes are within the range of their predecessor. The equations given below express this relation, where r is the range of the mobile node and points (p, q) are the co-ordinates of the final mobile node position. Equation 3.13 calculates the distance w.r.t to PI_1 and (p, q), where (p, q) is at a distance of r

$$(X_1 - p)^2 + (Y_1 - q)^2 = r^2 \tag{3.13}$$

$$\frac{Y_1 - q}{X_1 - p} = k \tag{3.14}$$

On solving equations 3.13,3.14 :

$$p = \frac{\pm r}{\sqrt{k^2 + 1}} + X_1 \tag{3.15}$$

$$q = \frac{\pm kr}{\sqrt{k^2 + 1}} + Y_1 \tag{3.16}$$

If the x co-ordinates of points PI_1 and PI_2 are equal, the slope of line $\overline{PI_1PI_2}$ is undefined. In this case the x co-ordinate of the final mobile node position is equal to that of PI_1 and PI_2, and y co-ordinate can be obtained by adding range r directly to the y co-ordinate of PI_1.

Algorithm 3.7 Bisection Algorithm

```
$dist = distance(PI_1, PI_2)$
$ratio = dist/r$
$n = \lceil ratio \rceil - 1$
for all $i = 0 \rightarrow n$ do
    $range = r * (i + 1)$
    if $PI_1.x \neq PI_2.x$ then
        $find\ p_1, p_2$ using equation 3.15
        $find\ q_1, q_2$ using equation 3.16
    else
        if $PI_1.y < PI_2.y$ then
            $q_1 = PI_1.y + range$
            $q_2 = PI_1.y + range$
        else
            $q_1 = PI_1.y - range$
            $q_2 = PI_1.y - range$
        end if
    end if
    $calculate\ d11 = distance\ of\ (p_1, q_1)\ from\ PI_2$
    $calculate\ d12 = distance\ of\ (p_2, q_2)\ from\ PI_2$
    $choose\ the\ lesser\ one\ as\ final\ position$
end for
```

Algorithm 3.7 calculates the co-ordinates of the mobile nodes on a line $\overline{PI_1PI_2}$ by calculating the slop. Using the communication range of the mobile nodes, it computes the number of mobile nodes required to connect $\overline{PI_1PI_2}$.

Equations 3.15 and 3.16 give two sets of co-ordinates (p_1, q_1) and (p_2, q_2) and one of these is chosen as on the basis of distance. In order to get the next mobile node position, we iteratively increase *range* in multiples of r.

3.3.2 Implementation

Proof of concept of the proposed approach was done using 25 micaZ and 5 mobile relay nodes as shown in Figure 3.24. For implementation, we have used a custom designed B-bot as the mobile relay node. B-bot is built using Raspberrypi, a single-board computer with two wireless communication modules *i) zigbee: for communication with the static sensor node ii) IEEE 802.11n : to communicate with the base-station.* The details of B-bot design are given in Appendix A.

FIGURE 3.24: Testbed with B-bot for reconnecting partition.

Experiments were conducted in an area of $20m \times 20m$. The base-station was placed at the extreme end of the testbed which also has a Zigbee and Wifi interface. The base-station was programmed to communicate directly with the B-bot using *IEEE 802.11n* interface. Static nodes were to support a transmission range of $\leq 2m$. This was done by setting the power-level of transceiver. All static nodes were pre-programmed with their location in the target area. Partitions were deliberately introduced. Once the base-station gets the location of the partition, *Phase II* of the algorithm is executed, and the mobile nodes are directed towards the calculated coordinates.

The accuracy of mobile node movement depends on the localization algorithm used. Since experiments were conducted in an indoor location, using GPS on all mobile nodes was neither feasible nor economical. We have instead used static beacons to localize the mobile relays. A set of 16 micaZ nodes were deployed as beacon nodes for localization. Nodes were deployed in a grid of size 4×4 to cover an area $20m \times 20m$. Details of RSSI based localization method used is presented in Appendix B.

3.4 Summary

In this chapter, for event monitoring applications, we have proposed to use a modification of the Hungarian algorithm to minimize the maximum time taken for the mobile node to rearrange, in-order to connect the event and the base-station. We also extended our approach to monitor multi-events with minimum number of mobile nodes. For multi-event monitoring, we proposed a modification of iterative-1 Steiner approximation algorithm, to calculate the Steiner points with less number of iterations. Simulation results indicate that the modified iterative-1 gives near optimal result with less number of iterations. Combination of random-direction mobility model and frontier based area exploration method was used for area monitoring in Hybrid algorithm. Simulation results indicate that the proposed method performs better compared to frontier based method. Max-Gain algorithm moves mobile nodes to a distant unexplored position rather than directing them to a near frontier. Simulation results shows that Max-Gain approach explore the area faster than the frontier method with less communication overhead. The algorithms proposed for area monitoring and multi-event connectivity was used to discover partitions and to restore network connectivity. Proof of concept of the proposed method was verified using Berkeley motes and *B-bot*.

Chapter 4

Improving the Performance of WSNs using Mobile Nodes

The term QoS has been defined in literature from various perspectives depending on the application context [94]. Generally, it is used to represent guarantees for the service requirements of an application. Depending on the application, the QoS in WSN can be characterized based on reliability, timeliness, robustness and availability. In most wireless networks; parameters such as throughput, packet delivery ratio, delay, jitter etc., are used to measure the QoS. According to Dazhi and Pramod, the QoS perspective can be represented using a simple model, [95] shown in Figure 4.1. In this model, the users are concerned about the services that the underlying protocol layers can offer. From the perspective of the sensor network, the aim is to provide the QoS services that the user/application needs while optimizing the resource usage.

The characteristics of WSN is different when compared to regular data network. This is mainly due to various constraints such as energy, bandwidth, computing and dynamic topology. Research in WSN stack design is focused on reducing energy consumed in the network. Usually, the reduction in energy consumption is at the expense of network performance as QoS. This chapter proposes algorithms to improve the performance of existing terrestrial WSN with the help of mobile WSN. Compared to the approach proposed in the previous chapter (Section 3.1), here we use mobile node as a tool to sense

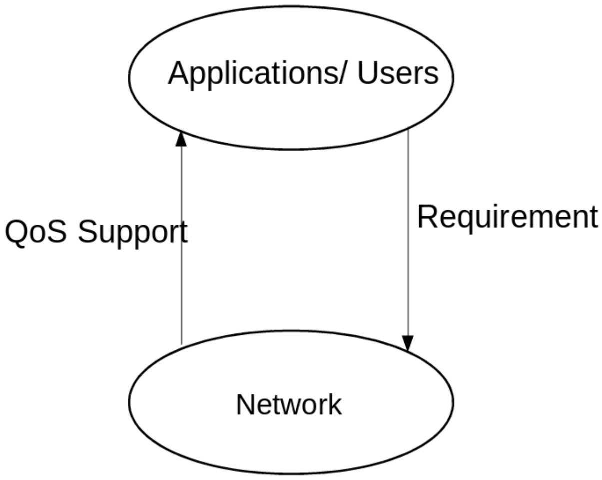

FIGURE 4.1: Simple QoS Model

the data from the sensor network and route the data via existing terrestrial WSN (routing data is done using Zigbee protocol). The proposed method uses multiple disjoint paths for streaming high bandwidth data on multiple frequency channels. The proposed method can be used in WSNs where the bandwidth requirement is high. An example of this is multimedia based routing applications.

4.1 Background

WSN sensor nodes use Zigbee (IEEE 802.15.4) for communication. TelosB and MicaZ modules from Crossbow use chipcon cc2420 radio(IEEE 802.15.4), which supports 16 orthogonal channels in an unlicensed band, each with a capacity of 250Kbps [96]. The practical data rate depends on the number of contending nodes, radio transceiver range and interference.

Results from WSN testbed show that the data rate achieved in a multi-hop communication model is much lesser than the theoretical limit due to multiple factors such as processing power and the bus architecture of the WSN node [97]. Other reasons for bad performance are intra channel interference, node orientation and environmental conditions [98].

Due to the broadcast nature of the wireless media, interference from neighbors is usually very high. One possible solution is to use multichannel MAC. Channel assignment in multichannel MAC can be done statically or dynamically. In multichannel assignment, each node is assigned a separate RF channel. Nodes need to switch the channel in order to communicate with the neighboring node. Channel allocation has to be done such that distance between motes using the same channel are separated by a distance d, where d is the *safe communication distance*, at which 99.9% of the packets are delivered and I is the interference range [72]. The advantages of using multichannel communication are increased data delivery ratio, possibility of transmitting data simultaneously from multiple sources, reduction in propagation delay and enhanced robustness [99]. For a structured deployment of sensor nodes, multichannel allocation can be done easily [72], as compared to an unstructured deployment.

Transferring high bandwidth data via terrestrial WSN nodes require a higher data rate and high packet delivery ratio. One way to achieve this is by establishing parallel multiple routes from the source to the destination. Use of multi-path routing to deliver data has advantages like increased throughput, improved security, reduction in the effective data rate in each path and hence improved energy efficiency. In this chapter, we propose the use of mobile nodes equipped multiple FDM modules, that can communicate with the base-station via multiple paths. Use of multiple FDM based sensor nodes are proposed in the literature [100] [101]. We also propose the use of multichannel MAC protocol to improve network performance.

Problem Statement and Assumptions: For a densely deployed sensor network consisting of N static sensor nodes, the objective is to sense and transfer high bandwidth data, using mobile nodes that have multiple radio modules, towards the base-station, with minimum delay without compromising the transfer rate and packet delivery ratio. The algorithm has been proposed for structured (grid) as well as un-structured (random) deployment of sensor nodes. We assume that the mobile nodes are capable of performing complex functions such as compression and coding of multimedia data. We also assume that the base-station can communicate at multiple frequencies and have multiple radio modules that can accept data arriving via multiple disjoint paths simultaneously.

4.2 Proposed Algorithm

This section proposes the use of mobile wireless sensor node for streaming high bandwidth data from the sensor field to the base-station with the help of static nodes deployed. For maximizing the capacity of multi-hop communication using zigbee, we have proposed to use node-disjoint, multi-path communication that guarantees the shortest path and energy efficiency with minimum interference. Mobile node is used to establish multiple parallel paths from sensor network to the base-station. In this section we have also empirically determined the maximum capacity zigbee can offer. We have proposed the multichannel MAC protocol for structured deployment of sensor node, while for un-structured deployment, a frequency hopping based multi-channel MAC protocol has been proposed.

The proposed algorithm has three phases that reflect the sequence of operations, that takes places in the sensor field from the initial deployment till the time at which data reaches the base-station. *Network initialization* phase (first phase) starts immediately after the node deployment. In the first phase, each static node determines its neighbor via whom it can route its data to the base-station. In *data sensing* phase (second phase), a static node senses any event in the sensor field. In the case of the event detection, the static node informs the base-station via a path computed in the previous phase. Once an event is detected, the base-station instructs a mobile node (geographically close to it) to relocate to the location of the event. The third phase of the proposed approach is the *data transfer* phase. This is done using multiple radio modules on the mobile nodes. This phase also includes positioning of the mobile node to capture an event and establish multiple paths to the base-station. Static nodes uses multichannel MAC with frequency hopping.

4.2.1 Network Initialization

Network initialization phase begins with route discovery, where static nodes need to find a route to the base-station. A route needs to be set-up between each sensor node and the base-station. We followed the approach in [102], to establish a route from every static node to the base-station. The sensor node uses this path to alert the base-station of any event. Mobile nodes connect to the nearest static node in order to communicate with the base-station. The base-station employs *reverse source routing* proposed by Johnson and Maltz [103] to communicate back to individual senor node. The *source path* field in the message received from the mobile node carries the address of the intermediate nodes along the route. The base-station uses this field to transfer the data to the mobile node.

Channel assignment: Channel assignment is done in advance for structured deployment. Here nodes are deployed in a pre-determined structure. The nodes in a grid are separated by distance d, where d is the safe communication distance. In wireless communication, the main reason for the reduction in traffic capacity is due to interfering nodes. Nodes which are communicating in the same channel create interference. One possible way to avoid interference is to use orthogonal channels for communication. The main challenge in multichannel MAC protocol is the channel assignment, where same channels should not be assigned to nodes within the Euclidean distance $I + d$, where the interference range I is always greater than the safe communication distance d [72]. In this work, we have assumed that the interference range is $2d$, so that the nodes using the same channel should be separated from each other, at a Euclidean distance of $3d$. Since our method follows a centrally coordinated system [104], we adopt the method by Guillaume et.al., [105] to determine the channel for each static node. Phase I and Phase II communications are done on common wireless channel. For an unstructured, dense deployment; it will be difficult to do channel assignment initially. The channel assignment is done after multiple node-disjoint routing paths between source and destination have been set-up.

4.2.2 Event Sensing

The static nodes sense parameters in the environment and transmit to the base-station via the routes established in the Phase I. If the base-station receives information about an event, it selects a mobile node in the proximity. To do this, Euclidean distance between the event location and the mobile node is computed. The base-station then selects a mobile node, whose Euclidean distance is minimum and sends a unicast message (M_i) to the mobile node to relocate to the new position. The message is routed to the chosen mobile node using the source route. After receiving the message M_i, the mobile node relocates to the new co-ordinates. The mobile node uses its localization unit and orientation sensors to navigate towards the assigned destination.

4.2.3 Data Transfer

A mobile node on reaching the assigned location activates its sensors. The data captured by the multimedia sensors is compressed and encoded before it is transmitted. The processed data is split into n streams, where n indicates the number of dis-joint paths calculated. The parameters used to measure the QoS of streaming media data are data-rate, reliability and propagation delay. Implementing a transport layer service for reliable data streaming requires complex algorithms and resources for handling congestion control. Delay and data transfer is proportional to the number of hops between source and destination as well as the queuing mechanisms implemented.

The data rate is determined by the number of hops between the source and destination [72], and also by the queuing mechanisms implemented. Satyajayant et.al., proposes implementing a priority based queuing model in relay nodes for streaming multimedia data [8]. Implementing such complex algorithms in a low computing capability mote such as TelosB or MicaZ may not be feasible. In this work, we also propose a node-disjoint multi-path routing protocol to establish multiple routes and hence increase the overall data rate (Section 4.2.3.1). The data packets will be delivered out-of order. We also focus on MAC layer protocol design so as to reduce node to node interference and hence by increasing the data rate.

4.2.3.1 Multi-path Routing

The number of paths that can be established between the mobile node and the base-station, depends on the total number of FDM modules available in the mobile node and the number of disjoint paths possible between the mobile node and the base-station which is represented as $n = \min(a, p)$. The value of a corresponds to the number of radio modules available in the mobile node, while p corresponds to the maximum number of disjoint paths possible between mobile node and the base-station. Based on the value of n, multiple connections are established between the mobile and static nodes. For a dense network, the value of p can be represented as $p = \min(M_n, D_n)$, where M_n is the total number of neighbors to the source(mobile) node and D_n is the total number of neighbors to destination. From the total number of paths calculated, the number of paths needed at a time is proportional to the sender data rate S_i. The total number of paths required is determined using the equation

$$k = \left\lceil \frac{S_i}{D_i} \right\rceil,$$

where D_i is the receiver rate for a single path. For example, if the sender is generating data at a rate of 500kbps and for a single path and throughput in a single path is 150kbps, then the number of paths needed is 4.

Algorithm 4.1 Algorithm to find the total number of paths

```
1: procedure TOTAL_PATH($a$, $S_i$, $D_i$, $M_m$, $D_m$)
2:     $p \leftarrow \min(M_m, D_m)$
3:     $n \leftarrow \min(a, p)$
4:     For a single path let $D_i$ be the receiver data rate
5:     No. of path needed are $k = \left\lceil \frac{S_i}{D_i} \right\rceil$.
6: end procedure
```

Algorithm 4.1 calculate the total number of paths possible. Line 2 of the algorithm calculate the disjoint paths possible by finding the minimum of M_m and D_m. Line 3 calculate the total paths possible by considering the number of disjoint paths and the number of radio modules. We have proposed a simple multipath routing technique that is not only suitable for grid topology but also for any dense random deployment. A fast, energy efficient and low complex protocols are needed for low profile sensor nodes in order to handle time critical applications. Most of the multipath routing protocols for WSN available in literature are either computationally complex, or follows a centralized approach (should know the topology in advance). Most of the protocols do not scale well with the increase in traffic and also less energy efficient. The proposed approach follows a decentralized approach and scale well with the increase in traffic and can be used in both structured and un-structured deployment. The scenario of the sensor field is as shown in Figure 4.2. The shortest path between the source and destination is taken as the *primary path*.

Then additional disjoint paths between source and destination are constructed with respect to *primary path*. This is done by k hop broadcast of control messages by the nodes in the *primary path*, where k depends on the number of alternate paths needed. A Hop_{count} field is used in the message to determine the nodes that are 1 hop, 2 hops ...k hops away from the *primary path*. Figure 4.2 shows the nodes in different colors based on the number of hop count value w.r.t *primary path*. Line $\overline{AB}$ which connects source S and destination D divides the area into two regions X and Y. S denotes the source and D denote the destination. The nodes S and D are on the line $\overline{AB}$. S' is the set of neighbors to S and D' is the set of nodes that are neigbour to D. The nodes that is green in color belong to the set of nodes in the *primary path*. Nodes around the *primary path* are grouped into set $S_1, S_2, S_3...S_k$ where k denotes the hop count. For the first alternate path, source S selects a node S'_1 from set S_1 such that the angle made between line $\overline{AB}$ joining line $\overline{SS_1'}$ is minimum, where S_1' is the intersection of set S' and S_1.

$$S'_1 = S' \cap S_1$$

The routing path is then formed via the nodes in the set $S1$, from the region X to connect to destination D. Similarly, a node from D' is then chosen based on the angle between $\overline{AB}$ and line joining $\overline{DD_1'}$, where

$$D'_1 = D' \cap D_1.$$

For the second alternate path, the same process will be repeated for the region Y. For calculating one more path from the region X, the source node S selects one of the nodes from the set S'_1 which in turn chooses a node from the set S_2 that is at minimum angle to line joining $\overline{AB}$ and $\overline{S_1{}'S'_2}$, where S'_2 is the intersection of nodes in $S_1{}'$ and S_2. If the path is to be established via set S_1, then node D needs to find a neighbor to connect to the route through nodes in set S_1. If the path is to be established to set S_2, then node D needs to finds a neighbor node from set S_1 which in turn find a neighbor from set S_2 and then connects to the route formed from the source and the process repeats.

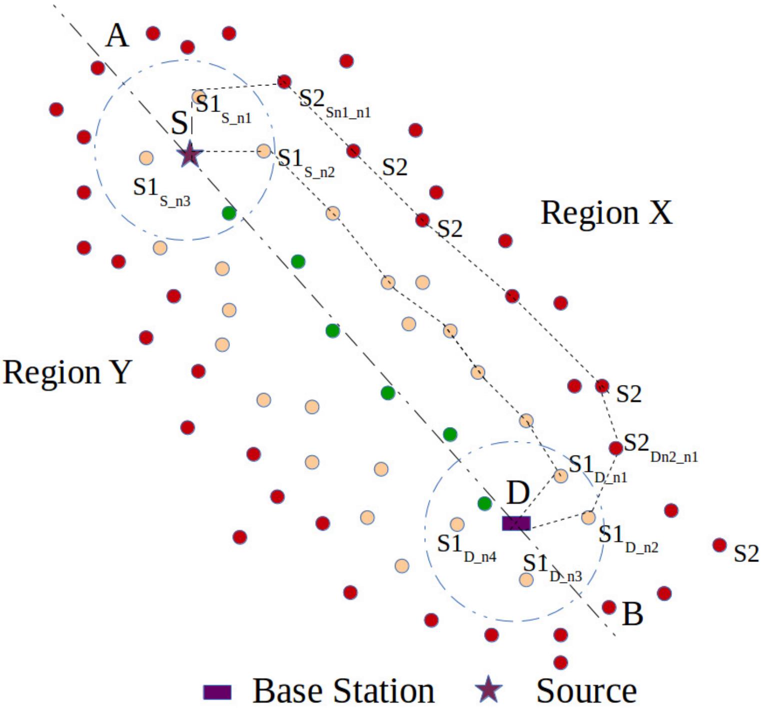

FIGURE 4.2: Multi path calculation:For Random deployment

4.2.3.2 MAC Protocol

For a grid deployment; since the nodes are deployed at a fixed distance from each other, a prior channel assignment can be done using available orthogonal channels. In case of random deployment, channels are not pre-assigned. Doing a channel assignment for a random deployment is not feasible for real-time applications. The algorithm proposed here uses Frequency Hopping (FH), for selecting the channel for communication. Use of FH reduces interference probability compared to random multiple channel allocation.

The protocol makes use of multiple frequencies to lower interference. The protocol uses slots to receive and transmit packets. In each slot, a node either transmits a packet to a neighbor or receives a packet from a neighbor. A random frequency is selected to enable frequency-hopping. For data transfer, the communication slots are to be used between two communicating nodes randomly and the slots can be computed locally by the two neighbors. For this purpose, a random seed is exchanged while establishing the routing path.

In FH, the communication channel is switched using a predetermined pseudo random channel sequence known to both the sender and the receiver. The transmitter sends a request using a predefined frequency channel (control channel). The sender and the transmitter use a common seed as input to a random number generation algorithm, which then determines the channel sequence, i.e. the sequence of frequencies that are to be used for communication. During the communication phase, both the transmitter and the receiver synchronously change the channels. For example, consider a scenario of a source (node A) sending data to a sink (node C) via node B. The communication can be shown as $A \to B \to C$. Each node maintain two variables ($\hat{C_S}$ and $\hat{C_N}$), corresponding to its current channel and the neighbor node channel(which is in the communication path). For example, the value generated corresponding to node B will be $\hat{C_S}^B$ and $\hat{C_N}^C$, where $\hat{C_S}^B$ indicates the channel node B listen to and $\hat{C_N}^C$ indicates the channel corresponding to neighbor node C. If node B need to transmit data to node C, it switches its channel from $\hat{C_S}^B$ to $\hat{C_S}^C$, transfers the data and then switch back from $\hat{C_S}^C$ to $\hat{C_S}^B$.

Nodes need to be time synchronized to implement FH. We have used Time synchronization Protocol for Sensor Networks (TPSN) for time synchronization. TPSN is a sender receiver synchronization protocol[106]. Time synchronization has to be done for the nodes after discovering multiple paths from source to destination.

4.3 Results and Analysis

Testbed experiments and simulations were conducted to evaluate the performance of the proposed approach. Experiments were conducted with Berkley motes to empirically determine the data rate and the reliability by comparing performance of single channel with CSMA and without CSMA. Experiments were conducted with a single sender and a receiver node. Sender is programmed to send at various rate with a packet size of 128 bytes. As shown in Figure 4.3, MAC without CSMA gives a better data rate. But when the number of users increase, the data-rate reduces because of collision.

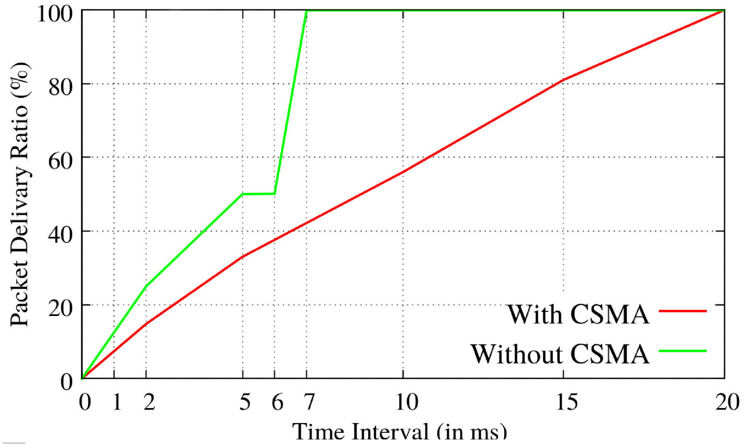

Figure 4.3: Single Channel with and without CSMA

Testbed experiments were conducted to evaluate the proposed FH based MAC protocol. The proposed method was compared with other single channel and multichannel for both structured and unstructured deployment. Since it is possible to pre-assign the channel [105], structured deployment gives a better performance. While FH based MAC protocol gives a better PDR for an unstructured deployment.

Comparison of test-bed and simulation was done to study how far the performance differs in a real-time scenario. Table 4.1 summarise the PDR difference with 5 hops for a structured deployment of sensor nodes, deployed in a grid like fashion. In the case of multichannel MAC, channels are assigned carefully in order to avoid channel interference.

It is observed that though the multichannel with FH gives less PDR compared to multichannel based MAC protocol, the proposed FH based approach will be effective in case of random deployment. This is because, assigning multichannel channel (considering the interference range) for a random deployment is difficult.

Table 4.1: PDR comparison: Test-bed Vs Simulation

Packet Delivery Ratio Comparison		
No.of hops in each path : 5	**Test-bed (%)**	**Simulation (%)**
Single Path Single Channel	68.6	87.4
Single Path Multichannel MAC	99.7	99.5
Two Path each with same Channel	67.1	87.5
Two path with Multichannel MAC	99.4	99.5
Two path with Multichannel with FH	95.4	98.5

Table 4.2: Channel Capacity

Receiving data rate (Kbps)	
No.of hops in each path : 5	**Test-bed (%)**
Single Path Multiple Channel	76.1 Kbps
Multiple Path(2 path) Each with single Channel	94.3 Kbps
Multiple path(2 path) with multichannel(Random Channel)	131.5Kbps
Multiple path(2 path) with FH	137.5Kbps

Receiver data rates for different cases are shown in Table 4.2. Multiple path with multichannel was done by allotting random channel at the time of network deployment. The pseudo random generation algorithm for FH runs periodically every 2 seconds. Test-bed results shown in Table 4.2 is obtained by multiple iterations. Test-bed result indicate that multi path with FH gives improved performance.

The proposed approach was simulated using Castalia [107] and is compared with Reliable Energy Aware Routing (REAR) [108] and Energy Efficient Collision Aware (EECA) [109] in terms of PDR. REAR consider residual energy of each sensor node in establishing routing paths. REAR supports multi-path routing protocol for reliable data transmission. REAR uses DATA-ACK oriented packet to confirm sucess of data transmission to other sensor nodes. Energy Efficient and Collision Aware (EECA) routing uses nodes position information to find two collision free routes using constrained and power adjusted flooding and then transmits the data with minimum power needed through power control component of the protocol. Simulation parameters are given in Table 4.3. Both REAR and EECA could generate maximum two paths effectively. The PDR comparison is plotted in Figure 4.4. Results indicate that the proposed approach performs better than the other approaches. This is mainly because the number of paths generated in the proposed approach is more when compared to other approaches.

Simulation parameters are given Table 4.3

Table 4.3: Simulation Parameters used for QoS Simulation

Parameters	Values
Network Size	$200 \times 200\ m^2$
Number of Nodes	75
Transmission Power	$-5dBm$
Packet size	$128byte$
Simulation Time	$500Sec$ (Multiple Iterations)

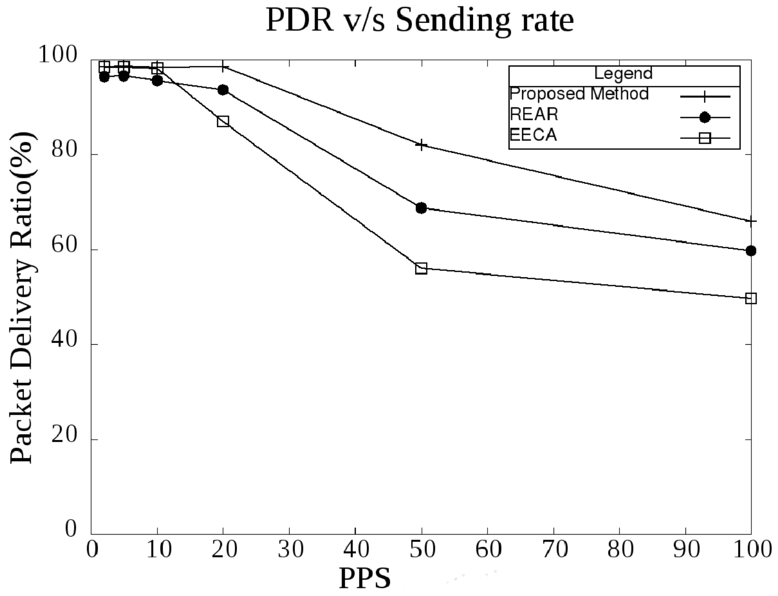

Figure 4.4: Sender data rate V/s PDR

It is also important to analyse the data rate of serial communication between the processing module and the radio at the sending side. Figure 4.5 shows the timing diagram of communication used in our approach. Serial data packets denoted as D^i_{serial} are sent from the microcontroller to the Zigbee radio. The Zigbee radio in turn forwards this message to its neighbor node (represented as $D^i_{wireless}$). Radio module after transmitting $D^i_{wireless}$ sends a serial packet back to the processor, denoted as D^i_{Ack} to acknowledge the sending of data packet. The microcontroller, after receiving the D^i_{Ack} will send the D^{i+1}_{serial} packet and the process continues. The value of t shown in the Figure 4.5 depends on the performance of microcontroller and its bus architecture.

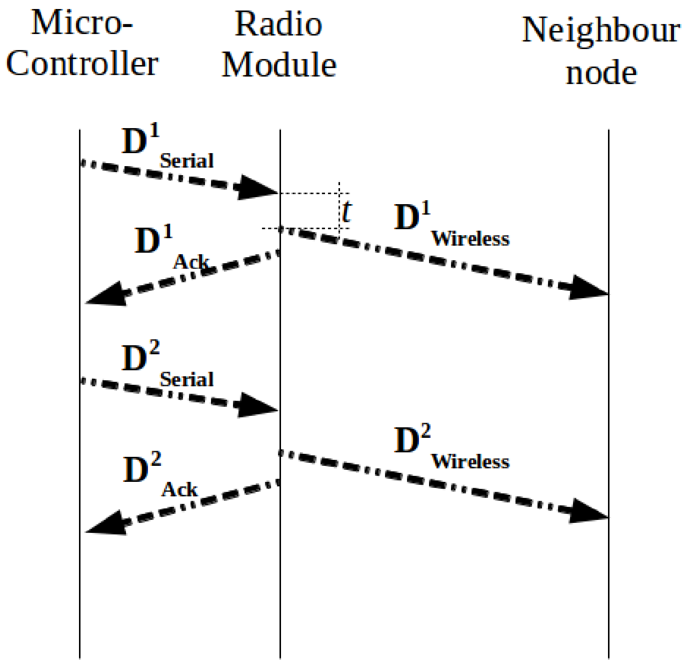

FIGURE 4.5: Serial data Timing diagram

Serial Communication: The proof of concept of the proposed model was done by connecting multiple TelosB motes into a single board computer via serial ports. Data packets were generated to send to the motes via serial ports. The motes after receiving

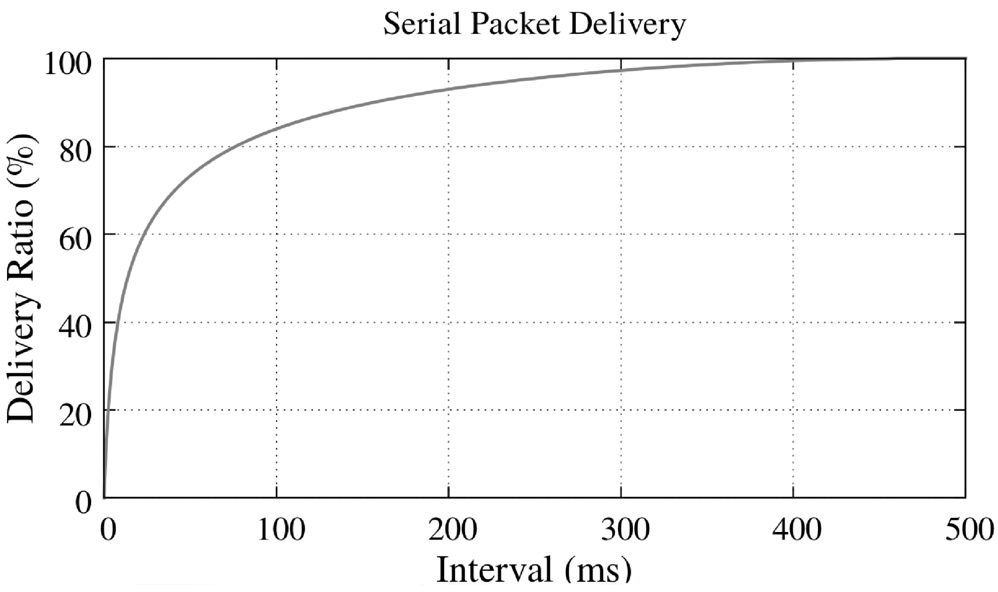

FIGURE 4.6: Serial data communication: PDR

this packets will in turn send this to the physical media. The serial communication sub-module is the TinyOS communication module which allows mote-pc communication over serial ports. In TinyOS 2.x, the serial stack structure is divided into four modules. *Dispatcher* handles the data packet bytes and delimiters. This module is responsible for reading as well as writing the data byte while receiving or sending a packet. *Protocol* unit is responsible for reading and sending all protocol control packets. *Encoder/Framer* converts the packet it received from *protocol* module into raw data byte using a serial protocol encoder/framer. The last module in the stack is *raw UART*, whose functionality is to configure the speed, stop byte, flushing the UART and sending/receiving bytes. Implementation details of the serial stack can be referred from [110]. Serial data packet includes 7 byte header. The size of the packet payload was set in $SerialPacketInfo$ interface in TinyOS. Analysis of serial data communication was done by sending a serial packet with payload size 16 bytes is shown in the Figure 4.6. Since we followed an acknowledgment based serial communication, the data rate of the sender was less. The maximum data rate achieved at the receiver was $68.8kbps$ (multichannel with 4 hop). This can be improved by building a dedicated hardware. The proof of concept of our approach was done using *B-bot*. Architecture of B-Bot is given in Appendix A.

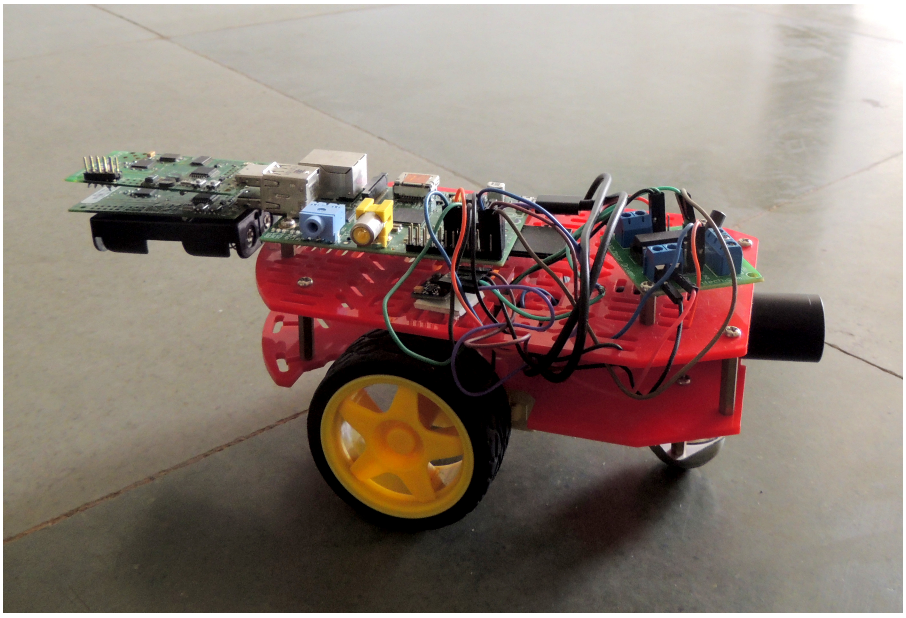

FIGURE 4.7: PoC of Mobile node

Mobile node used for the implementation is shown in the Figure 4.7.

4.4 Summary

In this chapter, we proposed to use mobile nodes to increase the performance of existing terrestrial WSN. The proposed mobile node architecture contain multiple FDM modules. The proposed method uses a multi-path node disjoint routing protocol for data communication with frequency hopping based MAC protocol. The proposed method performs better interms of packet delivery ratio compared to REAR and EECA protocol. Test-bed results indicate that multichannel based MAC protocol performs better when compared to single channel based MAC protocol.

Chapter 5

Improving Data Routing in Mobile WSN

Chapter 3 and Chapter 4 focused on improving the network performance and applications using controlled mobility feature of mobile WSN. Chapter 5 discuss on improving the routing algorithm for different mobility scenario.

In this chapter, we have considered different mobility scenario. In Section 5.1 discuss the scenario when sink is mobile. The protocol was further extended to support multiple mobile sources and sinks in Section 5.1.2. In order to maximize the contact time for data communication for a mixed deployment of mobile and static sensor nodes, a neighbor selection using mobility vector is discussed in Section 5.2. While Section 5.3 consider the data routing via mobile relays.

5.1 Data routing with a Mobile Sink

In a static WSN, nodes in the vicinity of destination are quickly drained of energy. This in turn reduces the lifetime of the entire network. One of the methods to improve the lifetime of the network is to make the sink mobile. If the sink is mobile, its neighbor changes and this ensures that the load in the network is more uniformly distributed.

Hence, the overall lifetime of the network is increased. Routing data to a mobile destination has its own challenges. Since the geographic location of the sink changes, delivery latency in a network increases. Also the geographical location of the sink must be constantly updated. Existing routing protocol use acknowledgments packets for updating the sink's location. The process of acknowledgment further adds to the latency in the network and also to increase the control overhead.

Other methods for sink location update used are full flooding based, using rendezvous points, local flooding and forming grids [23]. Using these update mechanisms, several data dissemination protocols have been developed. The majority of these protocols use rendezvous points-based updates. In this methods, node is aware of its geographical location via GPS or other localization methods. In Two-Tier Data Dissemination protocol(TTDD) proposed by Fan et.al., [111], a grid topology is created. Source node sends data to the destination via certain relay points. The disadvantage of this protocol is the huge overhead required for grid formation.

According to Elyes and Guillaume [112], Line Based Data Dissemination (LBDD) protocol divides the network into two parts, where a rendezvous line is created for data storage and lookup. The data generated by the source is disseminated via these rendezvous lines and the sink will query an event through these lines. The point where the source and the sink query meets is used to route the packet to the destination. Calculation of rendezvous points is highly complex and the resultant energy consumption is high.

Routing protocol proposed by Hyung et.al., [113], creates a dissemination tree by selecting an access node with the help of mobile sink for sending a query to the source node. The creation of dissemination tree each time as the sink moves, increases the overhead. Quadtree-based Data Dissemination (QDD) proposed by Mir and Ko [114], is yet another type of dissemination protocol and is based on a Quadrature based partitioning approach, where it divides the physical network into successive quadrants. When a source node detects a new event, it calculates a set of rendezvous points by successively partitioning the sensor fields into four equal logical quadrants, and the data is forwarded to nodes, which are closer to the centroid of each successive partition. Here only few static nodes are selected as rendezvous points for creating a hot spots and thereby reducing the network lifetime.

As suggested by Lee et.al., in [115], the Expect Grid based Real-time Routing protocol creates virtual grids in the network. The propagation of the location information to the dynamic sink is done by constructing Expect Grids (EG) in the network. An Expect Zone (EZ) is created around the location of the mobile sink. EZ includes those EGs, one of which contains the sink node. The data from the source node is initially propagated to a closest point in the EZ, which is then multicasted to all the EGs in the EZ. The EG where the mobile sink resides, will unicast the data to the actual destination. Although this protocol saves much energy as multicasting is more energy efficient than flooding within the network, the creation of EZs and EGs regularly as the sink moves creates overhead issues.

Although many other efficient schemes have been proposed in the recent past for routing the data to a mobile sink, elastic routing proposed by Fucai et.al., has several advantages in terms of control overhead, data delivery latency and energy consumption [116].

Elastic Routing utilizes the greedy forwarding technique for routing the data packet to the mobile sink. Geographic routing utilizing greedy forwarding [117] has been considered a promising routing protocol for the dynamic topology of sensor nodes, since it exploits pure location information rather than global topology information. Although many other efficient schemes have been proposed in the recent past for routing the data to a mobile sink, elastic routing was considered to be superior in terms of control overhead, data delivery delay and energy consumption. Elastic routing also makes use of overhearing feature of wireless transmission, in which the node can overhear the transmitting packets in the neighborhood, even-though the packets are not destined to it. When the sink moves, the updated position of the sink will be overheard by the last hop forwarding node, which then updates its neighbors about the new location of the sink. As the sink moves, the path between the sink and the source gets inflected, but finally converged like a shrinkage of an elastic band. It has been analyzed in [116], that the elastic routing is efficient in terms of less overhead packets, for routing the data packet to a mobile sink from a static source.

In this thesis, we adopt the concept of elastic routing [116] to route the data between multiple sources to mobile sink. Use of the elastic routing technique for transferring the data will result in individual paths from each source to the mobile sink. This in turn increases the number of hops which increases the data delivery latency. We suggest a modification to elastic routing. We have proposed a methodology that uses Steiner tree to reduce the number of hops. The algorithm has been designed and analysed for multiple mobility scenarios.

5.1.1 Scenario 1: Multiple Sources and Mobile Sink

In this section, the proposed algorithm is used in a scenario where there are multiple mobile sources and sinks. As the nodes are mobile, it is not possible to apply the approximation algorithm for the Steiner tree directly. We propose a dynamic Steiner tree algorithm that will adapt to the changes in the topology.

Problem Statement: A network composed of N static nodes uniformly distributed in an area A, with a transmission range of radius R. Out of the N static nodes, n nodes have data to transmit to a mobile sink (N_m). The aim is to transfer the data between multiple static sources and the mobile base-station, by determining the shortest path. The protocol reduces the number of control messages for updating the position of the sink, thereby enhancing energy efficiency.

Assumptions: We assume that each node knows their geographic location in advance. We assume that the initial location of the sink is known to all the sources. We also assume that $n \ll N$.

Motivation: A network where three sources are communicating to a base-station is shown in the Figure 5.1. This is defined by set $V = \{V_1, V_2, V_3, V_4\}$, where V_1, V_2, V_3 are the three sources and V_4 is the base-station. Distance between the vertices a and hence the corresponding diagonal distance is $\sqrt{2}a$. Assuming the sink is static, the aim is to connect all the vertices in set V via the shortest path.

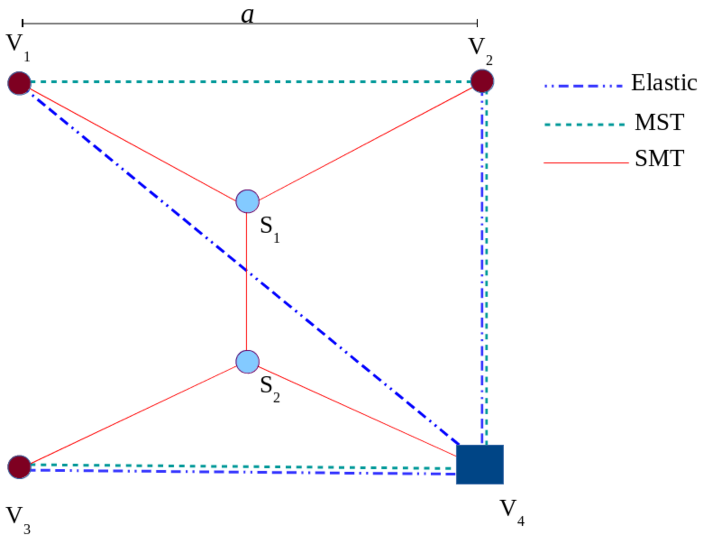

Figure 5.1: Elastic Vs MST Vs SMT

In Elastic routing, the location of the sink gets updated at each source separately using overhearing. The routing pattern is as shown in Figure 5.1. The cost (c) of the path is the sum of the individual paths for routing data to the sink, this is given as

$$c(Elastic) = \sum_{i=0}^{n} c(V_i, N_m),$$

where N_m indicates the sink. With respect to the Figure 5.1 cost c, of elastic routing is $2a + \sqrt{2}a$ ($3.41a$).

For the same topology, with the MST approach, the routing distance is further reduced. Here, the total cost of the route is $c(V_1, V_2) + c(V_2, V_4) + c(V_3, V_4)$, which is

$$c(MST) = a + a + a = 3a$$

SMT provides a better solution compared to MST, by creating extra vertices. In SMT, a Steiner tree is created, by finding extra vertices called Steiner points (S_1, S_2), from the set V shown in the Figure 5.1.

Once the Steiner points have been identified by the sink, these new set of points are propagated to each and every source nodes (V_1, V_2, V_3). The destination node also identifies the Steiner edges based on the calculated Steiner points. Here, the set of Steiner tree edges formed will be $(V_1, S_1), (V_2, S_1), (V_3, S_2), (S_1, S_2), (S_2, V_4)$ The cost corresponding to each edge in the resulting tree is

$$4a \times (\frac{\sqrt{3}}{3}) + a(1 - \frac{\sqrt{3}}{3}),$$

as given in [118]. So the overall cost of the resultant tree is given as

$$c(SMT(V)) = a(1 + \sqrt{3}) = 2.73a$$

From the above scenario, it is clear that $c(Elastic) > c(MST) > c(SMT)$, for the given topology in Figure 5.1. Since cost of SMT is less than or equal compared to other approaches, the number of hops and hence the communication overhead also reduces. Hence, we have used SMT based routing for data communication between sources and mobile sink.

Steiner tree aims to connect a given set of vertices V in a graph G using minimum length. Steiner tree is similar to Minimum Spanning Tree(MST), except that the algorithm calculates extra vertices called Steiner points. In Figure 5.1, S_1, S_2 are the steiner points for the set V. (S), whose cost (c) is given by:

$$c(SMT(V)) = c(MST(V \cup S))$$

where,

$$c(MST(V \cup S)) \leq c(MST(V))$$

In the proposed algorithm, the vertex V is the set of source nodes and mobile sink and is defined by,

$$V = (n \cup N_m)$$

To minimize the overall energy consumption, the proposed algorithm utilizes a modified version of iterative-1 Steiner approximation algorithm [70], for calculating the Steiner points. It then routes data using these Steiner points.

The objective is to construct the SMT from the given set of source and mobile nodes. For a given graph with V vertice's, a Steiner tree has at most $V - 2$ Steiner points [76]. The resultant number of Steiner points when iterative-1 approach is used may contain more than $V - 2$ points, this can be reduced later based on the degree of the node [70].

Unlike Steiner iterative-1 approach, the proposed algorithm uses the node location as the Steiner candidate set for finding the vertices. Steiner candidate set is denoted as $H(N)$ and contains $N - n$ elements. Algorithm 6.2 explains the approach we followed for Steiner node calculation. The calculation is done at the sink after it receives the location of the sources.

For any set V, the Steiner point with respect to a point $x \in H(N)$ that maximizes $\Delta MST(V, x) > 0$, where $V = \{n \cup N_m\}$. Starting with a set $S = \emptyset$ of Steiner points, the Iterated Steiner method repeatedly finds a Steiner point x for $\{N \cup S\}$ and sets $S = \{S \cup x\}$. The cost of $MST(N \cup S)$ will decrease with each added point, and the construction terminates, when there no longer exists any point x with $\Delta MST(V \cup S, \{x\}) > 0$. A sink node on receiving the source location co-ordinates, will calculate $c(MST(V))$, where V indicates the set of sources and the sink.

Algorithm 5.1 Steiner point calculation

```
procedure STEINER_POINTS(V, H(N))
    Steiner_set $S = \emptyset$
    while
cand_set=$\{x \in H(V \cup S)/\Delta MST(VUS), x) > 0\} \neq \emptyset$ do
        Find $x \in$ cand_set which Max($\Delta MST(V \cup S, x)$)
        $S = S \cup \{x\}$
    end while
    Remove points in $S$ with degree $\leq 2$ in $MST(V \cup S)$
end procedure
```

Algorithm for calculating the Steiner tree is given in Algorithm 5.1. Sink uses *STEINER_POINTS(V, H(N))* shown in Algorithm 5.1, to calculate the Steiner tree $SMT(V)$ and hence calculate $c(SMT(V))$. If $c(SMT(V)) < c(MST(V))$, then sink communicates the <*source*, *destination*> pair information corresponding to each edge of the resulting tree.

The resultant $SMT(V)$ contains the edges corresponding to a sender and a receiver, which can be a source or a Steiner point. The newly calculated Steiner tree edges (x_i, y_i) will then propagate to all source nodes to construct the Steiner tree. Steiner nodes are also updated to route the information it receives to its immediate vertex, based on the edge information available in $SMT(V)$. The edge (x_i, y_i) corresponds to a vertex $(x \in (V \cup S))$, which can be a source node or a Steiner node and y corresponds to the destination, where y can be a Steiner point or a sink node $(y \in (V \cup S))$.

The sink node uses greedy forwarding to route the information to the source corresponding to each $<source, destination>$ pair. Since the sink is mobile and the topology is dynamic, calculating $SMT(V)$ and frequently broadcasting the route increases the cost. Hence, we use a simple threshold based network update mechanism, that handles the dynamic topology of the network. Since source nodes are static, the coordinates corresponding to the sources will be fixed. Even though the sink is mobile, the sink can calculate the $c(MST(V))$ periodically, with respect to its current location and the source coordinates. The calculated cost is then compared with $c(SMT(V)')$, where $c(SMT(V)')$ indicates the last calculated Steiner tree cost. If the value of $c(MST(V)) \leq c(SMT(V)')$, then the sink will recalculate $SMT(V)$ and broadcast the new Steiner tree information to the corresponding nodes.

Once the Steiner edges have been propagated to the sources, each source then routes the data to its corresponding destination vertex. In the proposed protocol we have used the overhearing feature of elastic routing [116], to update the sink position. Once the sink moves, the vertices of $SMT(V)$ is updated via overhearing. The vertices are updated only if there is a change in Steiner node values, while calculating $SMT(V)$ periodically. Scenario of multiple mobile sources and mobile sink is a special case of scenario 2 which is explained in Section 5.1.2.

5.1.2 Scenario 2: Multiple Mobile Sources and Multiple Mobile Sink

Multiple mobile sinks are better than multiple static sinks as it increase the reachability in the network. The routing protocol designed for such a scenario should scale for a large network. In this section, a routing protocol, which is an extension of elastic routing has been proposed for a network with multiple mobile sources and multiple mobile sinks.

5.1.2.1 Routing Protocol

Problem Statement: The network is composed of N static nodes uniformly distributed in an area A, with a transmission range of radius R. Of the N sensor nodes, n mobile nodes have data to transmit. The network also has m mobile sinks which are used for data collection. The aim is to establish a route from each source to the nearest mobile sink. If the data transfer is from multiple mobile sources to a single mobile sink, then an optimal routing path needs to be calculated connecting all the nodes. The proposed routing protocol attempts to reduce the number of control messages for updating the sink's position.

The objective of the proposed protocol is to incorporate the overhearing feature utilized in the elastic routing technique, to route the data, in a scenario where there are multiple static sources and multiple mobile sinks present in the network. For finding an optimal sink as well as to keep track of the current sink position *Traffic Re-director(TR)* node is used . A static node in the centroid region is selected as a *Traffic Re-director* (TR) node.Figure 5.2 is a depiction of the network scenario. We assume that the static nodes deployed are aware of their location. We also assume that all the mobile nodes can determine their location.

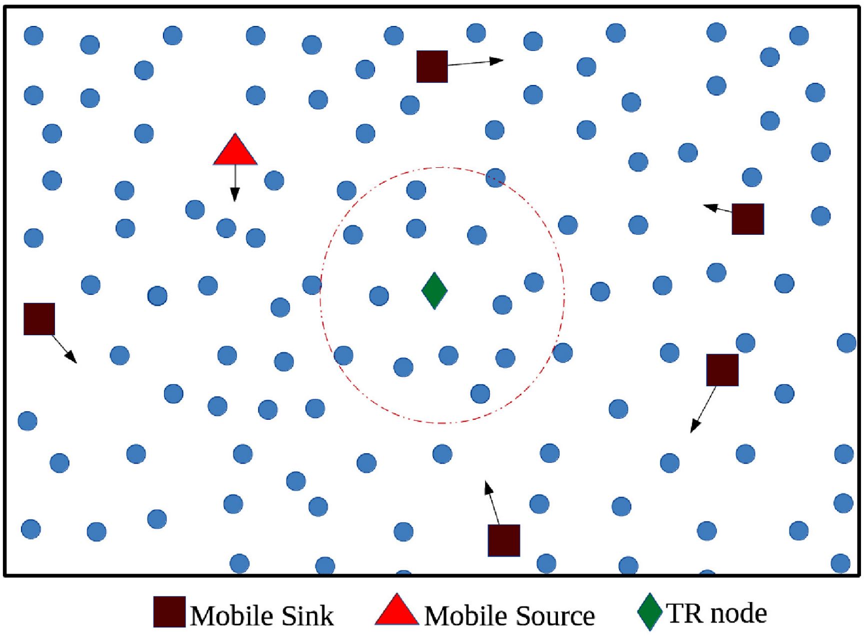

Figure 5.2: Initial scenario

5.1.2.2 Traffic Redirection with TR nodes

Static nodes collect information about their neighbor through exchange of messages. A static node broadcast Inf_{req} message requesting for information from its neighbors. When a node receives the Inf_{req} message, it responds with Inf_{rep} message that has information about its ID and location. Each static node stores neighbor information in its *neighbour record*. neighbor record is used for routing.

Routing protocol proposed in [116], assumes that the location of the mobile sink is known to the source in advance. This does not hold true in practical situations. When the position of the sink changes, the initial location information of sink contained in the source becomes obsolete. An alternative mechanism is required for updating the sink's location. We have proposed the use of a TR node. The sink periodically updates its location information to this TR node. We have also proposed an active virtual region called *centroid region CR*, formed around the centroid of the sensor network. The region is circular in shape and its radius depends on the node density. The minimum radius should be atleast r, where r is the transmission range of the static nodes. A

TR node is selected from the *centroid region*. The node selection criteria is based on residual energy and position.

As its position changes, the mobile sink updates its location information to the TR node via a position message P^i_M. The destination address in P^i_M is marked as (X_c, Y_c), where (X_c, Y_c) indicate the centroid coordinate of the area A. Routing technique used is greedy forwarding. When a P^i_M packet enters the CR region, any node within the CR region changes the destination address of P^i_M packet from (X_c, Y_c) to (X_{TR}, Y_{TR}). (X_{TR}, Y_{TR}) that represents the location of current TR node. The current TR node will keep track of the updated positions of each mobile sink in its *Sink Location* SL table.

TR Node Selection: CR region is a fixed circular area with its center at (X_c, Y_c) . A TR node is selected periodically based on the residual energy of the node. When the energy level of the current TR node drops below threshold, it broadcasts ETR_{Req} message within CR. A node on receiving ETR_{Req} send a ETR_{Rep} message with a *priority* value after a delay of Δt time. The value of Δt time depends on the *priority* value and is given as

$$\Delta t \propto \frac{1}{priority}$$

Priority depends on the available threshold value. The higher the priority value, lesser is the delay. When a node receives ETR_{Rep} message, it checks the priority value of the message. If the TR nodes priority value is less than the priority value in the received message, it rebroadcasts the ETR_{Rep} message. As the CR region is a small area, the communication overhead is lesser. The old TR node then will forward the SL table to the newly elected TR node.

5.1.2.3 Mobile Sink Routing

If a node has data to transmit, it forwards the data packet with destination addressed to (X_c, Y_c). The message also has the current position of the mobile node (X^i_s, Y^i_s). When the packet is received by any node in the CR region, it changes the destination address to (X_{TR}, Y_{TR}). TR node on receiving the packet searches for the node close to the sink. The (X_s^i, Y_s^i) from the message and the location information stored in SL table is used

An optimal sink selection is then done by using the Euclidean distance between the source and the available sink. A sink at minimal distance from the source is selected by TR node. The TR node hence forwards data greedily to the sink.

The routing is then using elastic routing protocol. The initial scenario of two sources transmitting to a mobile sink via elastic routing (Figure 5.3). Figure 5.4 shows the route between the same pair of $< source, sink >$ after some time.

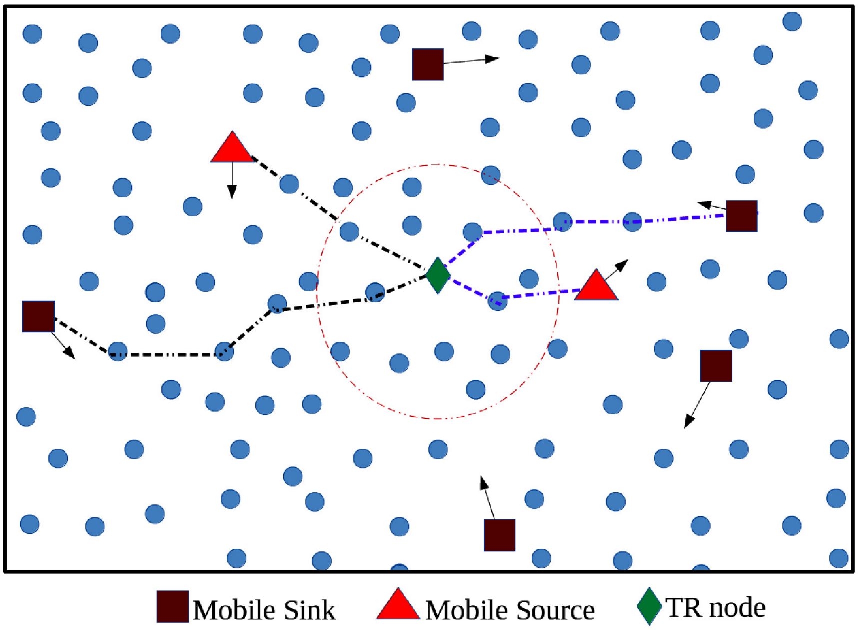

FIGURE 5.3: Source sending data to Sink via TR node

5.1.2.4 Mobile source Routing

When the source is mobile, a source selects a static node for forwarding the data to TR node. When a mobile sink moves to a new location, it selects a new neighbor. The mobile sink then updates the new neighbor information to the old neighbor. The data sent to the new neighbor will be routed to the destination via the old neighbor. Due to

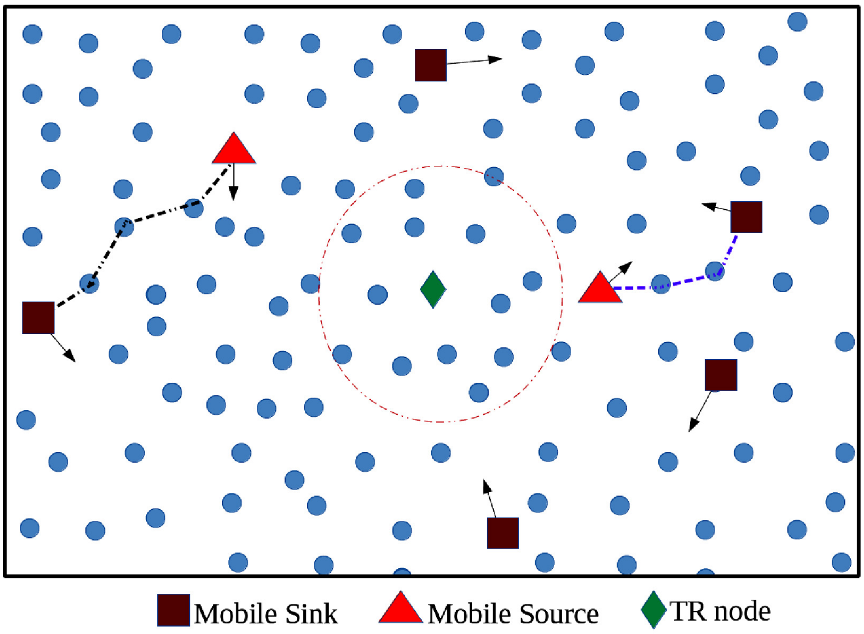

FIGURE 5.4: Source communicating directly with sink using Elastic routing

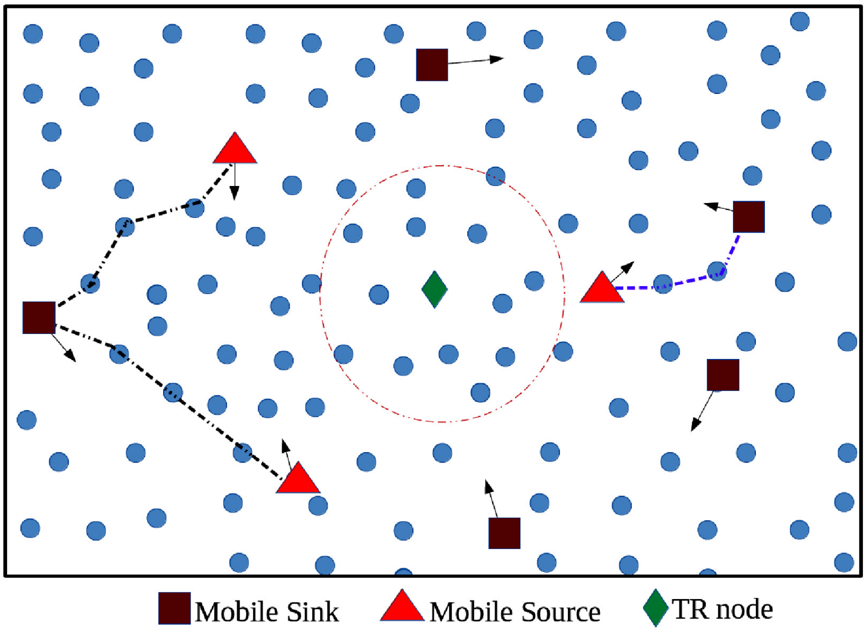

FIGURE 5.5: Scenario of multiple sources communicating with single sink node

the overhearing feature of the elastic routing protocol, the new location of the mobile sink will be known to the source within a few iterations of data transmission.

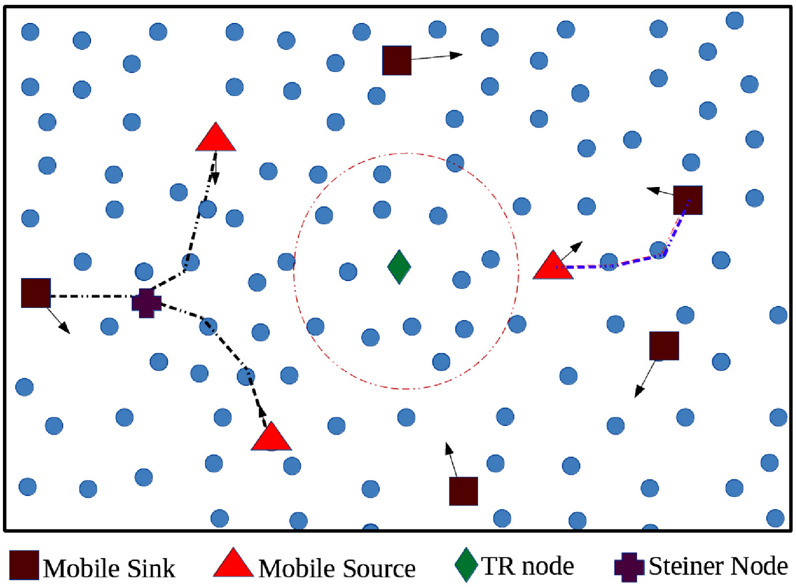

Figure 5.6: Communication between multiple source to single sink via dynamic Steiner tree

5.1.2.5 Routing between multiple mobile sources to a single mobile sink

Multiple mobile sources can route the data to a single mobile sink. The scenario where two mobile sources communicating to a mobile sink is shown in Figure 5.5. The scenario is Figure 5.5 is similar to the scenario of Section 5.1.1. The dynamic Steiner tree algorithm proposed in Section 5.1.1 can be used to establish data routing between multiple sources to single mobile sink. Figure 5.6 shows the data communication after establishing a Steiner tree. It is show in Figure 5.6 that if multiple sources communicate to a mobile sink, a steiner tree will be created to connect the sources and its corresponding sink.

5.1.3 Results

Performance evaluation was done by comparing the dynamic Steiner tree approach with MST, elastic routing and expected grid based approaches. Simulation was done using Castalia (an OMNeT++ framework), by varying the number of source nodes, data rate, and velocity of the sink. Table 5.1 illustrates the parameters used in the simulation. The mobility model used for the sink node movement was random way-point model. Evaluation of the proposed protocol was done to evaluate the energy consumption, number of hops, data rate, and packet delivery ratio. The results given below is for scenario where multiple sources are sending data to single mobile sink.

TABLE 5.1: Simulation Parameters: Routing between sources to mobile sink

Parameters	Values
Network Size	$200 \times 200\ m^2$
Number of Nodes	50 to 500
Transmission Power	$-5dBm$
Velocity	$0.5m/s$ to $20m/s$
Number of Mobile Nodes	1 (mobile Sink)
Packet size	$128byte$
Simulation Time	$500Sec$ (Multiple Iterations)
Mobility Model	Random Way-point

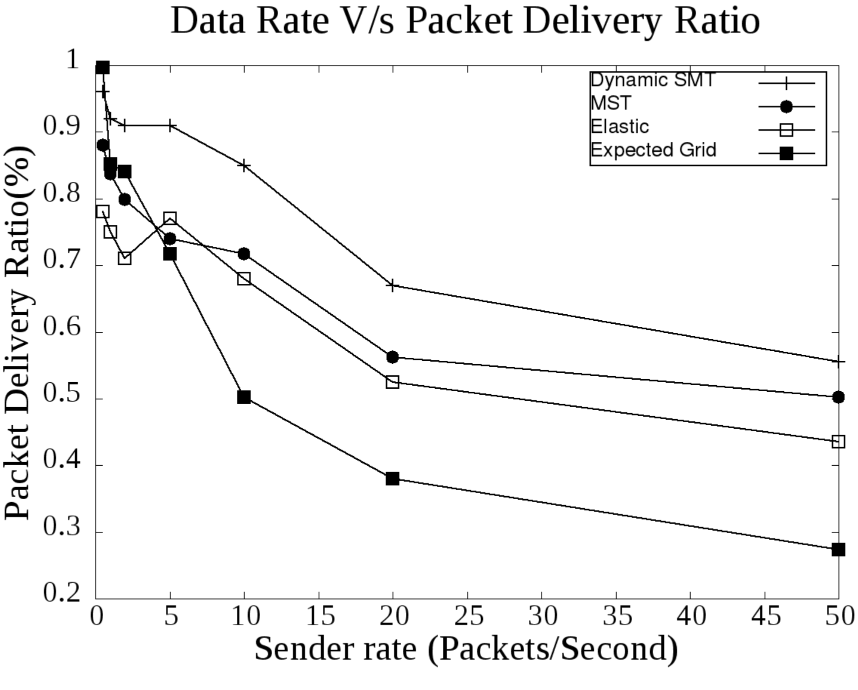

FIGURE 5.7: PDR Vs Data rate

The Packet Delivery Ratio (PDR) is the fraction of the number of packets received with respect to the number of packets sent. The PDR of the proposed approach was calculated, by varying the rate at which the packets are generated at the source nodes. As the data rate increases, the packets delivered at the sink changes. The Figure 5.7, compares the PDR of the proposed protocol with that of elastic routing, MST and EG

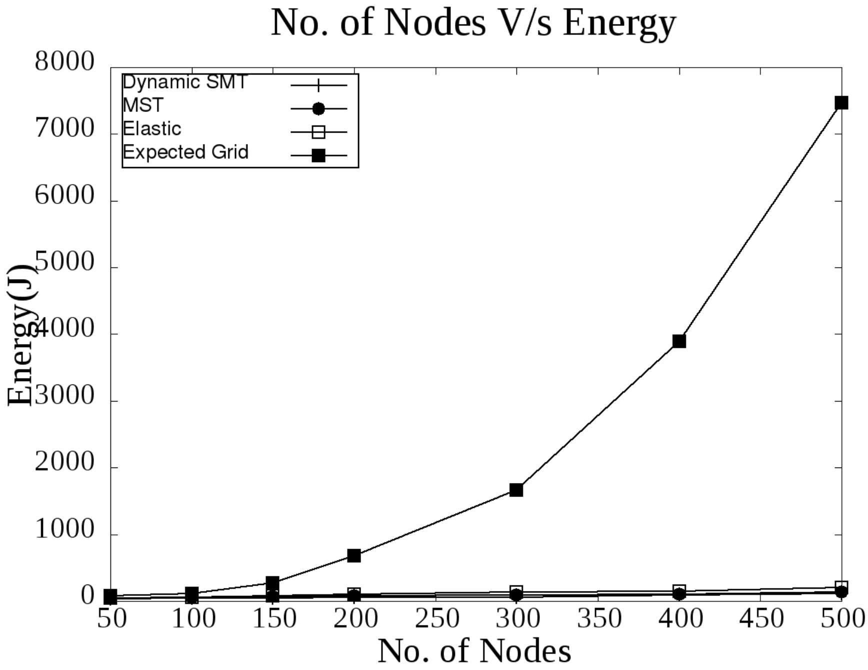

FIGURE 5.8: Energy Vs No of Nodes

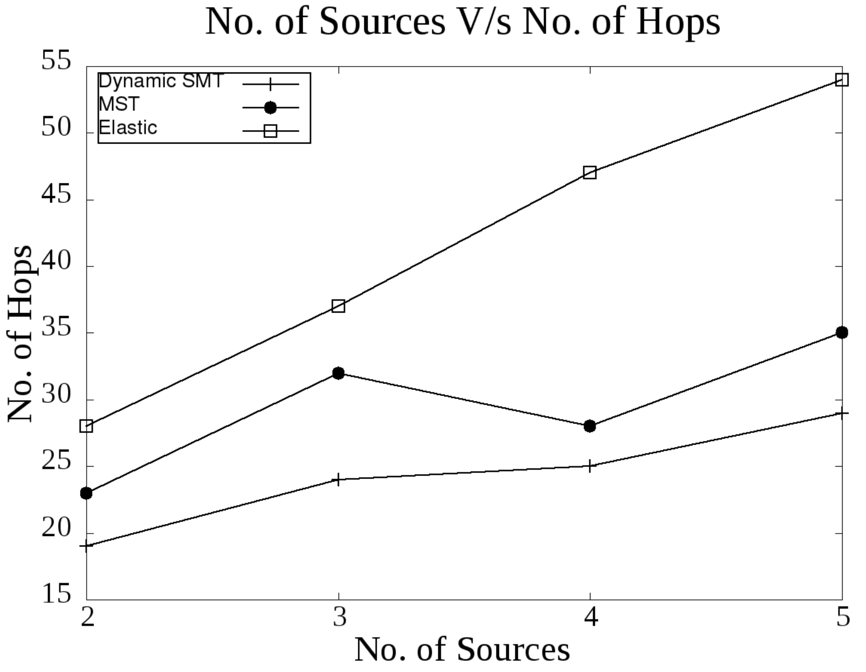

FIGURE 5.9: Hop Vs No of Sources

approaches. The result shows that the algorithm proposed performs better than the other algorithms.

The energy consumed in the network is proportional to the number of nodes participating (number of hops) for routing and it also depends on the control overhead on the

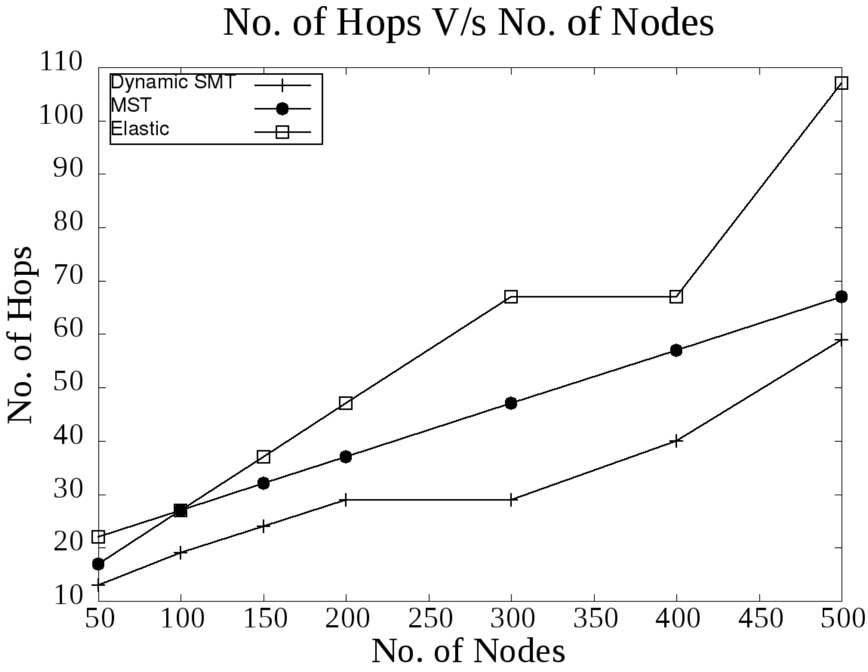

FIGURE 5.10: No of Hops Vs No of Nodes

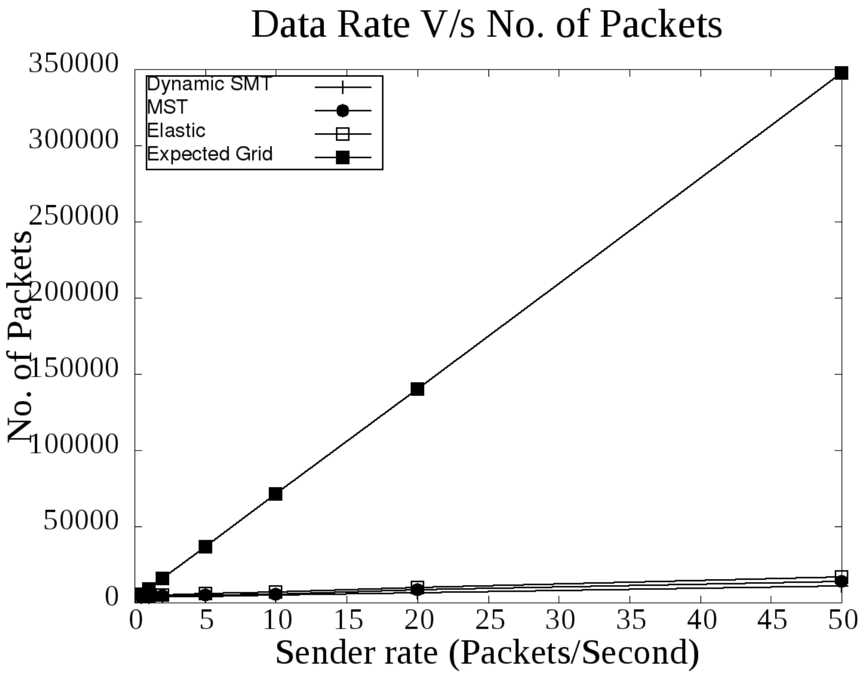

FIGURE 5.11: No of Packets Vs Data Rate

network. Energy efficiency of the proposed approach has been analysed by comparing it with other algorithms. Results are shown in Figure 5.8. The initial comparison was done with respect to number of hops. The analysis was done by varying the number of sources and also by varying the size of the deployment area.

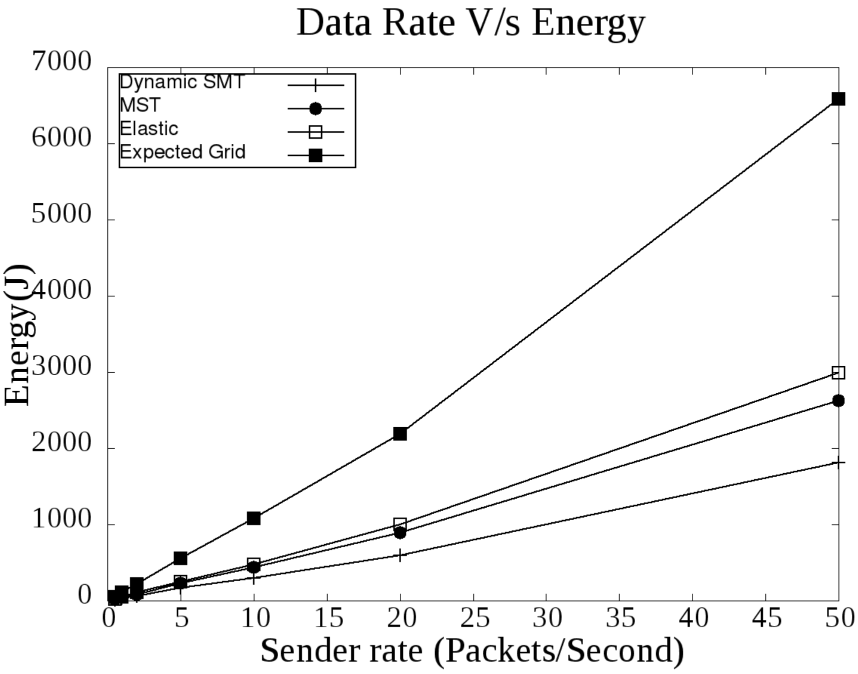

FIGURE 5.12: Data Rate Vs Energy

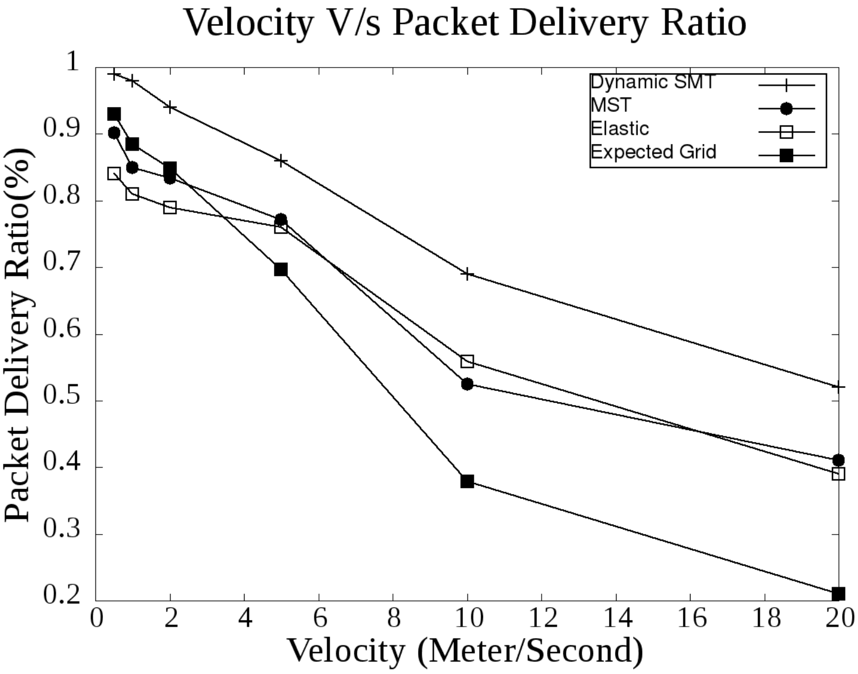

FIGURE 5.13: PDR Vs Velocity

Figure 5.9 shows the results obtained by varying the number of sources for a fixed area of $200 \times 200\ m^2$. The simulation was done by varying the number of sources (n). For i sources, the number of vertices in the graph is $i + 1$. Simulation result indicates that the number of hops needed for SMT is less compared to MST and Elastic routing.

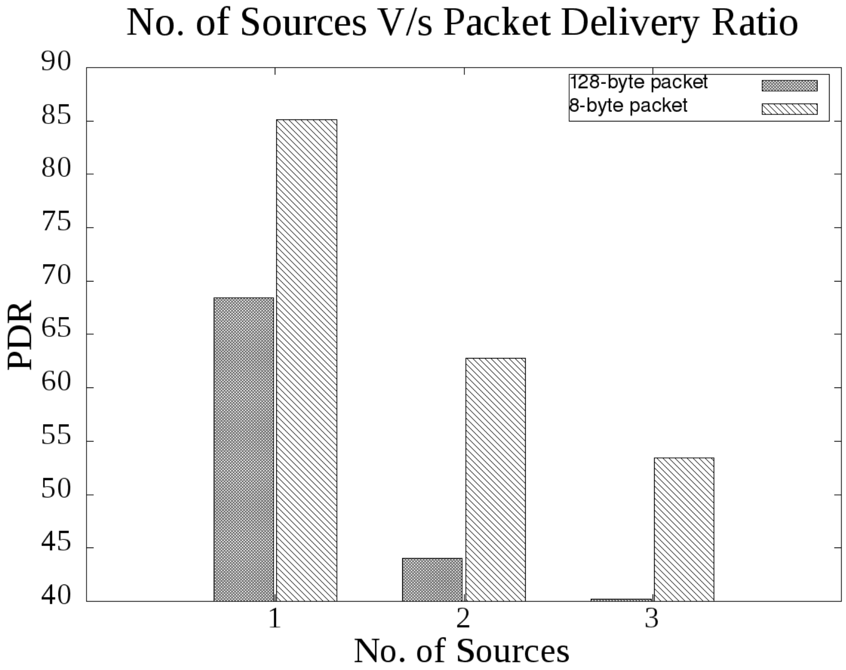

FIGURE 5.14: Test-bed:Source Vs Packets received

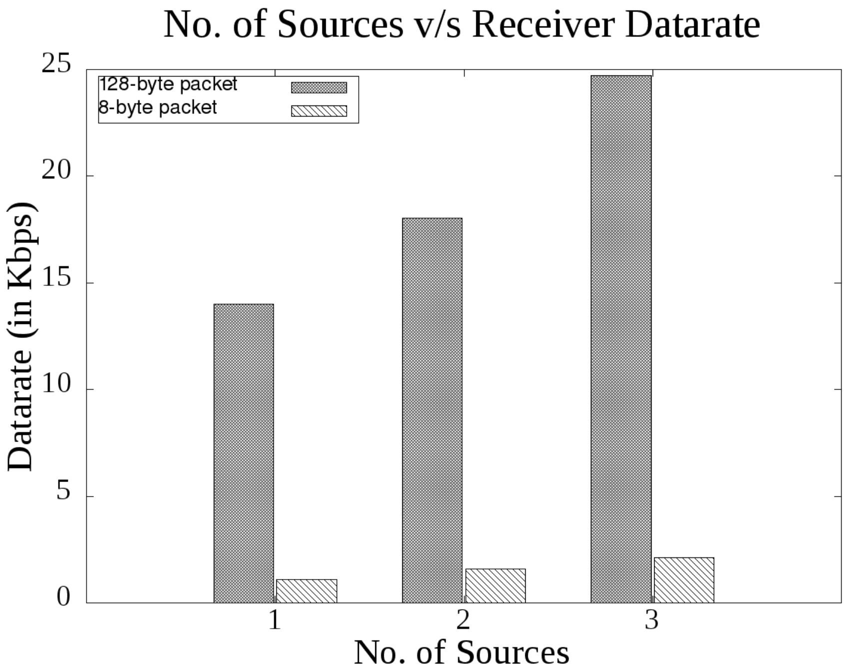

FIGURE 5.15: Test-bed:Packets Received at each vertices

Analysis was also done by varying the area of deployment. A uniform random deployment of sensor nodes was used. Sensor nodes transmission power was set to $-5dBm$, The number of nodes was plotted against the number of hops (Figure 5.10). Simulation was done with three sources and a mobile sink. The x-axis indicates the number of nodes corresponding to the deployment area and y-axis indicates the number of hops. From the Figure 5.10, it can be observed that the cost of SMT is less, hence the number of hops needed to connect from all the sources to the sink is also less. Number of packets generated in the network were plotted against the data rate, to compare the communication overhead. Figure 5.11 shows the comparison. The velocity of the mobile node was set to $2m/s$. From the figure, the control overhead is highest when using EG. Control overhead when using dynamic SMT approach is less, because of less number of hops.

Comparison of the residual energy of nodes participating in the data communication is shown in Figure 5.12. Energy was compared with varying data rates from $1pps$ to $50pps$, with a packet size of $128bytes$. The sum of the difference between the energy before and after the simulation was used for analysis. As the packet rate increases, the amount of energy consumed also increases. This is because, more packets are generated in the network due to the increased data rate. The amount of energy consumption in the dynamic SMT approach is significantly less because of less number of nodes participating in data communication. Similarly, a comparison was done by varying number of nodes in the deployment area, to see the variation when topology is denser or sparse. Three sources were programmed to communicate to a mobile sink with a data rate of $2pps$. As shown in Figure 5.8, results indicate that the EG consumes more energy as compared to the other protocols.

The packet delivery ratio of the proposed algorithm was analysed for variation of sink velocity. Sink velocity was varied from $0.5m/s$ to $20m/s$ and the PDR was calculated. Three sources were programmed to communicate at a rate of $200pps$, with a packet size of $128bytes$ to check the performance. The result shown in the Figure 5.13 indicates that the performance of dynamic SMT is much better when compared to other protocols. For pedestrian movement (velocity $\leq 2m/s$), delivery ratio of dynamic SMT approach was higher than 90%.

Test-bed based Analysis: Proof of concept of the proposed approach was also done using a hardware testbed with 25 MicaZ (as static nodes) and one Telosb (as mobile sink). Static nodes were deployed in a grid of size 5 × 5. Experiments were conducted in an area of $20m \times 20m$. Sensor nodes were programmed to communicate with each other with a transmission range of $\leq 2m$. All static nodes were pre-programmed with their positions.

FIGURE 5.16: Test-bed: Routing between multiple sources to mobile sink

Mobility or frequent change in the topology was implemented by programming B-Bot to move around the network. Data from the mobile sink was sent to a sensor node connected to a PC. Serial data from the TelosB mote was read to check the position of the sink. The Steiner points were calculated on the computer and then broadcasted in the network via the TelosB mote. Mote-PC communication was done using serial communication. The test-bed is shown in the Figure 5.16. Steiner vertex calculation and tree construction to route the data was analysed by changing the number of sources. The test-bed results indicate that the proposed algorithm can be effectively used in a network, where the sink mobility is high. Experiments were also done to calculate the PDR and receiver throughput, by varying the number of sources and by changing the packet size. Figure 5.14 indicates the PDR with respect to a source sending data at a rate of 20pps with varying packet sizes. It can be observed from the figure that PDR decreases with increase in the number of sources. Also, PDR is high for smaller packet size. Figure 5.15 indicates the receiver data-rate for the same scenario. It is clear that even though the PDR is less for a 128 byte packet, receiver data-rate is higher for a smaller packet. Decrease in the PDR and data-rate with the increase in the number of the source node is due to network interference. More the number of source nodes, more the packets in the network, which in turn result in collision and data interference and hence reduce the PDR.

5.1.3.1 Performance Evaluation of Scenario 2

Simulations were performed on Castalia to analyse the performance of the proposed protocol. Performance evaluation was done by varying the number of source nodes, data rate, and the velocity of the mobile node. Table 5.2 illustrates the parameters used in the simulation. The movement of sink node was modelled using random way-point model.

TABLE 5.2: Simulation Parameters

Parameters	Values
Network Size	$200 \times 200\ m^2$
Number of Nodes	50
Transmission Power	$-5dBm$
Velocity	$0.5m/s$ to $10m/s$
Number of Mobile Sink	3
Number of Mobile Sources	7
Packet size	$128byte$
Simulation Time	$500Sec$ (Multiple Iterations)
Mobility Model	Random Way-point

The size of the network was increased to analyse the number of hops on the route between multiple mobile sources and mobile sinks. Simulation was done with different number of nodes and is shown in Table 5.3. Table 5.3 give the number of hops required to connect 7 mobile sources and 3 mobile sinks. Speed is at $0.5m/s$.

Table 5.4 shows the PDR obtained for 7 mobile sources and 3 mobile sink with varying Packets-per-Second(pps). The mobile nodes are programmed to move at $1m/s$. As the packet per second increases, PDR decreases. One of the reasons for the drop in PDR is due to multiple sources parallel sending data. This increases the number of sources contending for the media. When the packet per second was increased from 0.5 to 50, PDR dropped from 73% to 13%.

TABLE 5.3: No.of hops to connect 7 sources and 3 sinks

No.of Nodes	No.of hops
50	19
100	26
150	35
200	43
300	58
400	71
500	87

TABLE 5.4: PDR obtained for 7 mobile sources and 3 mobile sinks by varying pps

Sender Rate(in pps)	PDR(%)
0.5	73.4
1	69.1
2	57.3
5	47.2
10	33.6
20	19.4
50	13.1

TABLE 5.5: Change in PDR with Velocity

Velocity (m/s)	PDR(%)
0.5	78.1
1	73.5
2	61.8
5	43.5
10	27.4

Performance of the proposed protocol was also done by varying the velocity. The packets per second was kept constant at 0.5. The results are tabulated in Table 5.5. As the velocity varies from $0.5m/s$ to $10m/s$, the PDR drops from 78% to 27%. This drop in PDR at higher velocity is due to the failure to update the sink in synchronization with the high mobility. The result shows that the performance is better for lower velocity.

5.2 Data Routing: Static relays

Most of the sensor network applications need a static deployment of sensor nodes, where they monitor an event and report to the base-station. In certain WSN applications such as health-care, military, managing inventory and animal monitoring the sensor network consists of a mixture of static nodes and mobile nodes [119] [120]. According to Theofanis and Panayiotou [68], an efficient mixed WSN is developed that employs a smaller number of stationary nodes that collaborate with few mobile nodes in order to improve the area monitoring.

For data communication; unlike the static WSN, mobile nodes need to discover a neighbor node before it starts transferring data. It is also important to maintain an unbroken node-to-node communication while transferring data. Due to node mobility, traditionally used protocols for static WSN cannot be adopted directly. Since WSN applications are energy-aware applications, protocols designed for Mobile-Ad-hoc networks cannot be applied directly.

Communication protocols for mobile WSN should be designed by considering the mobility pattern of the mobile node. Mobility pattern is the way in which a node moves in the sensor network and is different for various applications. Dong and Dargie classify mobility pattern as pedestrian mobility pattern, vehicular mobility pattern and dynamic medium mobility pattern [121]. A protocol developed for one specific mobility pattern may not be suitable for another. For example, protocol developed for vehicular mobility pattern for a predefined path may not be efficient for a military application [122].

Developing an energy-aware communication protocol for mobile WSN is challenging. Certain applications such as health care systems and military applications demand minimum latency during data collection. In this section, we propose a mobility-vector based neighbor selection, for communication between mobile and static nodes. The neighbor selection takes into consideration the available residual energy of the static node and its coverage time with respect to the velocity of the mobile nodes in an energy efficient manner. Once the neighbor selection is done, the mobile node establishes a binding for data transfer. The proposed approach take care of topology control and optimal neighbor selection.

5.2.1 The proposed approach

The objective of the proposed protocol is to use mobility pattern to choose a suitable node for communication. The scenario can be modeled as follows. The sensor network consists of n static nodes deployed in an area A, with m mobile nodes. The aim is to establish an effective communication between the mobile node and the static node, so as to transfer maximum data in minimum time. This can be achieved by finding an appropriate neighbor, with maximum contact time for data communication while moving. Static nodes will act as relay nodes to route the data to the base-station.

Assumptions: Following assumptions have been made here. We assume that all the nodes know their own locations. We also assume that a mobile node can estimate its mobility parameters using any of the method explained in [121]. We assume that a safe circular communication region is available, which can fit within the communication region of the node, where the packet delivery ratio within the region is 99.9%. The proposed approach is divided into three phases.
Phase I: During network initialization, static nodes update their neighbor details.
Phase II: Mobile node will select its first static node for data communication.
Phase III: After phase II, the mobile node will find optimal static nodes on its path to continue the data communication. The proposed approach is a de-centralized routing approach.

5.2.2 Network initialization

Network initialization is one time process and is done at the beginning, immediately after the node deployment. All static nodes are programmed to a sleep time of t units, with a wake time of $\Delta t(\Delta t << t)$ time units.

The value of t is chosen based on the maximum velocity of the mobile nodes (V_{max}) and the communication radius (r). A node with *fast mobility* will spend less time in the communication range of a static node, when compared to a mobile node with *slow mobility*, which takes the same path. Hence, for a high speed mobile node, a small value of t is preferred. The value of t also depends on the communication range. Contact time for a mobile node is high for a large value of r and hence value of t is directly proportional to the communication range.

From the above observations, t can be modeled as:

$$t \propto \frac{1}{V_{max}} \tag{5.1}$$

Also

$$t \propto r \tag{5.2}$$

Therefore, from 5.1 and 5.2

$$t \propto \frac{r}{V_{max}} \tag{5.3}$$

Or,

$$t = K\frac{r}{V_{max}} \tag{5.4}$$

Where K is a constant. Depending on the communication radius of the static node and the velocity of the mobile node, the value of t can be determined using equation 5.4.

Nodes are not time synchronized and the sleep wakeup pattern will start immediately after the neighbor discovery. During the network initialization, static nodes find their neighbor nodes via one hop broadcast. Each node keeps a record of its neighborhood information such as the node location and their wakeup time from the received message. Static node uses the set $T = \{T_1, T_2, T_3, ..., T_k\}$, where $T_1, T_2, T_3, ..., T_K$ represent the wakeup time of k neighboring nodes. This value is stored with respect to its own local clock.

5.2.3 Selecting a primary node

After *network initialization* phase; if a mobile node has data to send, it switches on the radio and searches for a static node to which the data can be transferred. Since the static nodes follow a sleep-wake pattern (which is not time synchronized with its neighbors), finding an optimal neighbor is difficult. Objective of Phase II is to select a *primary node* for data communication. For a data communication session, the node that the mobile node initially selects is termed as the *primary node*. Primary node will in-turn wake up its neighbors, while communicating with the mobile node.

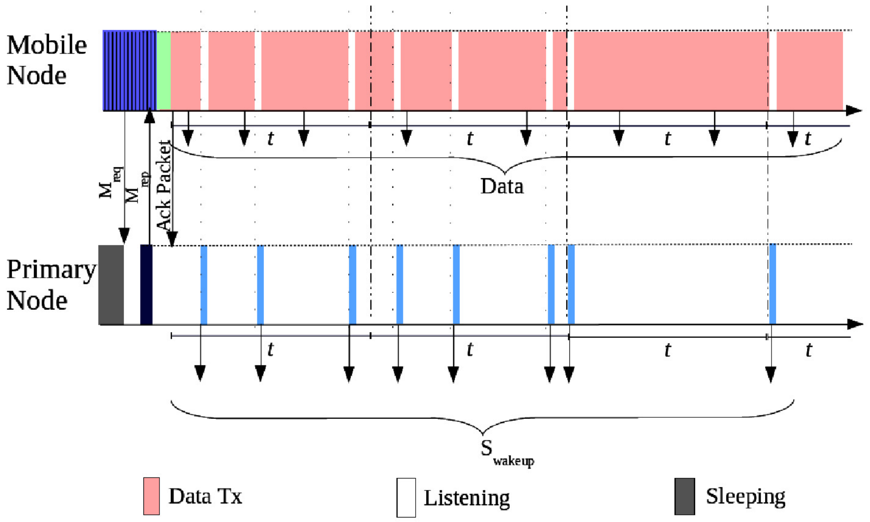

FIGURE 5.17: Timing Diagram: Primary Node Communication

Timing diagram shown in Figure 5.17 explains the communication establishment between a mobile node and a primary node. When a mobile node has data to send, it turns on its communication module and search for a static node by sending a broadcast packet M_{req}, with an interval less than Δt. M_{req} message contains the mobility vector information such as node's current location $\overrightarrow{d_m}$, its velocity $\overrightarrow{v_m}$ and the amount of data needed to be transmitted.

When a static node receives M_{req} message, it calculates the *contact time* T_s, with its current location information and with mobility parameters of the mobile node. If $T_s > t$, it accepts the binding request by sending M_{rep} message. Static node calculates priority p, based on the value of T_s. M_{rep} message is sent after a short interval t_{wait}, where $t_{wait} = f(T_s)$. The value of t_{wait} depends on the priority p calculated. For higher p value, t_{wait} should be shorter and vice versa. So,

$$t_{wait} \propto \frac{1}{p}.$$

This is done to avoid collision, in case if multiple static nodes receives M_{req} message at the same time.

M_{rep} message contains the wakeup time information T' and data channel information (a random channel from $|C-1|$ channels to communicate). T' indicate the set of time slots of those neighboring nodes, which are in the direction of the mobile node movement within a period of time t. The mobile node accepts the connection to the first M_{rep} message it receives and continues communicating via the mutually agreed channel. Details of the contact time calculation is explained in Section 5.2.4.1.

The node bounded to the mobile node needs to turn on its neighbors. To guarantee a smooth handover of the mobile node data, it broadcasts S_{wakeup} message to awake its neighbors. This is sent by the static node which is currently communicating with the mobile node. The S_{wakeup} messages are sent with respect to the set T' for a period of $2t$ time, and henceforth will broadcast only once. If a static node which is awakened, is not receiving S_{wakeup} for a period of $3t$ time, it will switch to sleep mode and continues the sleep wake pattern for the same interval it followed earlier. Time t shown in Figure 5.17 is with respect to the primary node. Since $T' \subseteq T$, primary node needs to send only few S_{wakeup} messages to wake up its neighbors. It should be noted that, while a static node broadcasts S_{wakeup} messages, the mobile node needs to halt its communication. Static node uses M_{rep} to update this information to the mobile node. The mobile node halts the data communication based on the value of T', so that the static node can send S_{wakeup} messages. Static node switches its channel from data channel to control channel and vice versa to send the broadcast messages.

The structure of the messages M_{req}, M_{rep} and S_{wakeup} is as follows,

$$M_{req} : \langle \overrightarrow{v_m}, \overrightarrow{d_m}, data\_size \rangle$$

$$M_{rep} : \langle lease, t, T' \rangle$$

$$S_{wakeup} : \langle Broadcast, \overrightarrow{v_m}, \overrightarrow{d_m} \rangle$$

5.2.4 Selecting an optimal node

The third phase deals with node selection for seamless handover. Once the mobile node moves away from the range of a *primary node*, it needs to find an optimal neighbor for continuing the data communication. Like Phase II, here also we propose to use mobility vector information of mobile node to choose a neighbor. Since nearby static nodes are awakened via S_{wakeup} messages, mobile node can choose an optimal static node to which it can have a maximum contact time with.

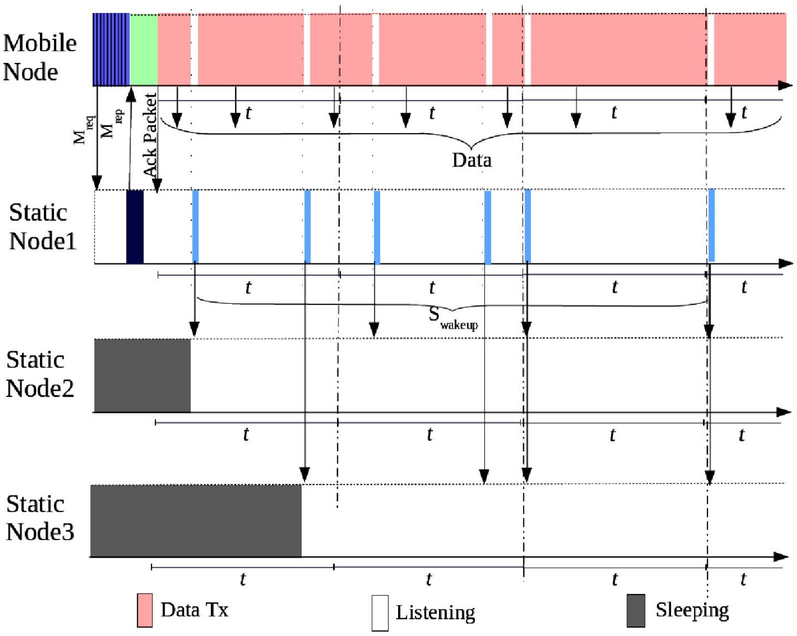

FIGURE 5.18: Timing Diagram: Optimal node Selection

In Phase III, similar to Phase II, we calculate T_s and t_{wait} and the steps are shown in Algorithm 1. Algorithm 1 explains the calculation of T_s and t_{wait}.

The value of T_s needs to be at least t, as it will take t time units for a static node to ensure that all its neighboring nodes receive the S_{wakeup} messages, and thereby ensuring that the handover process proceeds smoothly, in case the mobile node moves out of the range of the static node.

As shown in Figure 5.17 and Figure 5.18, mobile node will halt its communication for a time δt, when the static node broadcasts the S_{wakeup} messages. This synchronization is done with the help of set T' it received from each static node, while binding via M_{rep}

message. Let t_d be the one hop communication delay and $T'_1, T'_2 \dots T'_{K'}$ be the time stamp information in set T', where $K' \leq K$. For each value of T'_i in the set T', mobile node pauses its communication at $t_d + T'_i$. Since we are broadcasting the data for two iterations of time t, with respect to the static node, the pause time is calculated as $t_d + T'_i$, for the first iteration and $t + t_d + T'_i$, for the second iteration. For the subsequent iterations, the mobile node will halt its communication at the beginning of each time slot. Hence the time slot at which the mobile node pauses the data transfer is $\{t_d + T'_i\}, \{t + t_d + T'_i\}, 3t, 4t, \dots$, for all values of T'_i, where t corresponds to the clock cycle of the static node. Correspondingly, static node which is binded to the mobile node will send S_{wakeup} message at an interval of $\{T'_i\}, \{t + T'_i\}, 3t, 4t, \dots$, for all values of T'_i, till the mobile node is in the range.

5.2.4.1 Contact Time Calculation

Here, the aim is to select a node that can accept the maximum amount of data packets. This is possible if the contact time between the mobile node and the static node is high. If the location and the direction of the mobile node is known, a static node can calculate the duration in which the node is going to be in its communication radius. When a mobile node broadcasts the M_{req} packet, it includes fields in the message header such as the direction, speed and the location. The static node uses these values to calculate the contact time, which in turn can be used to calculate the amount of data the static node can accept.

A geometrical way of solving this problem is to find the length of the chord that the mobile node's path will intersect in the communication range of the static node. Given the direction and the location of the mobile node, static node can calculate the length of the trajectory and the point at which the mobile node will intersect in the circumference of the communication radius. Contact time can be calculated using the current speed of the mobile node and from the length. Figure 5.19 shows the scenario of a mobile node, whose present location is at d_m, moving with a velocity v_m. The mobile node is currently under the communication range of a static node at location d_s whose communication radius is R_s. Figure 5.20 shows the geometrical model represented in the form of a triangle with respect to the co-ordinates d_m, d_s and P, where P is the point at which the mobile nodes will intersect the circumference of the circle with radius R_s in the future. The aim is to find the distance between d_m and P and hence find the contact time using V_m.

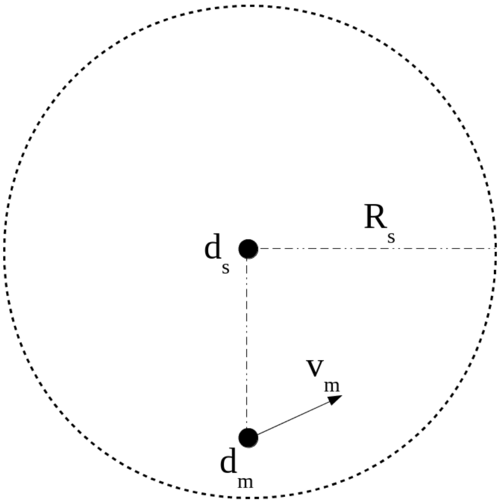

FIGURE 5.19: Geometric view and the corresponding vectors

The notations used in the mathematical model are given in Table 5.6.

Contact Length Estimation:

Given $\overrightarrow{d_s}$ and $\overrightarrow{d_m}$ we have,

$$\overrightarrow{d_{rel}} = \overrightarrow{d_s} - \overrightarrow{d_m} \tag{5.5}$$

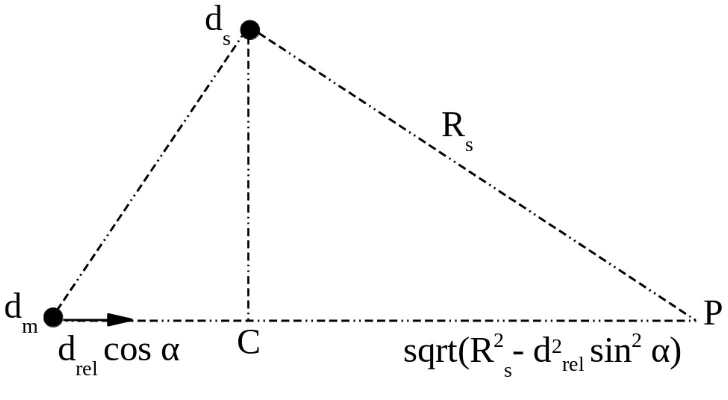

FIGURE 5.20: Geometric relation between the chord length and vectors

Table 5.6: Symbol Reference Table

Symbol	Description
$\overrightarrow{v_m}$	Velocity of the mobile node
$\overrightarrow{d_s}$	Location of the static node
$\overrightarrow{d_m}$	Location of the mobile node
$\overrightarrow{d_{rel}}$	Relative position of the static node with respect to the mobile node
t_s	Time to travel along the chord
β	Size of data that can be accepted
β_{max}	Maximum amount of data that can be a can be accepted in a single lease
θ	Angle that v_m makes with the $x - axis$ of the frame of reference
α	Angle between v_m and d_{rel}
R	The rate at which the static node can accept data
RE_i	The i^{th} node residual energy
t_{wait}	Waiting time before a static node sends a *bind-acknowledgement* packet

From equation 5.5 and $\overrightarrow{v_m}$, we can calculate the angle α by calculating the dot product.

$$cos\ \alpha = \frac{\overrightarrow{v_m} \cdot \overrightarrow{d_{rel}}}{|\overrightarrow{v_m}||\overrightarrow{d_rel}|} \tag{5.6}$$

From the Figure 5.20, the length of the line $\overline{d_m, P}$ can be calculated from the equation

$$l = d_{rel}cos\alpha + \sqrt{R_s^2 - d_{rel}^2 sin^2\alpha} \tag{5.7}$$

and therefore, the contact time T_s can be calculated as follows

$$T_s = \frac{l}{|\overrightarrow{V_m}|} \tag{5.8}$$

So the amount of data that can be accepted with a data rate R for a time T_s is:

$$\beta = T_s R \tag{5.9}$$

Since we are working with a random way-point model, it is rational to impose an upper bound on the amount of data that a node can offer to accept as the lease shall expire if there is a change in velocity. We call this limit β_{max}, and therefore, in the cases where $\beta > \beta_{max}$, we set $\beta = \beta_{max}$.

Algorithm 5.2 From Mobile Node Perspective

```
procedure MOBILE_NODE
    INPUT: $\overrightarrow{v_m}, \overrightarrow{d_m}$
    while ∃ outGoingPacket do
        if is_valid(currentLease) then
            Send the packet to associated node as lease-followUp
        else
            bind-request := $\langle \overrightarrow{v_m}, \overrightarrow{d_m} \rangle$
            broadcast(bind-request)
            Switch to Receive mode
            currentLease ← receivedLease
            Send the packet to the associatedNode
        end if
    end while
    currentLease.data := currentLease.data - outGoingPacket.size
end procedure
```

Lease and Priority calculation

The amount of data that a static node can accept is calculated from the equation 5.9. After requesting M_{req} message, each static node calculates a *priority value*, that is proportional to the lease time it can offer. Calculation of priority value not only on the value of β, but also depends on the residual energy RE_i, that the static node possess. A static node with less energy level should generate a lower priority number and vice versa.

$$priority = \gamma_1 \beta + \gamma_2 RE_i,$$

where γ_1, γ_2 are the weights associated with the parameters β and RE_i. Priority for a static node will be directly proportional to the amount of data that it can accept. Higher the value of priority number, more precedence it has over the other nodes. In order to avoid simultaneous reply, static nodes wait for a small period of time before sending M_{rep} message. The wait time before the acknowledgement to the bind-request is sent, t_{wait}, is inversely proportional to the priority number, allowing nodes with higher priority

Algorithm 5.3 From Static Node Perspective

```
procedure STATIC_NODE
    INPUT: $\overrightarrow{d_s}, R_s$
    while true do
        if Channel is free then
            Go back to sleep
        else
            if inComingPacket.type = bind-request then
                $\langle lease,\ t_{wait} \rangle :=$
                $calculateLease(\overrightarrow{d_m}, \overrightarrow{v_m}, \overrightarrow{d_s}, R_s)$
                if $T_s < t$ then
                    abort
                    Go back to sleep
                end if
                pause($t_{wait}$)
                Send $S_{wakeup}$ w.r.t appropriate sleep time
                if No bind-Acknowledgment for same node is overheard then
                    Send out bind-acknowledgment
                end if
            else
                if inComingPacket.type = lease-followUp then
                    Pass the packet to upper layer
                elseinComingPacket.type = $S_{wakeup}$
                    Wait for $3t$ time units
                    Sleep if no activity is detected
                end if
            end if
        end if
    end while
end procedure
```

value to establish a binding first.

$$t_{wait} \propto \frac{1}{priority}$$

The mobile node will bind to a static node from whom it receives the M_{rep} message first. It accepts the bind request by sending an *ack* packet, and then starts communicating in the channel, where both the nodes agreed upon. Changing velocity, i.e., either speed or direction invalidates the lease. Otherwise, the mobile node continues to send the data, till it crosses the communication radius of the static node. In case of an expired/invalid lease, the mobile node sends out a broadcast message M_{req}, and the whole process is repeated. When a static node receives M_{req} message, it calculates the lease, and waits for t_{wait}, before replying back to the mobile node with a M_{rep} packet. Once the static node receives ACK packet, the static node switches to the channel it agreed upon and starts listening to the data packets. When the time for *free slot* comes, the mobile node hold its communication, while the static node switches back to the control channel, sends S_{wakeup} message and switches back to the channel it was listening for data communication. Mobile node continues to send the data packet till the next *free slot*.

Algorithm 5.2 describes the basic procedure that a mobile node shall follow, when it receives any data packet from the upper layer to transmit. On the static node side, Algorithm 5.3 is followed to complete the protocol.

5.2.5 Performance Evaluation

The proposed protocol was simulated and different metrics are compared with a similar protocol, MA-MAC proposed by Zhiyong et.al [123]. The approach proposed in [123] uses RSSI value to select a static neighbor for data communication. Packet delivery ratio and data rate was calculated by changing the data rate of sender. The capacity at which the protocol communicates can be assessed by analyzing the throughput. Simulation were done using Castalia. We considered an area of $200m \times 200m$, which consists of a random deployment of both static nodes and mobile nodes. Table 5.7 shows the parameters used in the simulation. Simulation was done for low speed scenario (pedestrian movement). The mobility model used for the node movement is random way-point mobility model.

Latency of proposed approach was compared with MA-MAC and is shown in Figure 5.21. Average latency of MA-MAC protocol is higher than the proposed approach. This is because of the acknowledgment packets, the receiver sends to the sender. While in the proposed protocol, the sender continues to send the packets, after establishing a binding. Initial binding delay is almost the same in both the cases to choose a neighbor.

TABLE 5.7: Simulation Parameters: Mobility Vector based Neighbor Selection

Parameters	Values
Network Size	$200 \times 200\ m^2$
Number of Nodes	80
Max Velocity	10m/s
Min Velocity	0.5m/s
Packets per Second	10pps to 200pps
Packet size	128byte

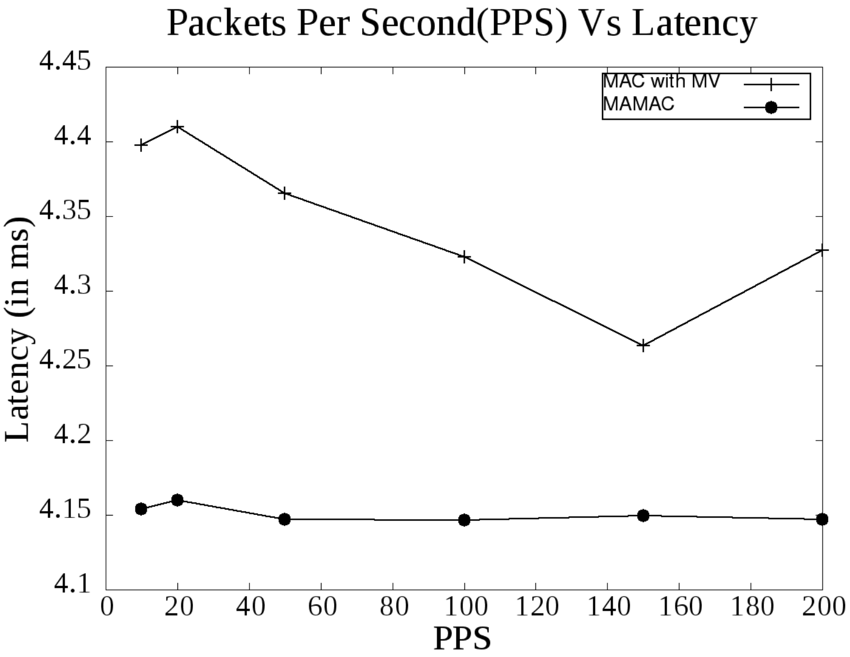

FIGURE 5.21: PPS Vs Latency

Delay in finding a neighbor node in Phase II is high for the proposed approach than in MA-MAC protocol. This is due to the waiting time for sending the M_{rep} message.

As the velocity of the node increases, the topology changes continuously. To study the influence of node velocity, the packet delivery ratio of MA-MAC and the proposed approach was calculated by changing the velocity. The node velocity was varied from $0.5 m/s$(pedestrian speed) to $10 m/s$. It is clear from Figure 5.22 that our approach performance is much better than the MA-MAC approach. The number of nodes contacted is also calculated by changing the velocity. It is clear from Figure 5.23 that as the velocity increases, the number of nodes contacted also increases in both the approaches for the same simulation time. This is due to the change in network topology. It is also clear

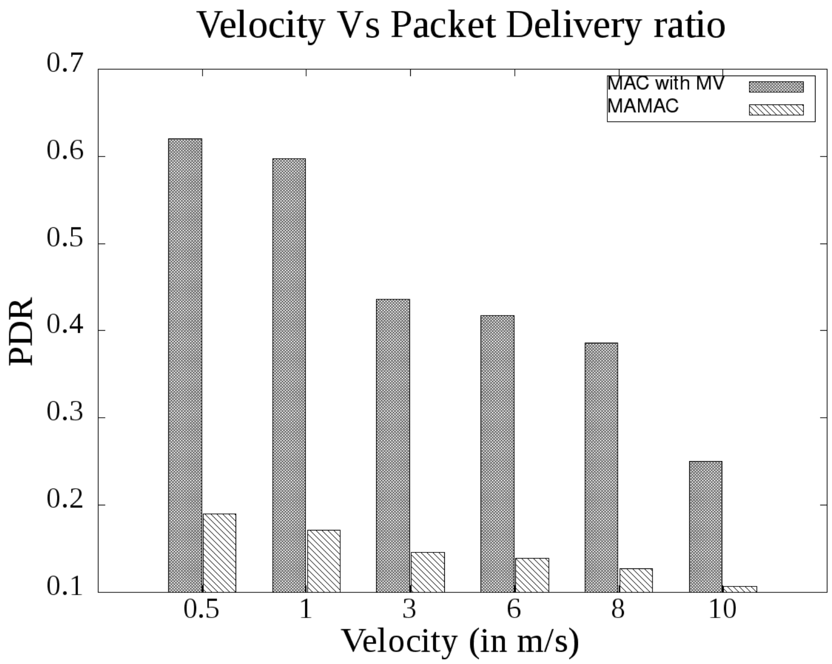

FIGURE 5.22: Velocity Vs PDR

that the number of nodes contacted is less for the proposed approach. This is because for each binding, the node will bind to the maximum time.

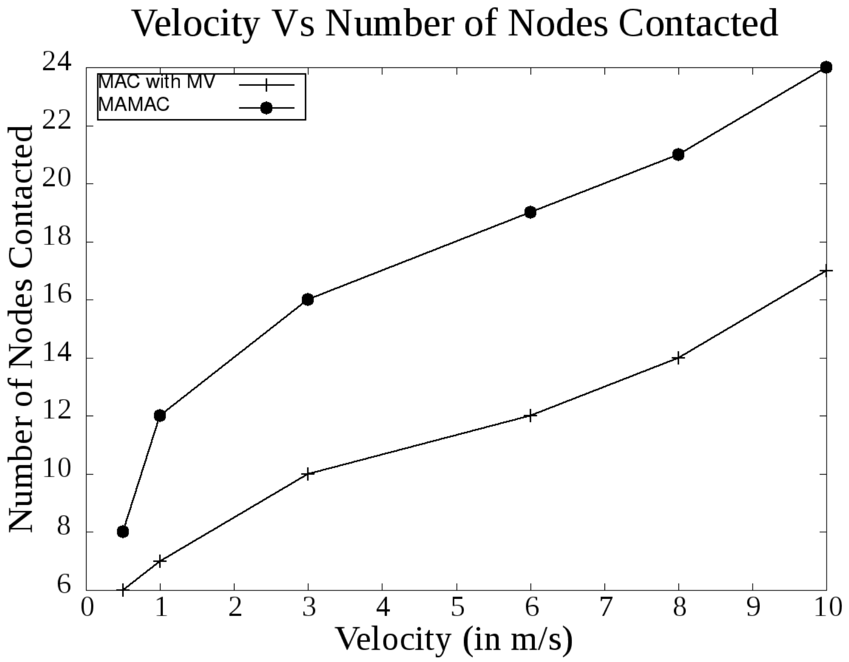

FIGURE 5.23: Velocity Vs No of Nodes Contacted

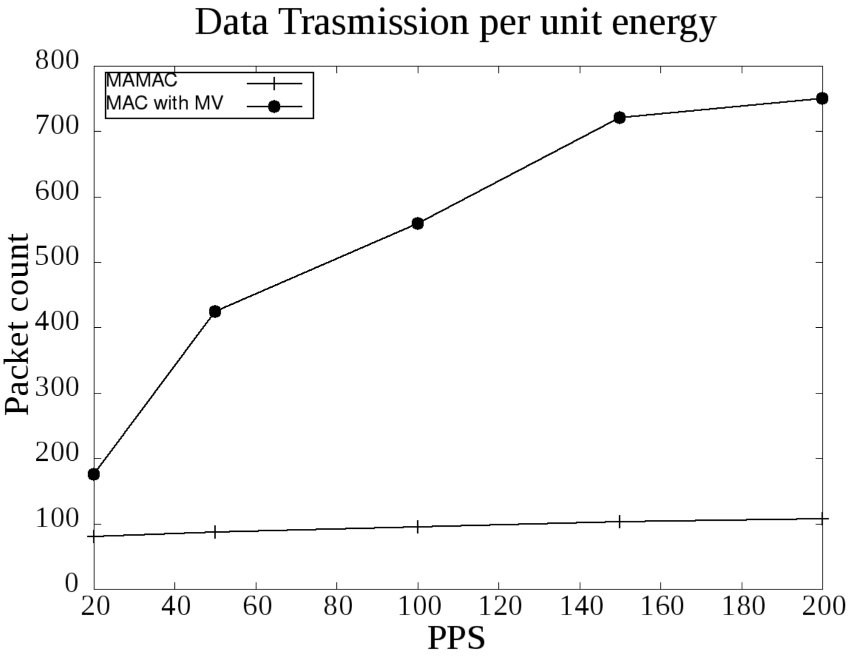

Figure 5.24: Velocity Vs Data Transmission per unit energy

Figure 5.24 compares the packet rate with the average energy consumed per successful packet delivery. It is clear from the figure that the proposed approach outperforms MA-MAC with a substantial margin. Moreover, it becomes evident that on higher packet rates, the performance of MA-MAC tends to deteriorate. The reason for the lower performance of MA-MAC can be attributed to the lower delivery ratio. The proposed approach on the other hand, performs better since it is able to utilize the lease efficiently, and thus reducing the number of dropped packets.

The communication overhead of the proposed approach is less when compared to MA-MAC because of the less number of control packets (including ACK packets). In the proposed approach, for continuous handover, the communication overhead is less, as the *primary node* will wake up all its neighbors and hence helps in smooth handover. This is done by sending S_{wakeup} message to wake up all its neighbors. For the MA-MAC, the control packets needed for the initial binding is high, and the overhead caused by the acknowledgement packets is substantial too. Moreover, the number of neighbors that the mobile node needs to bind is more for MA-MAC, for sending the same amount of data. This is because MA-MAC selects a neighbor from which it receives the first reply from, rather than choosing an neighbor with maximum contact time. Figure 5.25

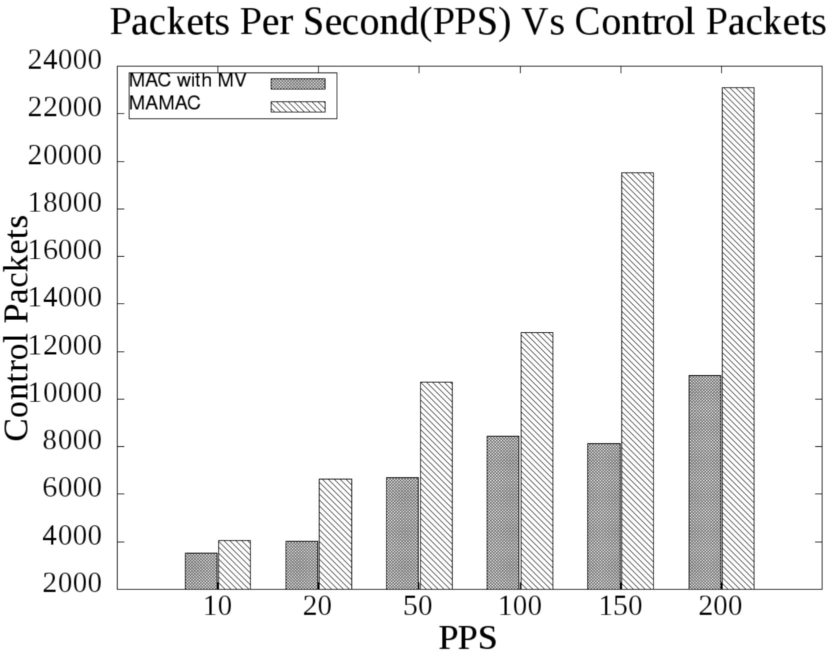

FIGURE 5.25: Controlpacket overhead

shows the control packet overhead for both the approaches. It is clear that the number of control packets needed for communication is less for our approach. In both approaches, the number of control messages depend on the number of nodes that the mobile node bind for sending the data, which is less for the proposed approach. In the proposed approach, the number of control messages depend on the neighboring nodes available, while for MA-MAC, it depends on the data packets communicated.

Figure 5.26 shows the delivery ratio of MA-MAC protocol and our approach. The packet delivery ratio is the fraction of the number of packets received with respect to the number of packets sent. The packet delivery ratio of the proposed algorithm was calculated by varying the packet generating rate from $10pps$ to $200pps$. Simulations were done for a low speed moving object ($1m/s$) with a sleep-wake time (t) of $1sec$ and δt time of $50msec$. In Phase I, mobile node binds to a *primary node* for data communication. Mobile node needs to wait for a maximum of t seconds to choose a primary node. When the mobile node moves out of the range of the primary node, Phase II starts. Data communication continues immediately after selecting a suitable neighbor. The plot shows the packet delivery ratio for a data communication session averaged over multiple simulation runs.

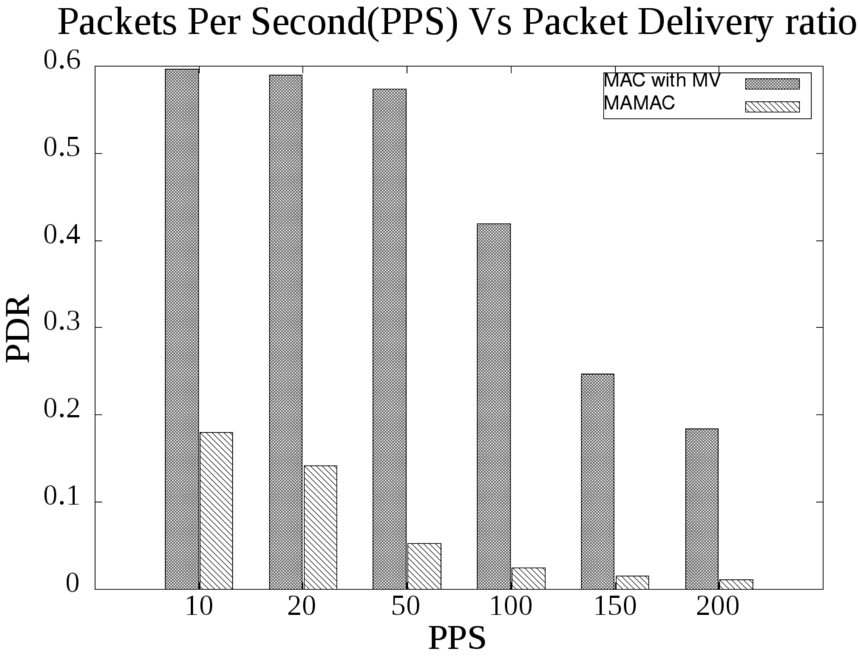

Figure 5.26: Packet Deliveriy Ratio Comparison

From the Figure 5.26, it clear that the packet delivery ratio of the proposed protocol is higher, when compared to MA-MAC protocol. This is because in the proposed approach, the mobile node will bind to an optimal neighbor (to which it can communicate for a longer duration), rather than selecting a neighbor randomly. Moreover, multi-channel approach for data communication creates less interference, when compared to MA-MAC protocol. It is also clear from the Figure 5.26 that the packet delivery ratio decreases for both the protocols with increase in sending rate. This is due to packet interference and queuing.

The current implementation does not focus on a network level time synchronization. After initialization, each node calculates the adjacent node's wake up time with respect to its current clock, and hence follow a sleep time of t seconds. For issues such as clock drift; nodes need to adjust the clock with respect to the neighbor node, so that the wakeup process during the Phase-I can go smoothly. This can be achieved by broadcasting a request packet for t Sec after τ time, where $\tau >> t$.

5.3 Data Routing via Mobile relays

Certain application such as military application and habitat monitoring, routing need to be done via mobile nodes with random mobility. This means, mobile nodes will act as a relay nodes to transfer the data from a source to sink. In this section we attempt to address this scenario by assuming that sink and source are static and all the intermediate nodes are mobile.

5.3.1 Proposed Routing protocol

Problem Definition

A sensor network consists of M mobile nodes, relays each moving randomly at velocity v_i at an angle of θ_i within the area A. The base-station is placed at location (x_b, y_b). All the nodes follow a sleep-wake pattern to improve energy efficiency. It is assumed that all nodes are localized. We have assumed a dense deployment.

A sleep-wake pattern has been used to reduce energy consumption. A node sleeps for T time and is awake for ΔT time, where $\Delta T \ll T$. In the proposed protocol, we have established communication by creating a *rounded rectangle region* (stadium shape) R between the source N_s and the base-station BS.

The protocol selects an optimal node to forward the packet based on the *contact time* calculated using *mobility vector* information. The neighbor selection protocol is presented below. The protocol has two phases, *i) Signaling Phase ii) Data transmission phase*

5.3.1.1 Signaling Phase

When the node has data to transmit, it starts with the Signaling Phase. Objective of the signaling phase is the formation of an active region between source and the base-station. All nodes in the active region will be awake. During this phase, the nodes use the source and destination location thus creating a virtual region connecting the source and destination. The region resembles a geometric stadium shape that has a circle of radius r' cut in half through the center. The two ends are then separated by a rectangle with a side length of $\hat{a}$.

The area formed by the region [124] is given as $\hat{A} = \pi r'^2 + 2r'a$. Nodes are classified into *out-area* and *in-area* based on their positions with respect to the stadium. The nodes that are in the region $\hat{A}$ are classified as *in-area*. If a node $x \in A : x \notin \hat{A}$, is classified as *out-area* nodes. The classification of nodes is shown in Figure 5.27. The source and the base-station are center of the circular region. Initially, every nodes starts as an *out-area node*.

All the nodes that are classified as *out-area* follow a sleep-wake pattern. While awake, they broadcast participation request P_{req} message.

If any node that is classified as *in-area* receives P_{req} message, it sends a participation reply P_{rep} message after a random wait time that is less than Δt. The source and destination information contained in P_{rep} message is used by a node to decide whether it belongs to the region R. A node keeps its radio on until it moves outside of the region R.

5.3.1.2 Data transmission phase

Data communication phase starts after the signaling phase. During data communication, only active nodes participate (in-area nodes). The signaling phase is used to wake up all the nodes in the region R. These nodes then forward data during the communication phase. Each data packet header contains the source and sink position in the S_{pos} and D_{pos} fields. Mobile node uses this field to recalculate the active region dynamically. If a node has data to transmit, it needs to find a neighbor for communication. A sender node broadcasts data request $data_{req}$ packet to all its neighbors. All the active nodes, who receives the packet reply back to the sender after a small time ρt. The value of ρt

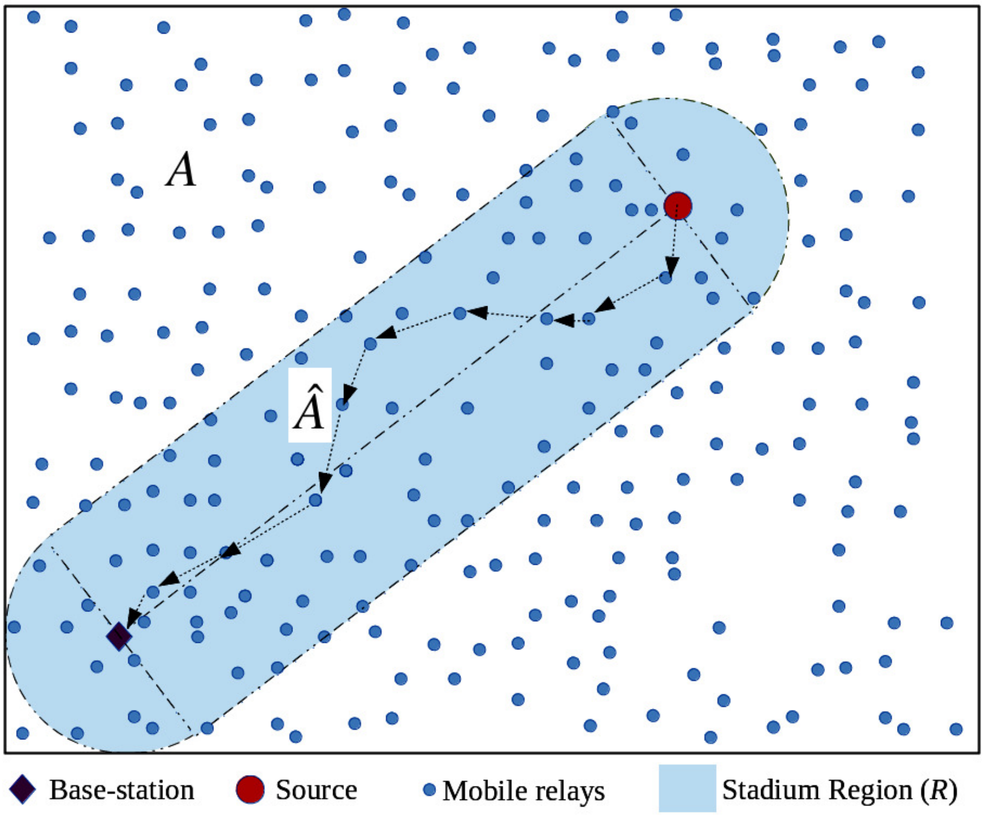

FIGURE 5.27: Source to Sink communication via mobile Relays

is inversely propositional to the *priority* value of each neighbor.

$$\rho t \propto \frac{1}{priority}$$

The priority is determined based on how long a node will remain within transmission range of the neighbor that sent the $data_{req}$ packets and its position with respect to the location of the destination. Neighbors that overhear the $data_{rep}$ message will refrain from sending a reply message themselves. The data is then sent to the mobile node selected. The data can be sent for upto $\hat{t}$ units of time. This process is repeated at each node with data to send. The equations and the method used for calculating the time for which two nodes will remain with transmission range (contact time $\hat{t}$) is described in the following sub section.

Contact Time Calculation

The contact time of two mobile nodes n_1 and n_2 is the time for which the two nodes remain in transmission range r. Let $\overrightarrow{X_1}$ be the position vector of node n_1 at time t. $\overrightarrow{X_1(t)}$ is the position vector for node n_1. Similarly $\overrightarrow{X_2}(t)$ is the position vector for node n_2. Scenario is shown in Figure 5.28

Let t_0 be the instant from which the value $\hat{t}$ is to be calculated. Then,

$$|\overrightarrow{X_1}(t_0) - \overrightarrow{X_2}(t_0)|$$

is the distance between the nodes initially and this will be lesser than the transmission range r of the nodes. Let t' be the time when the nodes move beyond the communication range. That is

$$|\overrightarrow{X_1}(t') - \overrightarrow{X_2}(t')| \tag{5.10}$$

$$t' = \min t \quad | \quad |\overrightarrow{X_1}(t) - \overrightarrow{X_1}(t)| > r \tag{5.11}$$

Then, $t' - t_0$ is the contact time $\hat{t}$, for nodes n_1 and n_2

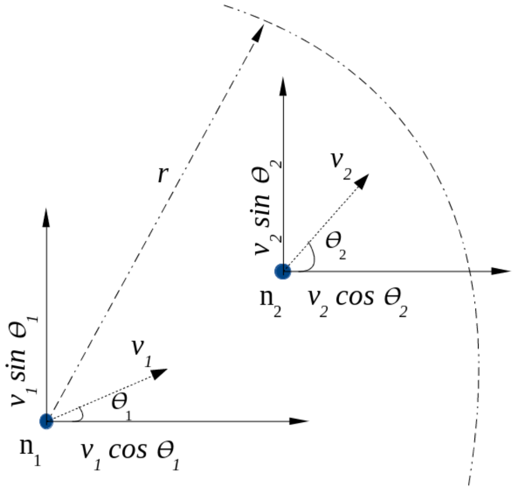

FIGURE 5.28: Scenario of two mobile nodes moving with velocity v_1, v_2

Thus, relative velocity v_r of the two nodes is $\overrightarrow{v_r} = \overrightarrow{v_1} - \overrightarrow{v_2}$ Along x direction,

$$v_{r(x)} = v_1 \cos\theta_1 - v_2 \cos\theta_2,$$

Where $v_1 = |v_1|, v_2 = |v_2|$ Along y direction,

$$v_{r(y)} = v_1 \sin\theta_1 - v_2 \sin\theta_2$$

Now, The magnitude and relative velocity can be obtained using the equation

$$\sqrt{v_{r(x)}^2 + v_{r(y)}^2} \tag{5.12}$$

Direction of relative velocity is:

$$tan^{-1}\left(\frac{v_1 \sin\theta_1 - v_2 \sin\theta_2}{v_1 \cos\theta_1 - v_2 \cos\theta_2}\right) \tag{5.13}$$

$$= tan^{-1}\left(\frac{v_{r(x)}}{v_{r(y)}}\right) \tag{5.14}$$

Initially, nodes n_1 and n_2 are at distance d_0 at time t_0.

$$|\overrightarrow{X_1}(t_0) - \overrightarrow{X_2}(t_0)| = d_0 \tag{5.15}$$

where $d_0 < r$, Time t' is given by the equation

$$t' = \min t \quad | \quad |\overrightarrow{X_1}(t) - \overrightarrow{X_1}(t)| > r \tag{5.16}$$

Hence, the contact time is;

$$(t' - t_0) = \left(\frac{r - d_0}{\sqrt{v_{r(x)}^2 + v_{r(y)}^2}}\right) \tag{5.17}$$

If (x_1, y_1) are the coordinates of nodes n_1 at time t_0 and (x_2, y_2) are the coordinates of node n_2 at time t_0. d_0 is the Euclidean distance between n_1 and n_2 at time t_0 it is given by the equation

$$d_0 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{5.18}$$

Contact time will be the time required for the nodes to move beyond the range r. Initially, n_1 and n_2 are d_0 distance apart. The relative velocity causes them to move apart by a distance $r - d_0$.

Hence $contact\_time = \frac{r-d_0}{|v_r|}$ where $|v_r|$ is the magnitude of relative velocity.
Hence $\hat{t}$ can be obtained using the equation

$$\hat{t} = \left(\frac{r - \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{\sqrt{(v_1 \cos\theta_1 - v_2 \cos\theta_2)^2 + (v_1 \sin\theta_1 - v_2 \sin\theta_2)^2}} \right) \tag{5.19}$$

The algorithm for the proposed approach is given below.

Algorithm 5.4 Start Up function

```
procedure START(node)
    if node has data then
        Call CALCULATE_NEXT_HOP(node)
    else
        if node ≠ source && node ≠ destination then
            call SET_TIMER(node, Check_area)
        end if
    end if
end procedure
```

Algorithm 5.4 checks whether a node has data to transmit. If yes, it uses Algorithm 5.5 to select the next hop. If there is no data to send, the node will sleep if outside the active region. If the destination is within the range of a sending node, it directly forwards the data to the destination. Otherwise, the sender selects a node based on the highest $\hat{t}$ value for forwarding the data. The contact time calculation is given in Algorithm 5.6. Algorithm 5.6 uses the equation 5.19 for calculating the contact time $\hat{t}$.

Algorithm 5.5 Next Hop Calculation

```
procedure CALCULATE_NEXT_HOP(node)
    max = NEGATIVE_VALUE
    for all $i$ do
        Calculate $\hat{t}$ = CONTACT_T_CALC(node, i),    where $i$ is neighbor(node)
        if $max < \hat{t}$ then
            $max = \hat{t}$
        end if
    end for
    if $d(node, destination) < r)$ then
        node→contact_time=CONTACT_T_CALC(node, destination);
        node→nexthop = destination
    else
        node→contact_time=max
        node→nexthop = hop
    end if
end procedure
```

Algorithm 5.6 Contact Time Calculation

```
procedure CALCULATE_T_CALC($n_1$, $n_2$)

    if $d(n_1, n_2) < d(n_1, dest)$ && $d(n_2, dest < d(n_1, dest))$ then
        $a = v_1 \cos\theta_1 - v_2 \cos\theta_2$,
        $b = v_1 \sin\theta_1 - v_2 \sin\theta_2$
        $c = x_1 - x_2$
        $d = y_1 - y_2$
            $$\hat{t} = \left( \frac{r - \sqrt{c^2 + d^2}}{\sqrt{a^2 + b^2}} \right)$$
        if $r - d(n_1, n_2) > 0$ then
            return 0
        else
            return NEGATIVE_VALUE
        end if
    end if
end procedure
```

5.3.2 Results

To verify and analyse the behavior of the proposed neighbor selection, protocol simulations were done in Castalia. Very few protocols are available in literature to support energy efficient routing in mobile WSN. In order to compare our proposed approach, we choosen Receiver-based Opportunistic Forwarding protocol (ROF) and Greedy Perimeter Stateless Routing (GPSR) because of low complexity, no global table needed (both uses the local topology information to select the immediate neighbor for communication) and ease of implementation.

Receiver based routing protocol does not need to establish global routing between source and destination. ROF select a neighbor node of the sender to contend for the forwarding right. ROF protocol optimized the forwarding priority calculation and designed a dual-channel based forwarding right contention mechanism, which dealt with the data collision and forwarding delay.

GPSR uses information about a sender's immediate neighbor in the network topology to make decisions. In GPSR, when a packet reaches a region where greedy forwarding is impossible; the algorithm recovers by routing around the perimeter of the region. For a dynamic topology with mobile nodes, GPSR can quickly find correct new routes quickly.

Analysis was done by varying the number of source nodes, data rate, and velocity of the node. Simulations were done using a single source - destination pair. The sensor field area was set to $200m \times 200m$ with 55 mobile nodes. Random-waypoint mobility model was used to determine node positions. The packet size was set to 128 bytes.

Figure 5.29 shows the PDR of the proposed protocol, ROF and GPSR. Nodes were programmed to move at a speed of $0.5m/s$ with a sender rate varying from $10pps$ to $200pps$. Results show that proposed method yields a better PDR as compared to other protocols. GPSR has the minimum PDR as per the result.

To analyse and validate the protocol further, node velocity was varied. The velocity was varied from $0.5m/s$ to $10m/s$ and source was programmed to send packets at a rate of $20pps$. PDR was calculated for each approach and the results are shown in Figure 5.30. Results indicate that proposed protocol performs better.

Efficiency of the proposed method was also evaluated by comparing the number of nodes within contact range of each other while varying velocity. Simulation was done for 300 seconds and the average of multiple iterations was used for analysis. Since the

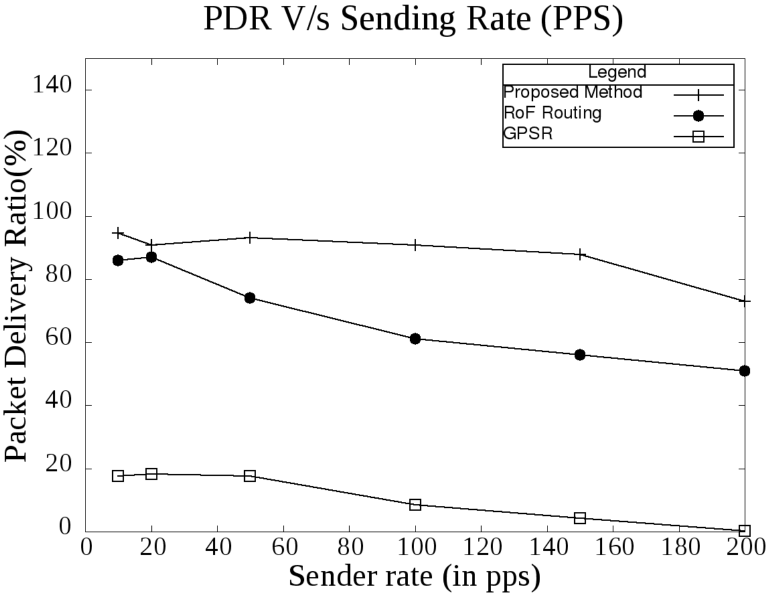

FIGURE 5.29: PDR vs Sending Rate

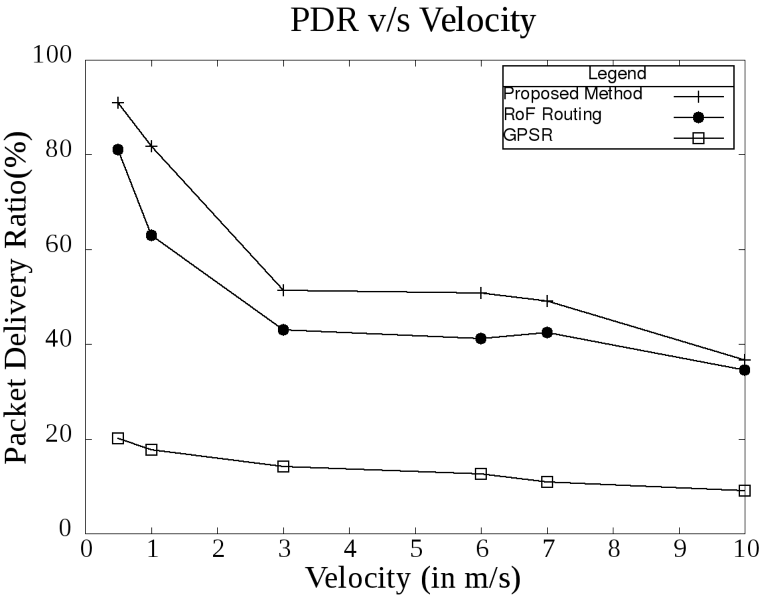

FIGURE 5.30: PDR vs Velocity

proposed approach select a neighbor node with maximum contact time, the number of nodes participating for entire data communication is less as compared to ROF. The results are plotted in Figure 5.31. It can be observed that as the velocity increases, the number of contacted mobile nodes also increases. This is due to the highly dynamic topology. The average contact time for each node calculated. Comparison of the proposed protocol and ROF in terms of contact time is shown in Figure 5.32. Since the

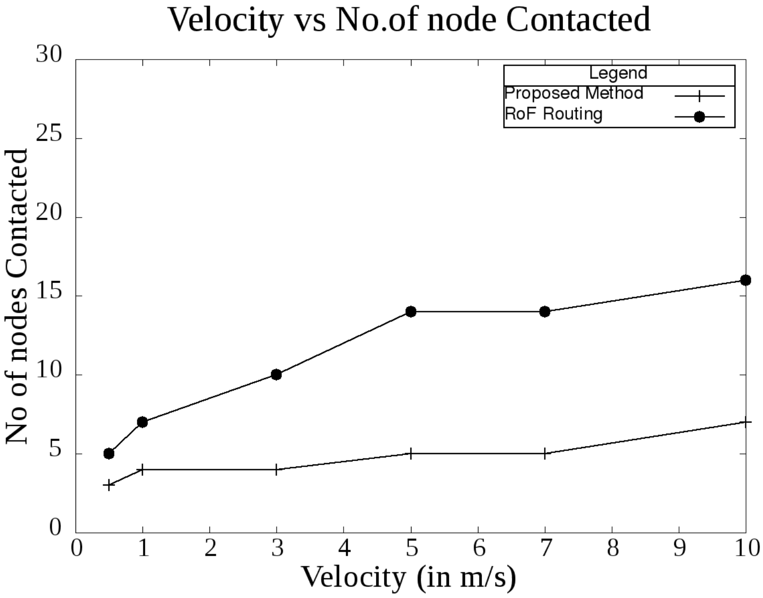

Figure 5.31: No.of nodes contacted vs Velocity

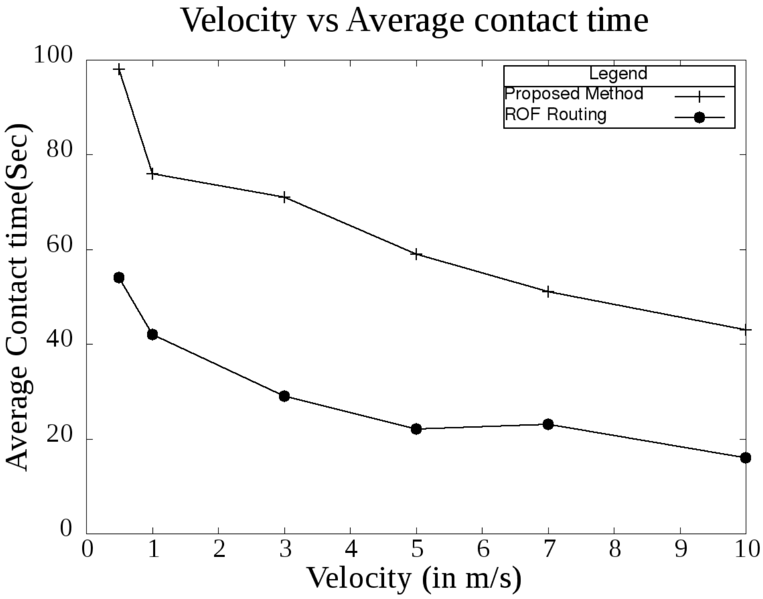

Figure 5.32: Average contact time vs Velocity

neighbor selection algorithm selects the neighbor optimally, the average contact time is highest for the proposed protocol.

5.4 Summary

In this chapter, we discuss different routing scenarios depending on the mobility of an element such as sink, source and relay nodes. For the scenario when sink is mobile, we proposed to use dynamic Steiner Minimum Tree to connect different sources to a sink. The principle of elastic routing was used to route the data from sources to mobile sink to reduce communication overhead. We extended our approach to address routing between multiple mobile sources to mobile sinks. Simulations were done to check the performance of the proposed approach by comparing against Elastic Routing and Expected Grid approach. Results indicated that the proposed approach performs better in-terms of PDR, energy consumption and control overhead. Mobility vector based neighbor selection has been proposed in Section 5.2 to select a neighbor for data communication. Simulation was done check the performance by comparing with a similar protocol MA-MAC which uses RSSI for neighbor selection. Section 5.3 proposed a routing method to transfer the data from source to sink via mobile relays. The mobility vector based neighbor selection was proposed for maximizing the data communication between the source and the destination. The proposed approach performs better when compared to ROF in-terms of PDR, Number of nodes contacted and average contact time.

Chapter 6

MAC Design for Mobile WSN

In the case of a wireless sensor network, the decision regarding when to send and listen for a packet are perhaps two most important functions that are implemented at the MAC layer. In this chapter, we investigate MAC protocols for mobile WSN and challenges in implementing them under varying mobility scenarios. Since the sensor nodes have only a single transceiver, the main challenge is to perform sending and receiving operations simultaneously or avoiding interference from adjacent nodes, which is also a function of MAC layer. Some of the features of a good MAC protocol are high throughput, low overhead, low error rate and energy-efficiency.

The criteria for designing an energy-efficient MAC protocol is similar to that of routing protocols as discussed in Chapter 5. In WSN, the energy consumed for receive operation is equivalent to the energy consumed during transmit. The energy consumed by the transceiver in idle mode is less compared to the energy consumed during data transmission. The data reception is expensive as data transmission. The transceiver in idle mode can be cheaper compared to other modes but still consumes more power than computation. Major design requirements for MAC in WSNs are the following:

- minimization of collision
- ideal sleep-wake cycle such that energy consumption is reduced without loss of packets

- minimization of control overhead

MAC protocol can be classified into, *Centralized and Distributed* MAC protocols. These can be further classified as *Schedule based and Contention based* MAC protocols. In centralized MAC protocols, the base-station or the sink controls the access over the media. While in distributed MAC, all nodes have equal access to the communication media. The centralized approach does not scale with varying network sizes and dynamic topology and thus used only in small static networks. In mobile WSN, the topology is highly dynamic, hence distributed MAC protocols are more suitable than centralized protocols.

In scheduled based MAC protocols, the media is shared between nodes according to pre-computed schedule. The sharing can be done by allotting different frequency bands to each contending node (FDMA) or can be done by assigning different time slots to each contending node (TDMA). Scheduling can either be fixed or dynamic. For a dynamic topology, the schedule generated must also be dynamic. When schedule based MAC protocols are used, time synchronization between neighbours is required: this is especially true in case of TDMA. In contention based MAC protocol, node contend for the control over the media. The MAC protocol resolves contention and gives access to one of the nodes based on some heuristics. In this chapter, two energy aware MAC protocols have been proposed for mobile WSNs.

Developing an energy-aware MAC protocol for mobile WSN poses many challenges. Some applications such as health care systems, military applications, etc., require that data collection should be done with minimum latency. Mobility aware MAC protocols available in literature can be classified as *scheduling-based* or *contention based* [125] [121]. In contention-based protocol, packet collisions are high, leading to reduction in packet delivery ratio. In scheduling based protocol such as H-MAC proposed by Srikanth et.al., , the latency is high due to the delay in selecting an appropriate neighbor and this latency increases with the number of nodes [126].

Very few available MAC protocols available that support mobility. MS-MAC [127] is a schedule-based protocol, which has been extended by modifying SMAC protocol for mobility support. MS-MAC uses RSSI values to estimate the level of mobility, assuming a one-to-one mapping between the RSSI value and the distance. MMAC protocol [128], that allows a flexible frame to accommodate a dynamic topology, is an extension of TRAMA [129]. MMAC uses node location prediction to determine the next frame format.

A TDMA-based MAC protocol proposed in [130] splits the time slot into control part and data part. Mobility management is done in the control phase, while the data transfer is done in the data phase. MC-MAC [131] is a schedule-based MAC protocol that supports group mobility, such as in body area networks or health care applications. MC-MAC uses random back-off in case of a collision that happens due to multiple clusters, moves close to each other.

6.1 Schedule-based MAC protocol

Developing an energy efficient MAC protocol for MWSN is challenging due to the dynamic topology. Most MAC protocols use low duty cycle to improve sleep-wakeup patterns with energy efficiency. Schedule based M-MAC [128] protocol employs clustering and flexible time framing to adapt to mobility. Mobility estimation in M-MAC is complex, as it depends on the mobility parameters in the previous round, that may also introduce inaccuracies in the estimated values. M-TDMA [132] organizes the network into non-overlapping clusters and some of the slots are shared between the clusters to support mobility. The new nodes joining a cluster need to wait for the next round to connect to the cluster. This results in higher latency and increased energy consumption. A cross-layer architecture as proposed in Mobisense [130], is designed for micro-mobility scenarios. Each static node in this network acts as a cluster head, for the mobile nodes in its range and the cluster head support up-link and down-link transmissions to and from the base-station. Energy Efficient Hybrid MAC proposed in [126], uses LEACH-C algorithm for clustering. It uses a hybrid method of scheduled and unscheduled channel access for accommodating topology changes. There are two phases proposed in Hybrid

MAC: *(i) Setup phase* and *(ii) Steady state phase*. The *Setup phase* involves cluster initialization and the data transfer happens during the *Steady state phase*.

In this section a TDMA based MAC protocol is proposed. It performs a dynamic scheduling of time slots to support the dynamic topology. The proposed MAC protocol employs a request-reply mechanism which not only improves the reliability of data transfer but also uses information on sleep-wake time for the next frame. The primary goal of this protocol is to improve energy efficiency, by introducing topology-control to the sensor nodes and also by reducing the number of control packets.

6.1.1 Proposed Approach

Problem Definition with Assumptions

Designing a MAC and/or routing protocol for a network with dynamic topology is a challenging task. An efficient routing algorithm in this case, requires an efficient MAC protocol. Contention based algorithms increase the network overhead and also thus probability of collision is high. Hence, a cluster based TDMA MAC protocol that can dynamically adapt to the changes in the network topology is proposed. It is assumed that the cluster head has the ability to aggregate data before forwarding it to the base-station.

6.1.1.1 Network Initialization Phase

The network initialization phase starts at node deployment. Ad-Hoc On-demand Distance Vector (AODV) routing is used to establish a route from the static nodes in the network to the base-station.

A TDMA slot is allocated for each static node starting from the base-station. None of the static nodes do start communication with their mobile neighbors before acquiring a time slot. Once the route is established, the base-station starts the slot allotment by transmitting information about available time slots to all its *children* in the routing tree. The child-node then randomly chooses a time slot different from their parent node and the process continues. This process completes when all static nodes have acquired their time slot. While the process continues, the nodes that have already acquired their time slot can commence communication with the mobile neighbors.

The communication with the mobile nodes has two stages : *Control Phase* and *Data Phase*.

6.1.1.2 Control Phase

In the control phase, the cluster head uses broadcast messages to discover new mobile nodes in its transmission range. Control phase is made-up of *(i) Discovery Request Phase (ii) Discovery Reply Phase* and *(iii) Schedule phase*.

Discovery Request Phase: At the start of the frame, the cluster-head broadcasts a discovery request packet (*disc_req*). The *disc_req* packet is an advertisement message broadcast by the cluster-head, to inform the mobile nodes in the neighborhood about the cluster. The *disc_req* is an advertisement message broadcast by the cluster head to inform the mobile nodes in the neighborhood about the cluster. This packet contains the cluster id, cluster size and remaining energy of the cluster head. A mobile node receiving this message can estimate the approximate distance between the cluster head and itself based on the RSSI value. It is possible that a mobile node may receive more than one *disc_req* message from multiple cluster head. In that case, the mobile node chooses a cluster based on the cluster size, remaining energy of the cluster head and the distance between the cluster head and mobile node.

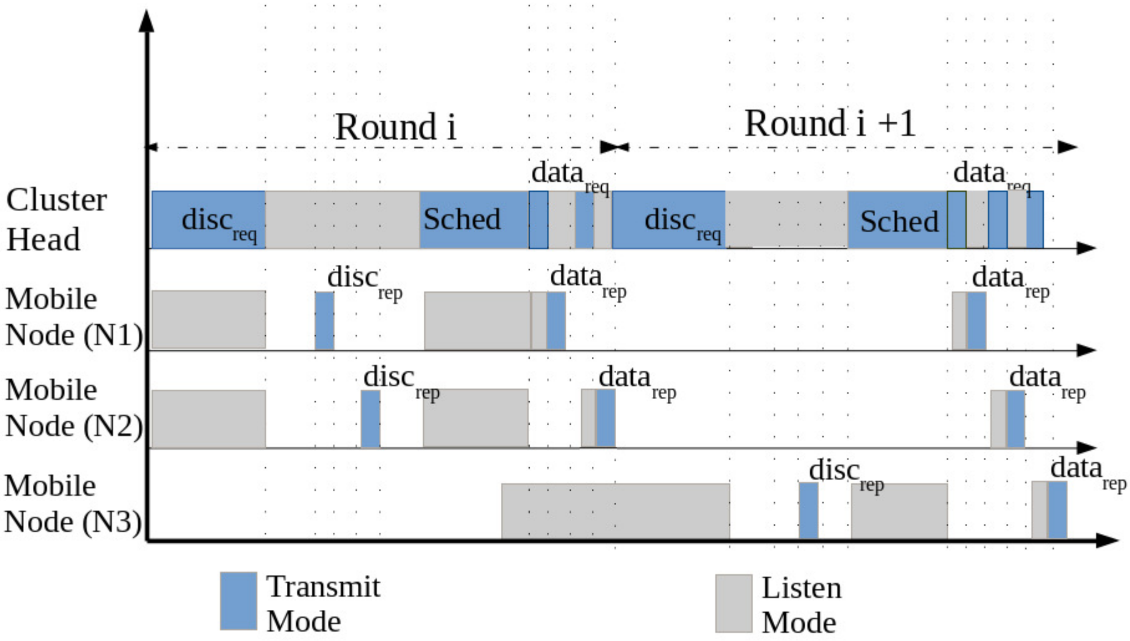

FIGURE 6.1: Time line Showing the Cluster head communication

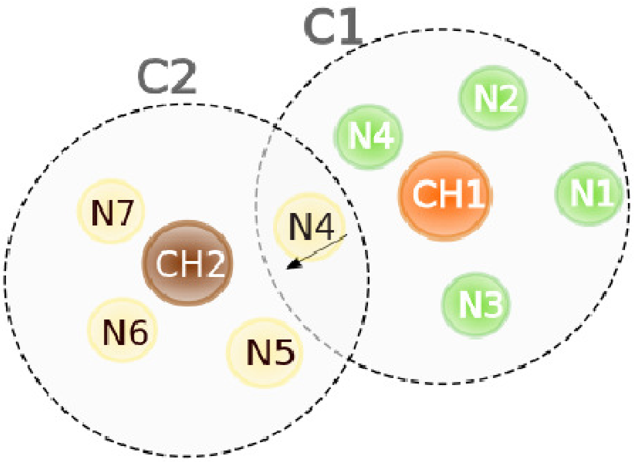

FIGURE 6.2: A sample scenario portraying inter-cluster node migration

Discovery Reply Phase: The Discovery Reply Phase consists of finite number of slots, during which the cluster head listens for Discovery Reply packets (*disc_rep*) from the mobile nodes. The mobile node, after choosing a cluster, wakes up after a random time and sends a *disc_rep* packet as shown in Figure 6.1. The maximum number of slots

(x), depends on the node density in the network.

$$x = m\pi r^2 \tag{6.1}$$

Where m is the density of mobile nodes per unit area, and r is the radio coverage radius of static nodes

Schedule Phase: Based on the *disc_rep* packets received, the cluster head creates a schedule for the mobile nodes. The order of frame access to nodes in this schedule is based on the order of reception of *disc_rep* packets. The schedule is broadcast by the cluster head.

Algorithm 6.1 From Mobile Node's Perspective

```
procedure DS_MMAC_MOBILE
    while true do
        if receive(disc_rep) then
            $CH \leftarrow disc\_rep.source$
            $disc\_slot \leftarrow rand()$
            Wake_up(disc_slot)
            $send(disc\_rep)$
            if receive(Sched) then
                for $i = 0 \rightarrow Sched.node\_count$ do
                    if Sched.node_id = ADDR then
                        $data\_slot \leftarrow Schedule[i].slot$
                        break
                    end if
                end for
                Wake_up(data_slot)
                while receive(DataReq) do
                    $data\_slot \leftarrow DataReq.slot$
                    $send(DataRep, CH)$
                    Wake_up(data_slot)
                end while
            end if
        end if
    end while
end procedure
```

6.1.1.3 Data Phase

The broadcast schedule message has a time slot assigned to each mobile node. The data phase is divided as time slot pairs (one for $data_{req}$ and another for $data_{rep}$) for each mobile node. When a mobile node wakes up during its data slot, it receives a $data_{req}$ message from the cluster head. The cluster head embeds a time value t' in this message which represents the time the mobile node needs to wake up in the future (next frame). The mobile node after receiving this message, sends a $data_{rep}$ message to the cluster head, sets a timer to wake up after t' seconds and puts the radio in sleep mode.

Consider the scenario given in Figure 6.2, where CH1 and CH2 are cluster heads of clusters C1, C2 respectively. Node N4 moves from C1 to C2. When N4 moves out of the range of CH1, it stops receiving the periodic data request message from CH1. N4 concludes that it has moved out of cluster C1, and runs the algorithm given in Algorithm 6.1). Initially, N4 switches on its radio to RX state and listens for any $disc_{req}$ packets from the cluster heads. When it receives a $disc_{req}$, it connects to the cluster head CH2 and sends a $disc_{rep}$ packet to the cluster head CH2, (lines 2 to 7 in Algorithm 6.1). The mobile node extracts its scheduled slot from *sched*. It then wakes up in its time slots and listens for a *data_req* packet from the cluster head (Lines 15,16). Lines 17 to 19 of the algorithm describes the data request and response mechanism.

The algorithm run by the cluster head is given in Algorithm 6.2. The cluster head initiates the communication by broadcasting a $disc_{req}$ message. The cluster head stores the node id of the mobile nodes from received $disc_{rep}$ message in a local buffer *boundNodes* (lines 3 to 7). In each $data_{req}$ message, the cluster head embeds the information about the slot in which the mobile node should wake-up for the next frame. Timing diagram, in Figure 6.1 shows the communication between cluster head and the mobile nodes.

6.1.2 Implementation and Performance Analysis

The performance evaluation was done with Hybrid MAC [126]. The parameters chosen for comparison are listed in Table 6.1. Simulation was done by varying the number

Algorithm 6.2 From Cluster Head's Perspective

```
procedure DS_MMAC_CH
    while true do
        if receive(disc_req) then
            broadcast(disc_req)
            while receive(disc_rep) do
                bound_nodes.add(disc_rep.source)
            end while
            for $i = 0 \rightarrow bound\_nodes.size$ do
                /* add to schedule */
            end for
            broadcast(sched)
            for $j = 0 \rightarrow bound\_nodes.size$ do
                $DataReq.slot \leftarrow next\_slot()$
                send(data_req,bound_nodes[j]
                if receive(data_rep) then
                    store data_rep
                else boundNodes.remove(j)
                end if
            end for
        end if
    end while
end procedure
```

of mobile nodes and by varying the velocity of the node. The simulation was done in Castalia.

Table 6.1: Simulation Parameters: DS-MMAC

Parameters	Values
Network Size	$100 \times 100\ m^2$
Number of Static Nodes	25
Communication radius	$30m$
Velocity	1m/s to 10m/s
Number of Mobile Nodes	1 to 100
Packet size	128byte
Simulation Time	1000Sec
Mobility Model	Random Waypoint

Energy consumption of the proposed protocol was compared with Hybrid MAC protocol. The simulations were repeated by varying the number of mobile nodes. The Figure 6.3 plots energy consumed vs number of mobile nodes. The velocity of the mobile node was set to 2m/s. It can be observed from Figure 6.3 that the power consumption in the proposed protocol is less when compared to Hybrid MAC. This is due to the efficient sleep-wake cycle.

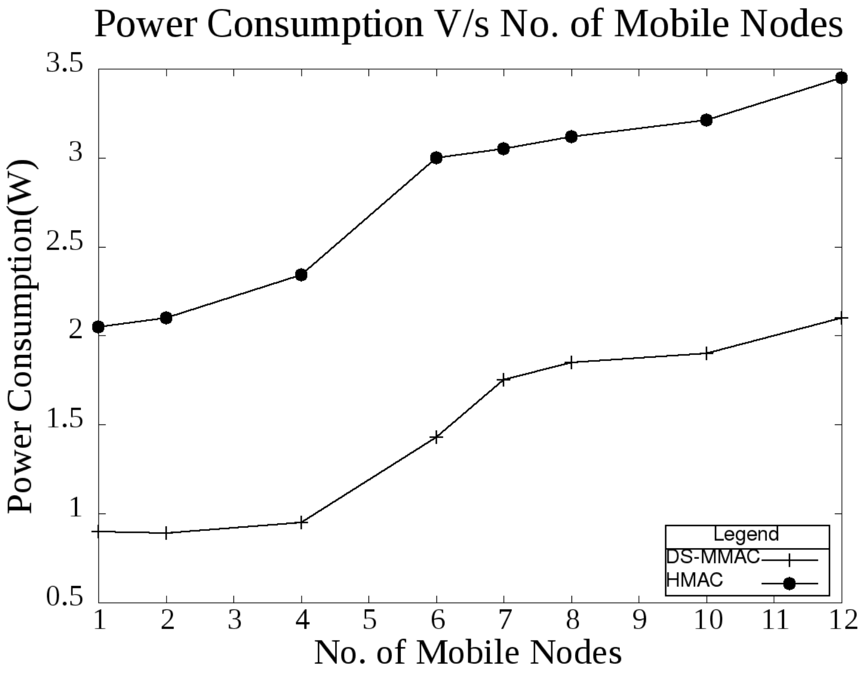

FIGURE 6.3: Power Consumption

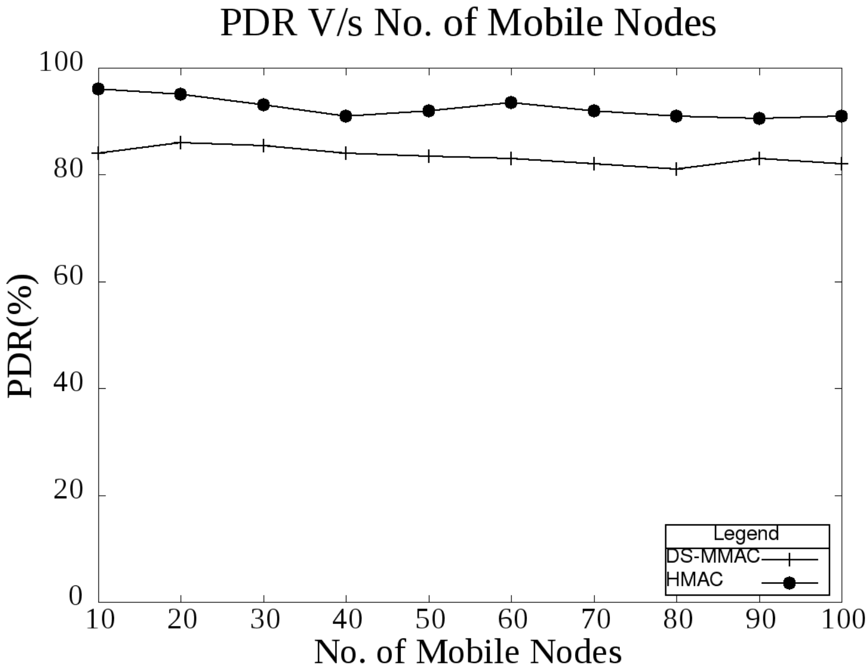

FIGURE 6.4: Packet Delivery Ratio

Figure 6.4 shows the difference in PDR of the proposed MAC protocol against the Hybrid MAC protocol. Hybrid MAC protocol performs slightly better. In the proposed

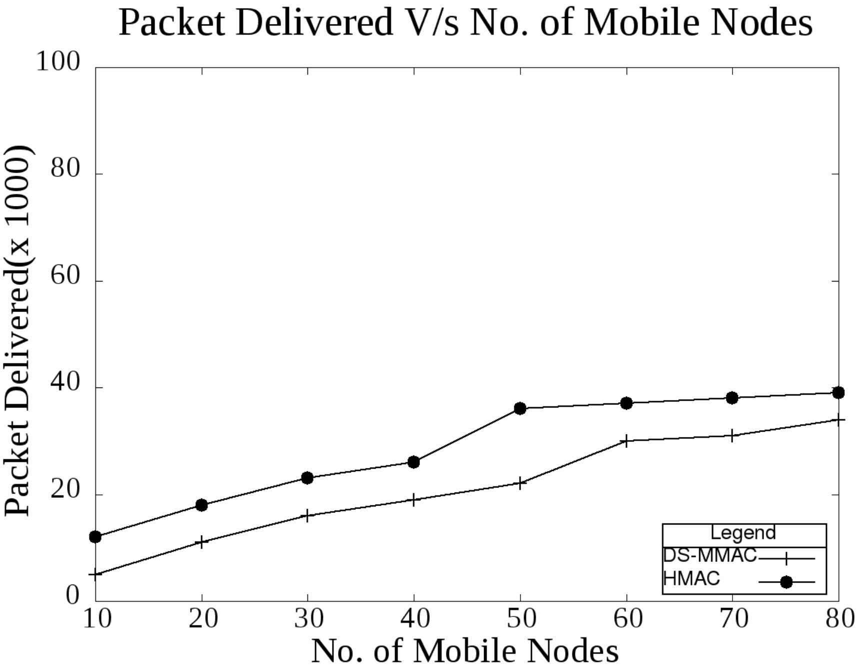

FIGURE 6.5: PDR

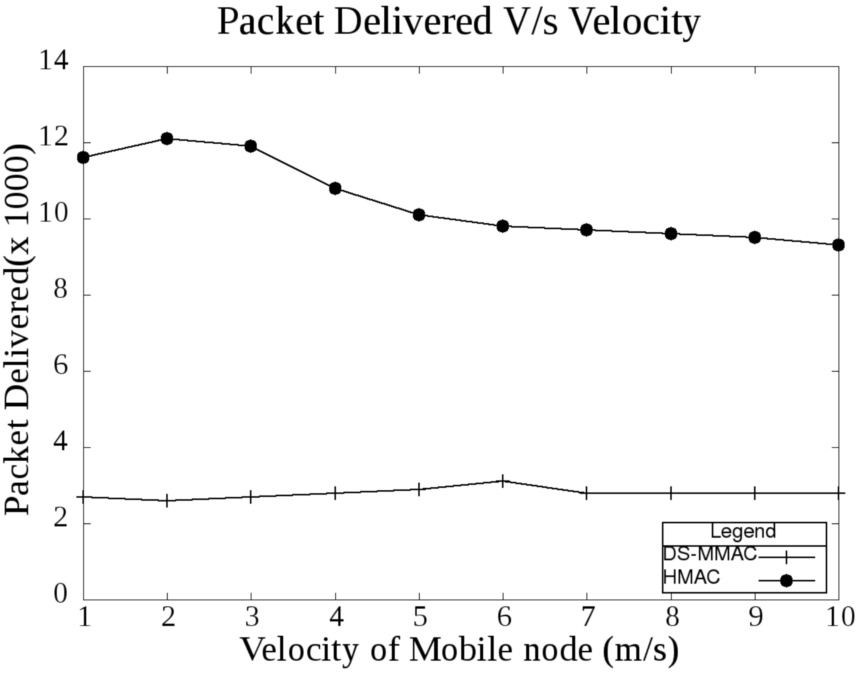

FIGURE 6.6: PDR

protocol the neighboring clusters operate on the same channel. Though there is no interference within the cluster, interference exists between adjacent clusters. This can be resolved by using multi-channel MAC.

Even though the packet delivery ratio is less, the number of packets received per second

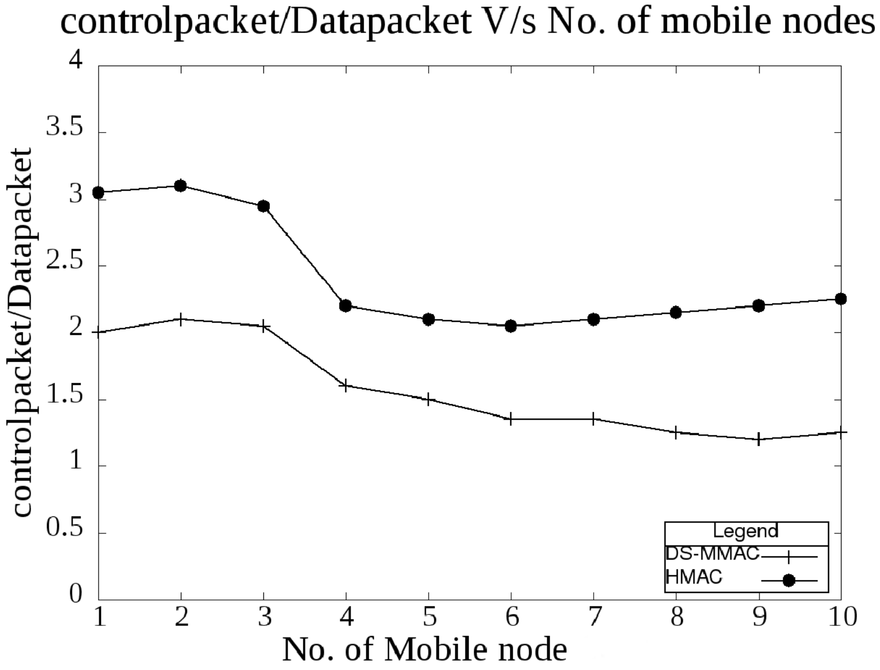

FIGURE 6.7: Control packets Vs Number of mobile nodes

(data rate) is higher as compared to DS-MMAC which can be observed in Figure 6.5. This is because the number of frames per unit time is higher. This demonstrates that the proposed protocol has a reasonable tradeoff between data rate and energy consumed. Simulations were done to study variation in PDR with respect to the speed of the mobile nodes. The graph in Figure 6.6, shows that the proposed protocol is better when compared to Hybrid MAC. Analysis of the control overhead was also done. Higher the control overhead, more is the energy expended. The is shown in Figure 6.7.

In WSNs, more energy should be spent on data transfer. If the control overhead is high, then the overall energy consumed is also high. This happens at the expense of the data packets. The ratio of the number of control packets per data packet transfer is used as an indicator of the control overhead.

Proof of concept of the proposed approach was implemented using TelosB (as mobile nodes) and one MicaZ mote (as cluster head). Mobility or the frequent change in the topology was imitated by switching the power on/off of the TelosB motes. Serial data from the MicaZ mote was read to check the performance of the proposed approach. The test-bed used for this is shown in the Figure 6.8. It has been found that the mobile node fails to communicate to the cluster head, if the mobile node is in the same cluster for a

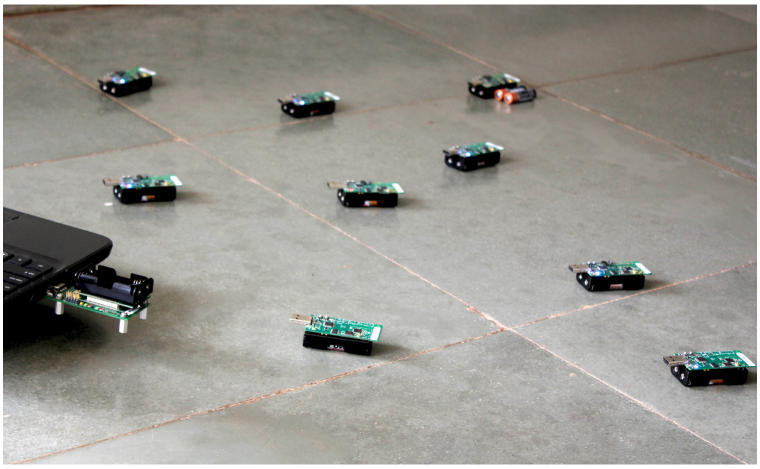

FIGURE 6.8: Testbed deployment

longer period of time. This is due to the clock skew [133]. This can be eliminated by resetting the mobile node clock with respect to the cluster head after a few thousands of rounds.

6.2 Summary

In this Chapter, we have proposed a TDMA based MAC protocol and compared our approach with Hybrid MAC. Performance of the proposed approach was compared with the packet delivery ratio, the power consumption and the number of control packets. Even though our approach offers a slightly lower packet delivery ratio compared to Hybrid MAC, our approach performs better in terms of overall packets delivered, energy efficiency and the control overhead.

Chapter 7

Conclusions and Future work

7.1 Conclusions

The present thesis covers both controlled and random mobility. Controlled mobility has been proposed to improve sensing resolution, network coverage and connectivity. Other applications of controlled mobility are removing partition in the network and use of mobile nodes as mobile sink or as data mules.

Here, we proposed algorithms that could be used to improve the network performance. The algorithms were used to enhance the performance of the existing routing and the MAC protocols. The thesis also proposed routing and MAC protocols that could be used in the case of random mobility. We have considered multiple mobility scenarios in our study.

The following mobility scenarios were considered:

- Mobile sources
- Mobile relay nodes
- Mobile sinks
- A combination of the above three

In traditional WSN, the optimal node deployment is not known until the sensed data is analysed at the base-station. Also it is impossible for static WSN to adapt to a changing environment. Even if there is a partition in the network due to node failure; the static WSNs cannot reorganize themselves. In event monitoring systems, incorrect data can lead to false alarm. The use of multimedia sensors or patrol team can handle such scenarios. Both the approaches may not be feasible in a remote location.

These issues could be overcome by utilizing the controlled mobility feature of mobile WSN. These mobile nodes can be programmed to redeploy themselves to adapt to the changes in the environment. The present thesis proposed multiple algorithms that use mobile nodes to enhance the performance of WSNs. The algorithms proposed in Chapter 3 are for event monitoring applications and for partition re-connectivity in a disjoint WSN.

Area monitoring in WSN has a multitude of applications. A broad range of applications include i) Network diagnostics ii) Residual energy scanning iii) Data collection iv) In-application programming and v) Static node localization. Even though each of these applications require a completely different level of implementation in terms of sensing or communication, their common objective is to monitor a given area with mobile nodes. Controlled mobility of mobile nodes could be utilized to perform area monitoring by optimizing on time and energy while having a minimum control overhead. In this thesis we have proposed both deterministic and non-deterministic method for area monitoring using mobile WSNs. The mobile node could also be used to enhance the performance of existing WSNs by improving QoS. A multi-path routing algorithm with frequency hopping MAC protocol for interference free data transfer is also proposed to increase the data rate.

Multiple algorithms for routing were proposed in this thesis for various mobility scenarios. Routing protocols designed in this thesis were optimized interms of energy, path length and control overhead. We addressed the following scenarios

- Routing between multiple sources and a mobile sink
- Routing between multiple sources and multiple mobile sinks

- Routing between multiple mobile sources and multiple mobile sinks

In this thesis algorithms were proposed in this thesis to address the routing problem by using the mobility vector information of mobile nodes. The proposed algorithms in this thesis were efficient in terms of speed of neighbor discovery and topology control. The thesis has proposed a schedule-based MAC algorithm for mobile WSN. The proposed method assigned different time slots for mobile node to transfer data to a static cluster head. The time slots were assigned dynamically based on the traffic condition in the mobile nodes available around the cluster. Novel features of the proposed services and protocol design can be briefly summarized as follows.

1. Mobile nodes equipped with multimedia sensors have been considered for event monitoring applications. The thesis focused on optimal placement of mobile nodes between an event and the base-station. In this work, we proposed the development of a modified Hungarian algorithm to minimise the maximum time needed for simultaneous movement of mobile nodes. The proposed algorithm was extended to address multi-event monitoring applications. A modification of an approximation algorithm for Steiner tree were used for monitoring. Multiple communication protocols have been used to handle the multichannel and normal traffic. Experiments were done to empirically determine the throughput of WSN with IEEE802.15.4. Results indicate that the MAC protocols that use multiple channels perform better than protocols that use a single channel. The protocols were tested on a test-bed that used custom designed B-Bot.

   Simulations were done using Castalia, an OMNET++ platform to study the performance of the modified Steiner tree algorithm. Results indicate that the proposed algorithm performs well even when the number of events increases. Simulations results also indicate that the proposed Steiner tree method takes less number of iterations for execution when compared to the other existing methods.

2. The objective for using mobile nodes for area monitoring was to ensure that the entire area was monitored with high resolution and low energy consumption and latency. Two area monitoring methods were proposed in this thesis:

  - Max-Gain method is a deterministic area monitoring service that uses the frontier-based algorithm. The proposed algorithm move the mobile node to an area that is currently not covered by other mobile nodes.
  - Hybrid method: A hybrid of frontier and random-direction area monitoring was used in this algorithm. The proposed method starts with random-direction movement pattern. After reaching the threshold, the algorithm switches to frontier-based method. Random-models did not converge in terms of time and area. Hence the proposed method switches to frontier-based model after reaching a threshold.

   Both the methods were simulated and compared with the frontier-based method. Results indicate that the total time required for the entire monitoring of both Hybrid and Max-Gain approach is lesser than the time taken by frontier-based method. The control overhead was also less.

3. Since sensor nodes were programmed to operate autonomously and collaborate with each other without human intervention, a node failure may cause partitions in the network. In this thesis, we suggested algorithms that can be used for partition discovery. We used area monitoring application to study the behavior of the algorithm. An optimal placement algorithm were used to place mobile node at strategic location to ensure network re-connectivity. The algorithm was implemented and tested using Berkely motes and B-bots.

4. In this thesis, we propose the use of a mobile node with multiple FDM modules. The proposed method uses a multi-path node disjoint routing protocol for data communication with FH based MAC protocol. The proposed method was compared against REAR and EECA protocol and performs better in terms of the number of paths generated, throughput and packet delivery ratio. Test-bed results indicate that the proposed method (with two paths) with frequency hopping MAC give a throughput of $137.5Kbps$, while for a single channel MAC, the throughput was $94.3Kbps$.

5. Routing in mobile sensor network was classified based on the network structure, energy efficiency, mobility, etc. In this thesis, we proposed an energy efficient

routing protocol between mobile/static sources and mobile sink(s). The proposed approach used energy efficient elastic routing with dynamic Steiner tree for establishing a minimum path connecting multiple sources to the mobile sinks. The advantage of using Steiner tree is that the number of nodes participating in routing is minimal. The dynamic Steiner tree proposed in this thesis helps to maintain a routing path for a dynamic network with mobile nodes. Simulation using Castalia was done to analyse the performance. The results indicated that the proposed method performs better in-terms of datarate, PDR and energy even when the speed of mobile nodes is increased. The algorithm was also tested on a testbed of MicaZ motes with a single source. With the increase in the number of sources, the PDR decreases (With three sources the PDR reduced to 41%) while the receiver (sink) data rate increases with more participating nodes.

6. Mobility information such as location, speed and direction could be used to optimally select a neighbor such that the link time is high. Mobility vector based neighbor selection was proposed in this thesis for maximizing the data communication between i) mobile nodes to static nodes ii) mobile nodes to mobile nodes. The simulations was done using Castalia.

   The performance was analysed by varying velocity, and by changing the sending rate. The results indicated that the proposed algorithm performs better in-terms of energy, PDR and throughput when compared to MA-MAC algorithm.

7. Finally, in Chapter 6, we proposed a schedule based MAC protocol for a network with dynamic topology. The proposed MAC protocol used request-reply mechanism to improve reliability. The primary goal of the proposed dynamic TDMA based MAC protocol was to improve energy efficiency by using topology control and by reducing the control overhead. The proposed method was implemented in Castalia by varying the velocity at different PPS. The results indicated that the dynamic TDMA based MAC protocol performs better in terms of energy, control overhead and packet delivery ratio as compared to the H-MAC protocol.

7.2 Future Work

Work undertaken in this thesis could be extended in the following directions:

One of the major issues in a mobile scenario is node localization. In this thesis, the localization method proposed in Appendix B has been used. A better localization method for mobile WSN is needed to minimize deployment failures.

Though the algorithms proposed in this thesis aim to address applications such as healthcare, military, etc., security issues have not been addressed. Security issues that include hardware, software and communication needs to be addressed. Insecure connection could lead to unwanted confidential data leak. Furthermore, an attacker can even introduce false commands/data into the network and thus affect the behavior of individual sensor nodes.

Another scope of this research is to design time synchronization algorithms. Time synchronization algorithms for mobile WSNs are still in their recent stage. Time synchronization will be required if the application needs to record the time at which events occur or if MAC protocol is TDMA based. Time synchronization could also be required for localization.

Thesis did not focus on implementing a reliable transport layer data communication between the sender and the receiver. Using complex protocols such as TCP in WSN transport layer design is not feasible. When data is sent from the sensor field to a TCP/IP network, the network needs to have a reliable transport layer. Hence there is a need to develop a light weight reliable transport layer for mobile WSN.

Appendix A

B-Bot Architecture

We made a custom designed robotic vehicle called B-Bot to verify the proposed algorithms for controlled mobility. B-Bot consist of a processing module, sensors, communication module and driver circuit. For processing, we used Raspberry-Pi 2, a single-board computer with Broadcom SoC with 900 MHz 32-bit quad-core ARM Cortex-A7 processor [134]. I/O pins of Raspberry-Pi is interfaced with driver circuit to send control signals. External sensors can be interfaced directly with Raspberry-Pi via I/O pins. For the B-Bot to move from one co-ordinate to another, it should have a notion of direction and the angle to take a turn. HMC5883L triple axis compass module from Honeywell is used as digital compass for calculating the direction. Digital compass is interfaced with Raspberry-pi through I2C protocol. Pulse Width Modulation(PWM) signals generated from Raspberry-Pi is sent to the motor driver circuit for controlling the movement of the B-Bot.

Depending on the type of application, communication modules are selected for B-Bot. IEEE802.11 or/and IEEE802.15.4 module is interfaced with Raspberry-Pi via serial port.

Figure A.1 shows the B-Bot we used for event monitoring application. This bot support two communication protocols, one for IEEE 802.11 for transferring high-bandwidth data and IEEE802.15.4 for communicating with the sensor nodes. The architecture of the B-Bot is shown in Figure A.2.

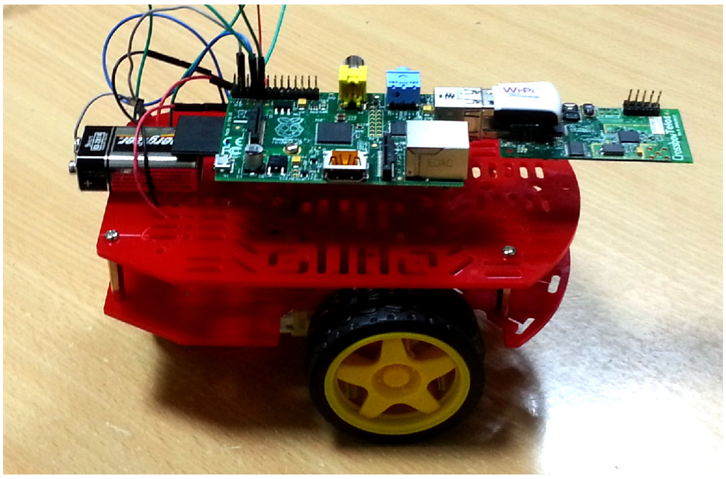

FIGURE A.1: B-Bot: Mobile node used for event-monitoring

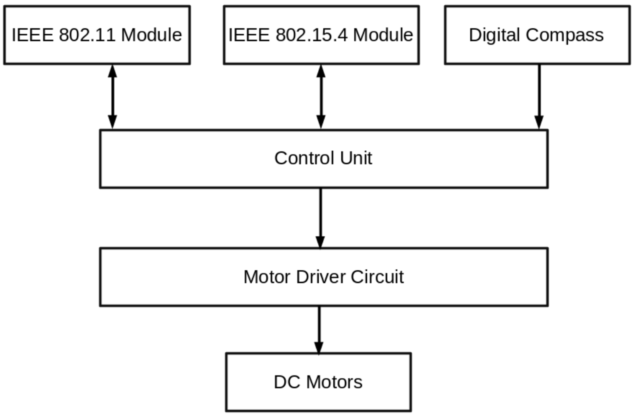

FIGURE A.2: B-Bot Architecture

Appendix B

Node Localization

The accuracy of the mobile node movement depends on the localization algorithm used. Using GPS for not localization is not economically feasible and also not suitable for indoor environment. For B-Bot localization, in-order to automate applications such as event monitoring, we used static node location to trilaterate the location of the b-bot. In our implementation, RSSI based weighted trilateration is used to calculate the current position of the B-Bot. B-Bots broadcast a position request P_{req} packet every $\Delta\hat{t}$ seconds. If a static node receives a P_{req} packet, it replies with a position reply P_{rep} message which contains the information needed for trilateration such as its position and signal strength of received P_{rep} packet that is $RSSI(P_{req})$. If a mobile node receives P_{rep} packets from its ρ neighboring static nodes, it calculates its location (x', y') as follows:

$$x' = \frac{\sum_{i=1}^{\rho} x_i * W(RSSI(P_{req})_i)}{\rho} \tag{B.1}$$

$$y' = \frac{\sum_{i=1}^{\rho} y_i * W(RSSI(P_{req})_i)}{\rho} \tag{B.2}$$

where (x_i, y_i) is the coordinate of i^{th} static node and $W(RSSI(P_{req})$ is the weight in terms of RSSI of P_{req} message of the corresponding node.

The B-Bot uses M_{loc} packet to update its location information to the base-station via a static node which is geographically closer. The selection of close static node is based on RSSI value of the received P_{rep} packet, i.e $\max\{RSSI(P_{req})\}$. In practice, the node localization using RSSI was not very accurate. When static nodes were deployed in an grid like structure separated by a distance of $2m$, RSSI based localization gives an accuracy of $\pm 1m$.

Bibliography

[1] Ian F Akyildiz and Mehmet Can Vuran. *Wireless sensor networks*, volume 4. John Wiley & Sons, 2010.

[2] Joseph M Kahn, Randy H Katz, and Kristofer SJ Pister. Next century challenges: mobile networking for "smart dust". In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 271–278. ACM, 1999.

[3] Shuang-Hua Yang. *Wireless Sensor Networks Principles, Design and Applications*. Springer-Verlag, London, 2014.

[4] Isaac Amundson and Xenofon D Koutsoukos. A survey on localization for mobile wireless sensor networks. In *Mobile Entity Localization and Tracking in GPS-less Environnments*, pages 235–254. Springer, 2009.

[5] Jane K. Hart and Kirk Martinez. Environmental sensor networks: A revolution in the earth system science? *Earth-Science Reviews*, 78(3–4):177 – 191, 2006. ISSN 0012-8252. doi: http://dx.doi.org/10.1016/j.earscirev.2006.05.001. URL `http://www.sciencedirect.com/science/article/pii/S0012825206000511`.

[6] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless sensor networks: technology, protocols, and applications*. John Wiley & Sons, 2007.

[7] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.

[8] Satyajayant Misra, Martin Reisslein, and Guoliang Xue. A survey of multimedia streaming in wireless sensor networks. *Communications Surveys & Tutorials, IEEE*, 10(4):18–39, 2008.

[9] Eren Gürses and Özgür B. Akan. Multimedia communication in wireless sensor networks. *Annales Des Télécommunications*, 60(7):872–900, Aug 2005. ISSN 1958-9395. doi: 10.1007/BF03219952. URL `https://doi.org/10.1007/BF03219952`.

[10] John Heidemann, Milica Stojanovic, and Michele Zorzi. Underwater sensor networks: applications, advances and challenges. *Phil. Trans. R. Soc. A*, 370(1958): 158–175, 2012.

[11] Ian F Akyildiz and Erich P Stuntebeck. Wireless underground sensor networks: Research challenges. *Ad Hoc Networks*, 4(6):669–686, 2006.

[12] Arun A Somasundara, Aditya Ramamoorthy, and Mani B Srivastava. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International*, pages 296–305. IEEE, 2004.

[13] Ming Ma, Yuanyuan Yang, and Miao Zhao. Tour planning for mobile data-gathering mechanisms in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 62(4):1472–1483, 2013.

[14] Ilkyu Ha, Mamurjon Djuraev, and Byoungchul Ahn. An energy-efficient data collection method for wireless multimedia sensor networks. *International Journal of Distributed Sensor Networks*, 10(9):698452, 2014.

[15] Min Chen, Chin-Feng Lai, and Honggang Wang. Mobile multimedia sensor networks: architecture and routing. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):159, Nov 2011. doi: 10.1186/1687-1499-2011-159. URL `https://doi.org/10.1186/1687-1499-2011-159`.

[16] Mario Gerla and Kaixin Xu. Multimedia streaming in large-scale sensor networks with mobile swarms. *ACM SIGMOD Record*, 32(4):72–76, 2003.

[17] Y. C. Tseng, Y. C. Wang, K. Y. Cheng, and Y. Y. Hsieh. imouse: An integrated mobile surveillance and wireless sensor system. *Computer*, 40(6):60–66, June 2007. ISSN 0018-9162. doi: 10.1109/MC.2007.211.

[18] John Tooker and Mehmet C Vuran. Mobile data harvesting in wireless underground sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*, pages 560–568. IEEE, 2012.

[19] Abdul Waheed Khan, Abdul Hanan Abdullah, Mohammad Hossein Anisi, and Javed Iqbal Bangash. A comprehensive study of data collection schemes using mobile sinks in wireless sensor networks. *Sensors*, 14(2):2510–2548, 2014.

[20] Romain Kuntz, Julien Montavont, and Thomas Noël. Improving the medium access in highly mobile wireless sensor networks. *Telecommunication Systems*, 52(4):2437–2458, 2013.

[21] Saad Ahmed Munir, Xie Dongliang, Chen Canfeng, and Jian Ma. Mobile wireless sensor networks: Architects for pervasive computing. In *Wireless Sensor Networks*. InTech, 2011.

[22] Mario Di Francesco, Sajal K Das, and Giuseppe Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(1):7, 2011.

[23] Getsy S Sara and D Sridharan. Routing in mobile wireless sensor network: A survey. *Telecommunication Systems*, 57(1):51–79, 2014.

[24] U. Baroudi. Robot-assisted maintenance of wireless sensor networks using wireless energy transfer. *IEEE Sensors Journal*, 17(14):4661–4671, July 2017. ISSN 1530-437X. doi: 10.1109/JSEN.2017.2709698.

[25] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[26] F. D. Tolba, C. Tolba, and P. Lorenz. Topology control by controlling mobility for coverage in wireless sensor networks. In *2016 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2016. doi: 10.1109/ICC.2016.7511232.

[27] Izzet F. Senturk, Kemal Akkaya, and Sabri Yilmaz. Relay placement for restoring connectivity in partitioned wireless sensor networks under limited information. *Ad Hoc Networks*, 13, Part B(0):487 – 503, 2014. ISSN 1570-8705. doi: http://dx.doi.org/10.1016/j.adhoc.2013.09.005. URL `http://www.sciencedirect.com/science/article/pii/S1570870513002084`.

[28] Zhuofan Liao, Jianxin Wang, Shigeng Zhang, Jiannong Cao, and Geyong Min. Minimizing movement for target coverage and network connectivity in mobile sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(7): 1971–1983, 2015.

[29] Stefano Basagni, Alessio Carosi, Emanuel Melachrinoudis, Chiara Petrioli, and Z Maria Wang. Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wireless Networks*, 14(6):831–858, 2008.

[30] Kemal Akkaya, Fatih Senel, Aravind Thimmapuram, and Suleyman Uludag. Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility. *IEEE Transactions on Computers*, 59(2):258–271, 2010. ISSN 0018-9340. doi: http://doi.ieeecomputersociety.org/10.1109/TC.2009.120.

[31] Gianluca Dini, Marco Pelagatti, and Ida Maria Savino. An algorithm for reconnecting wireless sensor network partitions. In *Proceedings of the 5th European Conference on Wireless Sensor Networks*, EWSN'08, pages 253–267, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 3-540-77689-3, 978-3-540-77689-5. URL `http://dl.acm.org/citation.cfm?id=1786014.1786036`.

[32] Alfred Uwitonze, Jiaqing Huang, Yuanqing Ye, and Wenqing Cheng. Connectivity restoration in wireless sensor networks via space network coding. *Sensors*, 17(4):902, 2017.

[33] Eylem Ekici, Yaoyao Gu, and Doruk Bozdag. Mobility-based communication in wireless sensor networks. *IEEE Communications Magazine*, 44(7):56–62, 2006.

[34] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002.

[35] Milica Pejanović Đurišić, Zhilbert Tafa, Goran Dimić, and Veljko Milutinović. A survey of military applications of wireless sensor networks. In *2012 Mediterranean conference on embedded computing (MECO)*, pages 196–199. IEEE, 2012.

[36] Enrico Natalizio and Valeria Loscrí. Controlled mobility in mobile sensor networks: advantages, issues and challenges. *Telecommunication Systems*, 52(4): 2411–2418, 2013. ISSN 1572-9451. doi: 10.1007/s11235-011-9561-x. URL `http://dx.doi.org/10.1007/s11235-011-9561-x`.

[37] Saad Ahmed Munir, Biao Ren, Weiwei Jiao, Bin Wang, Dongliang Xie, and Jian Ma. Mobile wireless sensor network: Architecture and enabling technologies for ubiquitous computing. In *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, volume 2, pages 113–120. IEEE, 2007.

[38] Karthik Dantu, Mohammad Rahimi, Hardik Shah, Sandeep Babel, Amit Dhariwal, and Gaurav S. Sukhatme. Robomote: Enabling mobility in sensor networks. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press. ISBN 0-7803-9202-7. URL `http://dl.acm.org/citation.cfm?id=1147685.1147751`.

[39] Dibakar Saha and Nabanita Das. Self-organized area coverage in wireless sensor networks by limited node mobility. *Innovations in Systems and Software Engineering*, pages 1–12, 2016.

[40] David Jea, Arun Somasundara, and Mani Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *international*

conference on distributed computing in sensor systems, pages 244–257. Springer, 2005.

[41] Massimo Vecchio and Roberto López-Valcarce. Improving area coverage of wireless sensor networks via controllable mobile nodes: A greedy approach. *Journal of Network and Computer Applications*, 48:1–13, 2015.

[42] Jian Li and Prasant Mohapatra. Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. *Pervasive and Mobile Computing*, 3(3):233–254, 2007.

[43] Guiling Wang, Guohong Cao, Tom La Porta, and Wensheng Zhang. Sensor relocation in mobile sensor networks. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 4, pages 2302–2312. IEEE, 2005.

[44] Lynn Choi, Jae Jung, Byong-Ha Cho, and Hyohyun Choi. M-geocast: Robust and energy-efficient geometric routing for mobile sensor networks. *Software Technologies for Embedded and Ubiquitous Systems*, pages 304–316, 2008.

[45] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ACM Sigplan Notices*, volume 37, pages 96–107. ACM, 2002.

[46] Weifa Liang, Jun Luo, and Xu Xu. Prolonging network lifetime via a controlled mobile sink in wireless sensor networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6. IEEE, 2010.

[47] Fatme El-Moukaddem, Eric Torng, and Guoliang Xing. Mobile relay configuration in data-intensive wireless sensor networks. *IEEE Transactions on Mobile computing*, 12(2):261–273, 2013.

[48] Yatish K Joshi and Mohamed Younis. Restoring connectivity in a resource constrained wsn. *Journal of Network and Computer Applications*, 66:151–165, 2016.

[49] Rahul C Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2):215–233, 2003.

[50] Kemal Akkaya, Izzet F Senturk, and Shanthi Vemulapalli. Handling large-scale node failures in mobile sensor/robot networks. *Journal of Network and Computer Applications*, 36(1):195–210, 2013.

[51] Chunping Wang, Weihong Wang, Jian Jiao, and Feihang Ge. A multi-hop and load balanced routing protocol oriented to the neighbors of the sink for wireless sensor networks. In *Global High Tech Congress on Electronics (GHTCE), 2012 IEEE*, pages 64–69. IEEE, 2012.

[52] Subir Kumar Sarkar, TG Basavaraju, and C Puttamadappa. *Ad hoc mobile wireless networks: principles, protocols and applications*. CRC Press, 2007.

[53] Samuel Pierre. *Next Generation Mobile Networks and Ubiquitous Computing*. IGI Global, 2010.

[54] Branislav Kusy, HyungJune Lee, Martin Wicke, Nikola Milosavljevic, and Leonidas Guibas. Predictive qos routing to mobile sinks in wireless sensor networks. In *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*, pages 109–120. IEEE, 2009.

[55] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless communications, IEEE*, 11(6):6–28, 2004.

[56] Lian Li, Limin Sun, Jian Ma, and Canfeng Chen. A receiver-based opportunistic forwarding protocol for mobile sensor networks. In *Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on*, pages 198–203. IEEE, 2008.

[57] Sanjit Biswas and Robert Morris. Exor: opportunistic multi-hop routing for wireless networks. In *ACM SIGCOMM Computer Communication Review*, volume 35, pages 133–144. ACM, 2005.

[58] Shahin Farahani. *ZigBee wireless networks and transceivers*. newnes, 2011.

[59] David Kohanbash, Abhinav Valada, GA Kantor, et al. Base station design and architecture for wireless sensor networks.

[60] Young-Ho Song, Changsu Suh, and Tack-Geun Kwon. Design and implementation of efficient base station for wireless sensor networks. In *2006 5th IEEE Conference on Sensors*, pages 726–729. IEEE, 2006.

[61] L. Majer, J. Mihálov, V. Stopjaková, R. Záluský, J. Brenkuš, and M. Uram. An rf network combiner towards a wireless multi-communication system for smart households. In *2014 International Conference on Applied Electronics*, pages 193–196, Sept 2014. doi: 10.1109/AE.2014.7011699.

[62] Jang-Ping Sheu, Po-Wen Cheng, and Kun-Ying Hsieh. Design and implementation of a smart mobile robot. In *WiMob'2005), IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005.*, volume 3, pages 422–429 Vol. 3, Aug 2005. doi: 10.1109/WIMOB.2005.1512933.

[63] Jang-Ping Sheu, Kun-Ying Hsieh, and Po-Wen Cheng. Design and implementation of mobile robot for nodes replacement in wireless sensor networks. *Journal of Information Science & Engineering*, 24(2), 2008.

[64] Huiyong Wang, Minglu Zhang, and Jingyang Wang. Design and implementation of an emergency search and rescue system based on mobile robot and wsn. In *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, volume 1, pages 206–209. IEEE, 2010.

[65] Nathalie Mitton and David Simplot-Ryl. *Wireless Sensor and Robot Networks: From Topology Control to Communication Aspects*. World Scientific, 2013.

[66] Fan Bai and Ahmed Helmy. A survey of mobility models.

[67] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[68] Theofanis P Lambrou and Christos G Panayiotou. Collaborative event detection using mobile and stationary nodes in sensor networks. In *Collaborative Computing: Networking, Applications and Worksharing, 2007. CollaborateCom 2007. International Conference on*, pages 106–115. IEEE, 2007.

[69] Gurkan Tuna, V Cagri Gungor, and Kayhan Gulez. An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks*, 13:54–68, 2014.

[70] A. B. Kahng and G. Robins. A new class of iterative steiner tree heuristics with good performance. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(7):893–902, Jul 1992. ISSN 0278-0070. doi: 10.1109/43.144853.

[71] Long Cheng, Chengdong Wu, Yunzhou Zhang, Hao Wu, Mengxin Li, and Carsten Maple. A survey of localization in wireless sensor network. *International Journal of Distributed Sensor Networks*, 2012, 2012.

[72] P Gireesan Namboothiri and Krishna M Sivalingam. Throughput analysis of multiple channel based wireless sensor networks. *Wireless networks*, 19(4):461–476, 2013.

[73] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[74] W Kuhn, H. Variants of the hungarian method for assignment problems. In *Naval Research Logistics Quarterly*, volume 3, pages 253–258, 1956.

[75] Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM J. Discret. Math.*, 19(1):122–134, May 2005. ISSN 0895-4801. doi: 10.1137/S0895480101393155. URL `http://dx.doi.org/10.1137/S0895480101393155`.

[76] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968. doi: 10.1137/0116001. URL `http://dx.doi.org/10.1137/0116001`.

[77] Daniel Massaguer, Chien-Liang Fok, Nalini Venkatasubramanian, Gruia-Catalin Roman, and Chenyang Lu. Exploring sensor networks using mobile agents. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 323–325. ACM, 2006.

[78] Fei Yin, Zhenhong Li, and Haifeng Wang. Energy-efficient data collection in multiple mobile gateways wsn-mcn convergence system. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 271–276, Jan 2013. doi: 10.1109/CCNC.2013.6488457.

[79] M.S. Soliman, H.M.A Fahmy, and AE. Salem. Abrm: In-network aggregation based routing protocol for mobile sensor networks with multiple mobile sinks. In *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*, pages 340–347, March 2013. doi: 10.1109/AINA.2013.39.

[80] Chia-Ho Ou. A localization scheme for wireless sensor networks using mobile anchors with directional antennas. *Sensors Journal, IEEE*, 11(7):1607–1616, 2011.

[81] R.C. Arkin and J. Diaz. Line-of-sight constrained exploration for reactive multiagent robotic teams. In *Advanced Motion Control, 2002. 7th International Workshop on*, pages 455–461, 2002. doi: 10.1109/AMC.2002.1026963.

[82] Matthew Powers, Tucker Balch, and Borg Lab. Value-based communication preservation for mobile robots. In *in 7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[83] Weihua Sheng, Qingyan Yang, Jindong Tan, and Ning Xi. Distributed multirobot coordination in area exploration. *Robotics and Autonomous Systems*, 54 (12):945 – 955, 2006. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2006.06.003. URL `http://www.sciencedirect.com/science/article/pii/S092188900600114X`.

[84] J. de Hoog, S. Cameron, and A. Visser. Role-based autonomous multi-robot exploration. In *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World:*, pages 482–487, Nov 2009. doi: 10.1109/ComputationWorld.2009.14.

[85] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 2005.

[86] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151, Jul 1997. doi: 10.1109/CIRA.1997.613851.

[87] Mario Gerla and Kaixin Xu. Integrating mobile swarms with large-scale sensor networks using satellites. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 5, pages 2816–2820. IEEE, 2004.

[88] Yujie Zhang and Sale Hui. The design of network coordinator based on zigbee and gprs technology. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, volume 2, pages 40–43, March 2012. doi: 10.1109/ICCSEE.2012.418.

[89] Yoram Koren and Johann Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1398–1404. IEEE, 1991.

[90] Johann Borenstein and Yoram Koren. Real-time obstacle avoidance for fast mobile robots. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(5):1179–1187, 1989.

[91] Ozlem Durmaz Incel, Amitabha Ghosh, Bhaskar Krishnamachari, and Krishna Chintalapudi. Fast data collection in tree-based wireless sensor networks. *IEEE Transactions on Mobile computing*, 11(1):86–99, 2012.

[92] Shuguang Xiong and Jianzhong Li. An efficient algorithm for cut vertex detection in wireless sensor networks. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, pages 368–377, June 2010. doi: 10.1109/ICDCS.2010.38.

[93] Sookyoung Lee and Mohamed Younis. Recovery from multiple simultaneous failures in wireless sensor networks using minimum steiner tree. *Journal of Parallel and Distributed Computing*, 70(5):525–536, 2010.

[94] Feng Xia. Qos challenges and opportunities in wireless sensor/actuator networks. *Sensors*, 8(2):1099–1110, 2008.

[95] Dazhi Chen and Pramod K Varshney. Qos support in wireless sensor networks: A survey. In *International conference on wireless networks*, volume 233, pages 1–7, 2004.

[96] Texas Instruments. Cc2420 datasheet, 2007.

[97] Li-minn Ang, Kah Phooi Seng, Li Wern Chew, Lee Seng Yeong, and Wai Chong Chia. Wireless multimedia sensor network technology. In *Wireless multimedia sensor networks on reconfigurable hardware*, pages 5–38. Springer, 2013.

[98] V. Viswanathan, P. Arumugam, P. Girirajan, and M. Usha. Experimental investigation of signal attenuation for wireless motes with the mechanical housing. In *2015 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–6, Jan 2015. doi: 10.1109/ICCCI.2015.7218143.

[99] Ridha Soua and Pascale Minet. A survey on multichannel assignment protocols in wireless sensor networks. In *Wireless Days (WD), 2011 IFIP*, pages 1–3. IEEE, 2011.

[100] Raja Jurdak, Kevin Klues, Brano Kusy, Christian Richter, Koen Langendoen, and Michael Brunig. Opal: A multiradio platform for high throughput wireless sensor networks. *Embedded Systems Letters, IEEE*, 3(4):121–124, 2011.

[101] Mikko Kohvakka, Tero Arpinen, Marko Hännikäinen, and Timo D Hämäläinen. High-performance multi-radio wsn platform. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 95–97. ACM, 2006.

[102] Charles E Perkins and Elizabeth M Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*, pages 90–100. IEEE, 1999.

[103] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.

[104] Surachai Chieochan, Ekram Hossain, and Jeffrey Diamond. Channel assignment schemes for infrastructure-based 802.11 wlans: A survey. *Communications Surveys & Tutorials, IEEE*, 12(1):124–136, 2010.

[105] Guillaume Fertin, Emmanuel Godard, and André Raspaud. Acyclic and¡ i¿ k¡/i¿-distance coloring of the grid. *Information Processing Letters*, 87(1):51–58, 2003.

[106] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 138–149. ACM, 2003.

[107] Athanassios Boulis et al. Castalia: A simulator for wireless sensor networks and body area networks. *National ICT Australia Ltd, Australia*, 2009.

[108] Kee-Young Shin, Junkeun Song, JinWon Kim, Misun Yu, and Pyeong Soo Mah. Rear: reliable energy aware routing protocol for wireless sensor networks. In *The 9th international conference on advanced communication technology*, volume 1, pages 525–530. IEEE, 2007.

[109] Zijian Wang, Eyuphan Bulut, and Boleslaw K Szymanski. Energy efficient collision aware multipath routing for wireless sensor networks. In *2009 IEEE International Conference on Communications*, pages 1–5. IEEE, 2009.

[110] Ben Greenstein and Philip Levis. Tinyos extension proposal (tep) 113: Serial communication, 2006.

[111] Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, and Lixia Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 148–159. ACM, 2002.

[112] Elyes Ben Hamida and Guillaume Chelius. A line-based data dissemination protocol for wireless sensor networks with mobile sink. In *Communications, 2008. ICC'08. IEEE International Conference on*, pages 2201–2205. IEEE, 2008.

[113] Hyung Seok Kim, Tarek F Abdelzaher, and Wook Hyun Kwon. Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 193–204. ACM, 2003.

[114] Zeeshan Hameed Mir and Young-Bae Ko. *A Quadtree-Based Data Dissemination Protocol for Wireless Sensor Networks with Mobile Sinks*, pages 447–458. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-45176-1. doi: 10.1007/11872153_39. URL `http://dx.doi.org/10.1007/11872153_39`.

[115] E. Lee, S. Park, S. Oh, S. H. Kim, and K. D. Nam. Real-time routing protocol based on expect grids for mobile sinks in wireless sensor networks. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5, Sept 2011. doi: 10.1109/VETECF.2011.6093204.

[116] Fucai Yu, Soochang Park, Euisin Lee, and S-H Kim. Elastic routing: a novel geographic routing for mobile sinks in wireless sensor networks. *IET communications*, 4(6):716–727, 2010.

[117] Brad Karp and Hsiang-Tsung Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.

[118] Jörg D Becker, Ignaz Eisele, and Friedhelm W Mündemann. *Parallelism, Learning, Evolution: Workshop on Evolutionary Models and Strategies, Neubiberg,*

Germany, March 10-11, 1989. Workshop on Parallel Processing: Logic, Organization, and Technology-WOPPLOT 89, Wildbad Kreuth, Germany, July 24-28, 1989. Proceedings, volume 565. Springer Science & Business Media, 1991.

[119] J.M.L.P. Caldeira, J.J.P.C. Rodrigues, and P. Lorenz. Toward ubiquitous mobility solutions for body sensor networks on healthcare. *Communications Magazine, IEEE*, 50(5):108–115, May 2012 doi=10.1109/MCOM.2012.6194390. ISSN 0163-6804. doi: 10.1109/MCOM.2012.6194390.

[120] Anne-Sophie Tonneau, Nathalie Mitton, and Julien Vandaele. How to choose an experimentation platform for wireless sensor networks? a survey on static and mobile wireless sensor network experimentation facilities. *Ad Hoc Networks*, 30: 115–127, 2015 doi:10.1016/j.adhoc.2015.03.002.

[121] Qian Dong and Waltenegus Dargie. A survey on mobility and mobility-aware mac protocols in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 15(1):88–100, 2013.

[122] N. Suri, A. Hansson, J. Nilsson, P. Lubkowski, K. Marcus, M. Hauge, K. Lee, B. Buchin, L. Mısırhoğlu, and M. Peuhkuri. A realistic military scenario and emulation environment for experimenting with tactical communications and heterogeneous networks. In *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, pages 1–8, May 2016, doi=10.1109/ICMCIS.2016.7496568. doi: 10.1109/ICMCIS.2016.7496568.

[123] Tang Zhiyong and Waltenegus Dargie. A mobility-aware medium access control protocol for wireless sensor networks. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pages 109–114. IEEE, 2010.

[124] Joachim Dzubiella, Matthias Schmidt, and Hartmut Löwen. Topological defects in nematic droplets of hard spherocylinders. *Physical Review E*, 62(4):5081, 2000.

[125] Joseph Kabara and Maria Calle. Mac protocols used by wireless sensor networks and a general method of performance evaluation. *International Journal of Distributed Sensor Networks*, 2012, 2012.

[126] B Srikanth, M Harish, and R Bhattacharjee. An energy efficient hybrid mac protocol for wsn containing mobile nodes. In *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on*, pages 1–5. IEEE, 2011.

[127] Huan Pham and Sanjay Jha. An adaptive mobility-aware mac protocol for sensor networks (ms-mac). In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 558–560. IEEE, 2004.

[128] Muneeb Ali, Tashfeen Suleman, and Zartash Afzal Uzmi. Mmac: A mobility-adaptive, collision-free mac protocol for wireless sensor networks. In *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, pages 401–407. IEEE, 2005.

[129] Venkatesh Rajendran, Katia Obraczka, and Jose Joaquin Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1):63–78, 2006.

[130] Antonio Gonga, Olaf Landsiedel, and Mikael Johansson. Mobisense: Power-efficient micro-mobility in wireless sensor networks. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pages 1–8. IEEE, 2011.

[131] Majid Nabi, Milos Blagojevic, Marc Geilen, Twan Basten, and Teun Hendriks. Mcmac: An optimized medium access control protocol for mobile clusters in wireless sensor networks. In *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9. IEEE, 2010.

[132] Arshad Jhumka and Sandeep Kulkarni. On the design of mobility-tolerant tdma-based media access control (mac) protocol for mobile sensor networks. In *International Conference on Distributed Computing and Internet Technology*, pages 42–53. Springer, 2007.

[133] Ill-Keun Rhee, Jaehan Lee, Jangsub Kim, Erchin Serpedin, and Yik-Chung Wu. Clock synchronization in wireless sensor networks: An overview. *Sensors*, 9(1): 56–85, 2009.

[134] Raspberrypi. URL: https://www.raspberrypi.org/documentation/. URL `https://www.raspberrypi.org/documentation/`.

Publications Based on Present Work

Journal Publication

1. Sreejith, V.; Prateek Khandelwal.; K.R. Anupama.; Lucy J Gudino.- ”Energy Efficient Mobile MAC protocol with Mobility Vector for Neighbour Selection”- Published in International Review on Computers and Software, Vol 12.

Conference Publication

1. Sreejith V, K. R. Anupama, Lucy J. Gudino, and R. Suriyadeepan. 2015. Partition Discovery and Connectivity Restoration in WSN using Mobile Relays. In Proceedings of the 16th International Conference on Distributed Computing and Networking (ICDCN ’15). ACM, Article 36 , 9 pages. doi: 10.1145/2684464.2684487
2. Sreejith V, K.R. Anupama, Lucy J Gudino and R Suriyadeepan - ”A Fast Area Exploration in Wireless Sensor Network using Mobile Nodes” - 7th International Conference on COMmunication Systems & NETworkS (COMSNETS) -Poster Track- Jan 7-9 2015. Bangalore, India. Jan-2015
3. Sreejith, V.; Anupama, K.R.; Gudino, L.J.; Suriyadeepan, R., ”A fast exploration technique in WSN for partition recovery using mobile nodes,” 21st National Conference in Communications (NCC) IIT-Bombay, India March-2016 doi: 10.1109/NCC.2015.7084923
4. Sreejith, V.; Anupama, K.R.; Gudino, L.J.; Suriyadeepan, R.“High bandwidth data streaming in sensor network with mobile nodes,” in 3rd IEEE International

conference on Control, Communication and Computing India 2015 (IEEE ICCC), Trivandrum, India, Nov-2015. doi: 10.1109/ICCC.2015.7432980

5. Sreejith, V.; Prateek Khandelwal.; K.R. Anupama.; Lucy J Gudino. “Mobility vector based neighbour selection in mobile WSN with multichannel MAC,” 8th International Conference on COMmunication Systems & NETworkS (COMSNETS) -Poster Track, Bangalore, India. Jan-2016
doi: 10.1109/COMSNETS.2016.7439987

6. Sreejith V, Suriyadeepan R, Anupama KR and Lucy J Gudino. ”DS-MMAC: Dynamic Schedule based MAC for Mobile Wireless Sensor Network,” 31st ACM Symposium on Applied Computing (ACM-SAC) Pisa, Italy April-2016
doi: 10.1145 /2851613.2851999

7. Meera G Sujatha, Vaibhav Gupta, Priya S Sekhar, Sreejith V, K.R. Anupama. ”An Efficient Mobile Sink Routing in Wireless Sensor Network Using Dynamic Steiner Tree”, 10th IEEE International Conference on Advanced Networks and Telecommunications Systems (IEEE ANTS) BANGALORE, India Nov-2016

Awards Based on Present Work

Awards Received

Received best PhD forum presentation award at 6th International Conference on COMmunication Systems & NETworkS (COMSNETS) held at Bangalore 2014.

Brief Biography of the Candidate

Sreejith V., is currently serving as a Lecturer in the Department of Computer Science & Information Systems, BITS Pilani K K Birla Goa Campus, Goa, India. He received his Bachelor's degree in Computer Science Engineering from Sun College of Engineering and Technology, under Manonmaniam Sundaranar University, Tamil Nadu in 2003. He did his Masters degree in Computer Science Engineering in 2007 from Rajalakshmi Engineering College, under Anna University, Tamil Nadu. He is currently pursuing Ph.D from BITS-Pilani, K K Birla Goa Campus, Goa. His research interests include Wireless Sensor Network, Internet of things and Cyber Physical Systems.

Brief Biography of the Supervisor

K. R. Anupama is Professor in the Department of EEE at BITS-Pilani, K K Birla Goa Campus, Goa, India. She received her Ph.D. from BITS-Pilani, Pilani Campus, Rajasthan, in 2005, on the topic of Routing Protocols for Mobile Ad Hoc Networks. She had research fellowship from Nokia Research Centre, Boston for a period of 4 years (2001-2005). She has developed a series of protocols using predictive source routing for unicast, multicast, geocast and inter-zone routing. She was Co-Investigator for 'Intelligent Water Resource Management Project at BITS, Pilani -Pilani Campus', funded by Microsoft Research (2005-2008) and Principal Investigator for 'Wireless Sensor Networks for Inaccessible Terrains', funded by UGC Major Research Project (2010-2012). She is currently Principal Investigator for 'Design and Development of Condition Based Monitoring of Pipelines using Wireless Sensor Networks' sponsored by GAIL India. She is currently guiding six PhD students in the area of Embedded Systems and Deeply Embedded Systems.

Brief Biography of the Co-Supervisor

Lucy J. Gudino is Assistant Professor in the Department of Computer Science and Information Systems at BITS-Pilani, K K Birla Goa campus, Goa, India. She received B.E. degree in Electronics and Communications Engineering from Kuvempu University, M.Tech. and Ph.D. in Computer Science from Vishveswaraya Technological University, Karnataka, India. She has 20 years of teaching experience in the field of Electronics and Communications and Computer Science. Her research interests are wireless communications, adaptive arrays for cellular base stations and digital filter design.