STATUS OF THESIS

Title of thesis	Clustered Data Diffusion Routing Protocol for Large-scale Wireless Sensor Networks

I SHEMSHEDIN MOHAMED ALI FARAG

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP.
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

____

The contents of the thesis will remain confidential for ___________ years.

Remarks on disclosure:

____

Endorsed by

______________________	______________________
Shemshedin Mohamed Ali Farag	Dr. Nor Hisham B Hamid
Universiti Teknologi PETRONAS	Universiti Teknologi PETRONAS
Date:	Date:

UNIVERSITI TEKNOLOGI PETRONAS

"CLUSTERED DATA DIFFUSION ROUTING PROTOCOL

FOR LARGE-SCALE WIRELESS

SENSOR NETWORKS"

by

Shemshedin Mohamed Ali Farag

The undersigned certify that they have read, and recommend to The Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree of Master of Science in Electrical and Electronics Engineering.

Signature: ______________________________

Main Supervisor: ______________________________

Date: ______________________________

Signature: ______________________________

Head of Department: ______________________________

Date: ______________________________

UNIVERSITI TEKNOLOGI PETRONAS

CLUSTERED DATA DIFFUSION ROUTING PROTOCOL FOR LARGE-SCALE WIRELESS SENSOR NETWORKS

by

Shemshedin Mohamed Ali Farag

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER OF SCIENCE

IN ELECTRICAL AND ELECTRONICS

ENGINEERING

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SRI ISKANDAR

PERAK

November, 2010

DECLARATION OF THESIS

Title of thesis	Clustered Data Diffusion Routing Protocol for Large-scale Wireless Sensor Networks

I SHEMSHEDIN MOHAMED ALI FARAG

hereby declare at the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

____________________________	____________________________
Shemshedin Mohamed Ali Farag	Dr. Nor Hisham B Hamid
Universiti Teknologi PETRONAS	Universiti Teknologi PETRONAS
Date:	Date:

To my beloved mother Ruqaya

ACKNOWLEDGEMENT

First and foremost, I would like to thank Allah the Almighty, for His blessings, and giving me health, strength and knowledge to complete this study.

My special and deepest gratitude and appreciation go to my supervisors, Dr. Nor Hisham B Hamid and AP Dr. Yap Vooi Voon, for their invaluable contributions, unlimited patience and guidance throughout this work. Without their support and encouragement, this work would not have been completed. I would like to thank Dr. Lee Sheng Chyan and Mr. Azlan B Awang for their help and support.

I would like to express my gratitude to the members of Electrical and Electronic Engineering Department and Post Graduate Office, at Universiti Teknologi PETRONAS, for their invaluable support.

My thankful goes to my friends and colleagues at UTP for their encouragement and unlimited patience. I have learnt a lot from my colleagues and friends. Special thanks go to my dearest friends Mr. Mohamed Elshaikh Elobaid and Mr. Samih Eisa Suliman for their invaluable help and support in my project.

Last but not least, I would like to thank all my family and my fiancé, for their love, encouragement, assistance and understanding throughout my study. I am very grateful to them.

ABSTRACT

One of the major challenges in the implementation of WSNs is to prolong the lifetime of the energy source in the sensor nodes. This can be achieved through designing energy-efficient routing protocol. Energy-efficient routing protocol enables WSNs to stay in operation for a long time by managing communication between the sensor nodes and the sink. In addition the routing protocol can handle a large number of sensor nodes in energy-efficient manner utilizing multihop communications among the sensors.

Four existing routing protocols performance were analyzed using J-Sim simulator. Results obtained from the performance analysis show that the Directed Diffusion (DD), creates a large amount of overhead when broadcasting a query message to the whole network. This causes huge amount of energy consumption, which reduces WSNs lifetime. Low Energy Adaptive Clustering Hierarchy (LEACH) routing protocol, which assumes one-hop communication range from the sink node is not scalable for large-scale WSNs.

To enhance the way the data can be gathered in query-driven data reporting method, a cluster-based data diffusion routing protocol for large-scale wireless sensor networks has been proposed. In the proposed method, the sink node sends the interest message unicastly, only to the cluster heads. In addition, multihop communication between the cluster heads for sending the interest, and receiving the data packets back from the source node has been used.

The results obtained have been analyzed and compared with DD and PCDD protocols as well as with other works, in which some enhancements over DD were made using different approaches. The overall results using different metrics have shown that, the proposed protocol outperforms DD and PCDD in saving the energy . The improvement of CDD in saving the energy is between 50% and 63.64% while comparing it to DD and, 8% and 42.8% compared to PCDD, for the fixed density scenario. For the fixed area scenario the improvement is up to 29.4% and at least 15% while comparing CDD to PCDD and up to 63.1% and at least 26.1% while comparing it to DD.

The proposed clustering data diffusion method also was extended to handle a combination of mobile and static nodes, and user's (sink's) mobility as well. It improved the coverage of the sensor network and for applications which need some movements of the sink node in order to gather data by sending the query message to the sensor field. A performance

analysis for the extended protocol was conducted, and a significant energy saving was achieved especially in denser networks.

ABSTRAK

Salah satu cabaran utama dalam pelaksanaan WSNS adalah untuk melanjutkan masa pakai sumber tenaga di node sensor. Hal ini dapat dicapai melalui perancangan routing protokol hemat tenaga. Tenaga-protokol routing yang cekap membolehkan WSNS tinggal di operasi untuk masa yang lama dengan menguruskan komunikasi antara node sensor dan wastafel. Selain protokol routing dapat menangani sejumlah besar node sensor dengan cara memanfaatkan tenaga-efisien multihop komunikasi antara sensor.

Empat routing protokol yang ada prestasi dianalisis dengan menggunakan simulator J-teki. Hasil yang diperolehi daripada analisis menunjukkan bahawa prestasi Difusi Pengarah (DD), mencipta sejumlah besar overhead saat penyiaran mesej query untuk keseluruhan rangkaian. Hal ini menyebabkan sejumlah besar konsumsi tenaga, yang mengurangkan WSNS seumur hidup. Tenaga rendah Hierarchy clustering Adaptif (Leach) routing protokol, yang menganggap liputan komunikasi satu-hop dari node wastafel tidak scalable untuk WSNS skala besar.

Untuk meningkatkan cara data dapat dikumpulkan dalam data-driven query kaedah pelaporan, difusi cluster data berasaskan protokol routing untuk skala besar rangkaian sensor wayarles telah dicadangkan. Dalam kaedah yang dicadangkan, simpul wastafel menghantar mesej bunga unicastly, hanya untuk cluster kepala. Selain itu, multihop komunikasi antara kepala cluster untuk menghantar bunga, dan menerima data paket kembali dari node sumber telah digunakan.

Keputusan yang diperolehi telah dianalisis dan dibandingkan dengan protokol DD dan PCDD serta dengan karya-karya lain, di mana beberapa tambahan atas DD dibuat dengan menggunakan pendekatan yang berbeza. Keputusan keseluruhan menggunakan metrik yang berbeza telah menunjukkan bahawa, protokol yang dicadangkan prestasi melebihi DD dan PCDD dalam menjimatkan tenaga. Peningkatan CDD dalam menjimatkan tenaga antara 50% dan 63,64% sedangkan membandingkannya dengan DD dan, 8% dan 42,8% dibandingkan dengan PCDD, untuk senario kepadatan tetap. Untuk senario daerah tetap peningkatan hingga 29,4% dan sekurang-kurangnya 15% sedangkan berbanding CDD untuk PCDD dan sampai 63,1% dan minimum 26,1%, sedangkan membandingkannya dengan DD.

Pengelompokan data yang dilakukan difusi dicadangkan juga telah dipanjangkan untuk

menangani gabungan node mobile dan statik, dan) mobiliti (tenggelam pengguna juga. Hal ini meningkatkan liputan rangkaian sensor dan untuk aplikasi yang memerlukan gerakan dari node tenggelam dalam rangka mengumpulkan data dengan menghantar mesej permintaan ke dalam sensor. Analisis prestasi untuk protokol diperpanjang dilakukan, dan penjimatan tenaga yang signifikan semua terutama dalam rangkaian padat.

TABLE OF CONTENTS

STATUS OF THESIS i
APPROVAL PAGE iv
TITLE PAGE v
DECLARATION OF THESIS iv
DEDICATION v
ACKNOWLEDGEMENT vi
ABSTRACT vii
ABSTRAK ix
COPYRIGHT PAGE xi
TABLE OF CONTENTS xii
LIST OF FIGURES xvi
LIST OF TABLES xviii
ABBREVIATIONS AND SYMBOLS xix
CHAPTER 1 INTRODUCTION 1
1.1 Background 1
1.1.1 Wireless Sensor Node Components 2
1.1.2 Applications of WSNs 5
1.1.3 Requirements and Challenges 6
1.1.4 Mobility in WSNs 7
1.1.5 Comparison of WSNs to Mobile Ad Hoc Networks 8
1.2 Problem Statement 8
1.3 Objectives 10
1.4 Thesis organization 11
CHAPTER 2 LITERATURE REVIEW 12
2.1 Introduction 12
2.2 Routing Methodology in WSNs 12
2.2.1 Network Structure 13
2.2.2 Protocol Operation 16

2.3 Clustering Routing Protocols for WSNs 17
2.3.1 Heuristic Schemes 18
2.3.1.1 Linked Cluster Algorithm (LCA) 18
2.3.1.2 Linked Cluster Algorithm 2 (LCA 2) 19
2.3.1.3 Highest-Connectivity Cluster Algorithm 20
2.3.1.4 MAX-MIN D-Cluster Algorithm 21
2.3.2 Weighted Schemes 23
2.3.2.1 Weighted Clustering Algorithm (WCA) 23
2.3.3 Hierarchical Schemes 24
2.3.3.1 Low Energy Adaptive Clustering Hierarchy (LEACH) 24
2.3.3.2 LEACH-C: LEACH-Centralized 26
2.3.3.3 LEACH-F: LEACH with Fixed Cluster, Rotating Cluster-Head 27
2.3.3.4 Two-Level Hierarchy LEACH (TL-LEACH) 27
2.3.3.5 Energy Efficient Clustering Scheme (EECS) 28
2.3.3.6 Hybrid Energy-Efficient Distributed Clustering (HEED) 28
2.3.4 Grid Schemes 29
2.3.4.1 Power-Efficient GAthering in Sensor Information System (PEGASIS) 29
2.4 Query Processing and Data Aggregation 29
2.4.1 Directed Diffusion 30
2.4.1.1 Naming 30
2.4.1.2 Interest Propagation 31
2.4.1.3 Gradient Establishment 33
2.4.1.4 Positive Reinforcement and Negative Reinforcement 33
2.4.1.5 Data Propagation 34
2.4.2 PCDD [28] and ECPC [29] 36
2.4.3 EPCDD [30] & ELPC [31] 36
2.4.4 DDLSC [23] 36
2.5 Performance Analysis and Comparison of Four Routing Protocols 37
2.6 Summary 39
CHAPTER 3 CLUSTER-BASED DATA DIFFUSION ROUTING PROTOCOL FOR LARGE-SCALE WIRELESS SENSOR NETWORKS 40
3.1 Introduction 40

3.2 Cluster-Based Data Diffusion Routing Protocol Overview........... 40
3.2.1 Assumptions 41
3.3 Cluster Head Election Phase 42
3.3.1 Cluster Head Election Criteria 42
3.4 Cluster Formation and Route Setup Phase 43
3.4.1 Cluster Head Advertisement..45
3.4.2 Path Set-up...45
3.5 Interest Propagation Phase 46
3.5.1 Interest in the Sink Node 46
3.5.2 Interest in the Cluster Heads..47
3.5.3 Interest in the Ordinary Nodes...48
3.6 Data Transmission Phase 50
3.6.1 From Ordinary Nodes to Cluster Heads 50
3.6.2 From Cluster Heads to Sink..50
3.7 Cluster Maintenance Phase 52
3.8 Summary.. 53
CHAPTER 4 SIMULATION RESULTS FOR STATIC NODES............................54
4.1 J-Sim Network Simulator 54
4.1.1 Characteristics of J-Sim simulator: 55
4.1.2 Components of J-Sim Simulator Framework 56
4.1.2.1 Target Node...56
4.1.2.2 Sensor Node 57
4.1.2.3 Sink Node..59
4.1.2.4 Sensor Channel 60
4.1.3 Operation of the Simulator 61
4.2 Metrics... 66
4.3 Methodology and Simulation Results 68
4.3.1 Performance Analysis with Fixed Density 68
4.3.2 Performance Analysis with Fixed Area.................................72
4.3.3 Impact of Different Number of Sources................................75
4.4 Evaluating CDD 77
4.5 *CDDM*: Query-driven Cluster-based Data Diffusion Routing Protocol for Mobile Sensor networks.. 81

4.5.1 Motivation 82
4.5.2 Cluster Head Election and Advertisement 83
4.5.3 Join-Request and TDMA Schedule 83
4.5.4 Interest and Data Messages 84
4.5.5 Cluster Maintenance 84
4.6 Simulation Environment and Simulation Results 84
4.6.1 Performance Evaluation with Fixed Density 85
4.6.2 Performance Evaluation with Fixed Area 87
4.7 Summary 90
CHAPTER 5 CONCLUSIONS AND FUTURE WORK 91
5.1 Conclusions 91
5.2 Future Work 92
REFERENCES 94
Appendix A 100

LIST OF FIGURES

Figure 1.1: Sensor Network Architecture 2

Figure 1.2: Basic sensor node hardware [7] 4

Figure 1.3: The breakdown of power consumption of a MicaZ node[11] 4

Figure 2.2: Single-hop Architecture[33] 13

Figure 2.3: Multihop flat network structure[33]...14

Figure 2.4: Single-hop Clustering Architecture[33] 15

Figure 2.5: Multihop Clustering Architecture[33]..16

Figure 2.7: Example of cluster formation (Lowest-ID) [33, 37] 20

Figure 2.8: Example of cluster formation (Highest-connectivity)[33, 37, 40]..........21

Figure 2.9: Clustering by the max – min 2 - clustering algorithm[33] 23

Figure 2.10: Flow of set-up phase [25] 26

Figure 2.11: Interest propagation for directed diffusion[49]....................................32

Figure 2.12: Initial gradients setup for directed diffusion[49] 33

Figure 2.13: Path reinforcement for directed diffusion [49] 34

Figure 2.14: Data delivery along reinforced path for directed diffusion[49].............35

Figure 3.1: Flow graph for cluster setup adopted from[25] 44

Figure 3.2: Time-line showing the operation using TDMA schedule of each node[25] 44

Figure 3.3 CDD multihop clustering...46

Figure 3.4: Cluster head receives the interest from the sink or previous cluster head 48

Figure 3.6: Cluster head detects or receives data..51

Figure 3.7: Message sequences between sink, cluster head and sensor nodes...........52

Figure 4.1: The model of a typical WSNs environment[52] 55

Figure 4.2: Internal view of a target node (dashed line)[52] 57

Figure 4.3: Internal view of sensor node (dashed line)[52]. 59

Figure 4.4: Internal view of sink node (dashed line)[52]..60

Figure 4.5: Snapshot for starting the simulation...62

Figure 4.6: Snapshot during the creation of sink node,sensor nodes and target nodes 63

Figure 4.7: Snapshot while the sensor nodes are randomly deployed and getting random location 63

Figure 4.8: Snapshot shows some messages when running the simulation 64

Figure 4.9: Snapshot showing the remaining energy while running 200 nodes-fixed area scenario-for 500 sec simulation time 65

Figure 4.10: Snapshot showing the average delay while running 200 nodes-fixed area scenario-for 500 sec simulation time 65

Figure 4.11: Snapshot showing the packet delivered while running 200 nodes-fixed area scenario-for 500 sec simulation time 66

Figure 4.13: Average Delay – fixed density scenario 71

Figure 4.14: Packet Delivery Ratio – fixed density scenario 72

Figure 4.15: Energy Efficiency – fixed area scenario 73

Figure 4.16: Average Delay – fixed area scenario 74

Figure 4.17: Packet Delivery Ratio – fixed area scenario 74

Figure 4.19: Average Delay - impact of number of sources 76

Figure 4.20: Packet Delivery Ratio - impact of number of sources 77

Figure 4.21: Energy Efficiency – fixed density scenario 86

Figure 4.22: Average Delay – fixed density scenario 86

Figure 4.23: Packet Delivery Ratio – fixed density scenario 87

Figure 4.24: Energy Efficiency – fixed area scenario 88

Figure 4.25: Average Delay – fixed area scenario 89

Figure 4.26: Packet Delivery Ratio – fixed area scenario 89

LIST OF TABLES

Table 2-1: Performance Analysis of Routing Protocols………………………….……38

Table 4-1: Features and limitations of DD related routing protocols……………….80

ABBREVIATIONS AND SYMBOLS

WSNs	Wireless Sensor Networks
ADC	Analogue to Digital Convertor
MAC	Media Access Control
CH	Cluster Head
GPS	Global Positioning System
QoS	Quality of Service
LCA	Linked Cluster Algorithm
LCA2	Linked Cluster Algorithm 2
WCA	Weighted Clustering Algorithm
LEACH	Low Energy Adaptive Clustering Hierarchy
LEACH-C	LEACH-Centralized
LEACH-F	LEACH-Fixed Cluster
TL-LEACH	Two-Level Hierarchy LEACH
TDMA	Time Division Multiple Access
EECS	Energy Efficient Clustering Scheme
HEED	Hybrid Energy-Efficient Distributed Clustering
PEGASIS	Power-Efficient GAthering in Sensor Information System
GS	Grid Seed
AC	Address-Centric
DA	Data-Centric
AODV	Ad-Hoc On-demand Distance Vector
CDD	Clustered Data Diffusion
DD	Directed Diffusion
GPSR	Greedy Perimeter Stateless Routing
MANET	Mobile Ad Hoc Networks
CSMA	Carrier Sense Multiple Access
ADV	Advertisement
J-Sim	Java Simulator

ACA	Autonomous Component Architecture
INET	Internetworking Simulation Platform
Tcl	Tool Command Language
Ns-2	Network Simulator
SSFNET	Scalable Simulation Framework Network Models
ARP	Address Resolution Protocol
PCDD	Directed Diffusion with Passive Clustering
DDLSC	Directed Diffusion based on Link-Stabilizing Clustering
ECPC	Energy Conserving Passive Clustering
EPCDD	Enhanced Passive Clustering Directed Diffusion
ELPC	Energy Level-based Passive Clustering
CDDM	CDD-Mobile

CHAPTER 1

INTRODUCTION

1.1 Background

Wireless Sensor Networks (WSNs) are networks of tiny devices equipped with sensors, microprocessors and wireless communication interfaces. They consist of several hundreds of sensor nodes that interact with surrounding environment by sensing or monitoring physical parameters. Figure 1.1 shows the architecture WSNs which consists of sensor nodes and a sink node. These sensor nodes communicate using wireless technologies which include RF, Bluetooth, UWB or infrared[1]. Data gathering, aggregation, and communication from remote environment using collaborated and densely deployed sensor nodes can be achieved in which the sensor nodes are connected through radio link[1, 2]. Wireless sensor nodes are a type of devices in which the network is formed in infrastructure-less manner and they are unattended by human being. WSNs are used for various applications, ranging from personal health care to environmental monitoring and military and security application[3].

Wireless sensor nodes are deployed over a geographical area to collect data based on the interest of the user and forward it back to the base station. Typically, these sensor nodes are battery-powered and should be operable unattended for several years, based on the applications. Thus, sensor nodes must be and able to be energy-efficient self-organizing[4].

Depending on the application of the sensor nodes, the way the sensor nodes use to report data from the environment can be divided into three methods: event-driven, time-driven and query-driven data reporting methods. In event-driven reporting method, the sensor node detects new events and it reports to the base station. In time-driven scenario, the sensor nodes send data periodically to the base station. However, the data reporting to the base station will be achieved based on the user's request in query-driven data reporting approach [5, 6]. The proposed cluster-based data diffusion routing protocol belongs to query-driven data reporting method, in which the user (sink) disseminates the interest message to get answer from the collected information, in particular, geographical area in the sensor field.

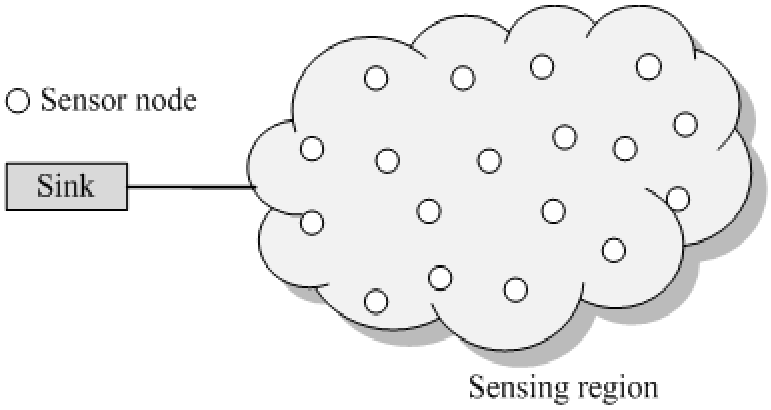


Figure 1.1: Sensor Network Architecture

1.1.1 Wireless Sensor Node Components

Most of the sensor nodes are composed of the following four basic sub-systems as illustrated in Figure 1.2 [7-10]:

- **Communication:**

The communication sub-system consists of a bidirectional wireless communication channel which helps the sensor nodes to communicate with each other and with the base station. Communication is established through short radio range; Bluetooth, laser, ultrasound, inductive fields and infrared. There are some drawbacks of using ultrasound and inductive field, for the network coordinator requires high energy and the low range of the inductive field[7]. Even though an infrared can be implemented easily, but it needs for a passable line-of-sight communication between the sensor devices. As radio frequency (RF) is not limited by line of sight and the recent technologies allows implementation of low-energy radio transceivers associated with more scalable ranges and data-rates, it is the most suitable communication for WSNs.

Communication is the most energy-consuming part of the sensor node [9]. Therefore, using different approaches to the save the energy through different communication strategies, i.e. multihop between the sensor nodes, is very important.

- **Power supply:**

This sub-system consists of a battery, which supplies the rest of the sensor node components, and a dc-dc converter. The usage of the energy for all components of sensor nodes must be in energy-efficient manner, as most of the time the battery of the sensor nodes is irreplaceable due to completely inaccessible environments in which the sensors are deployed, or thousands of nodes widely scattered over a large area [7].

- **Processing unit:**

It is composed of an embedded operating system, memory to store data and applications programs, a processor to process the data and an Analog-to-Digital Convertor which receives the signals from the sensing unit. It is responsible to organize the data obtained from the sensing unit and managing answers for some specific user or sink queries. The embedded processor of the sensor nodes are different based on the applications offered by WSNs; for example a video sensor processor is more robust than which designed to sense the temperatures. Moreover, the processors can be in one of the operational modes; Active, Idle and Sleep[7, 9].

- **Sensing unit:**

The sensing sub-system links the sensor node to the physical world. Depending on the applications, the sensing unit has different sensors which are built in, for example temperature, humidity, light and so on. Also, the smart sensor may include chemical and biochemical sensors to detect toxic or explosive trace in different public areas[7, 9].

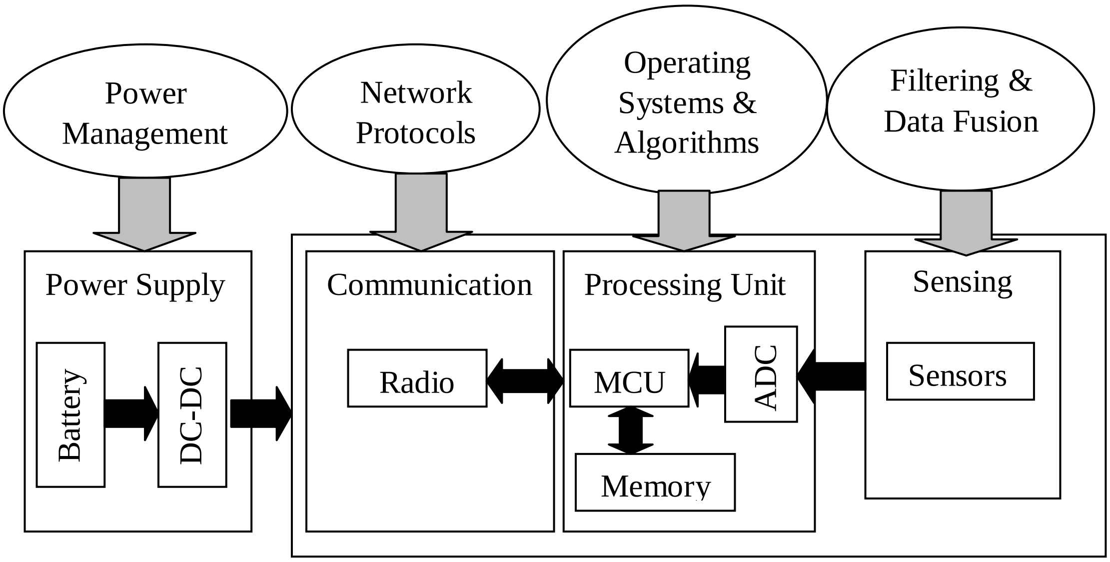


Figure 1.2: Basic sensor node hardware [7]

After the deployment of the sensor have been completed in the sensor field, the major task of the sensors is detecting the event, performing data processing locally, and forwording the detected data to to the base station. Then each node's power consumption can be devided in three different parts: sensing which is utlized by the sensors, communication which is used by the CPU and data processing which is used by the radio. The MicaZ sensor node's power consumption is shown in Figure 1.3 in which it can be obviosly seen that the sensor node depletes maximum energy for data communication among the three parts [11].

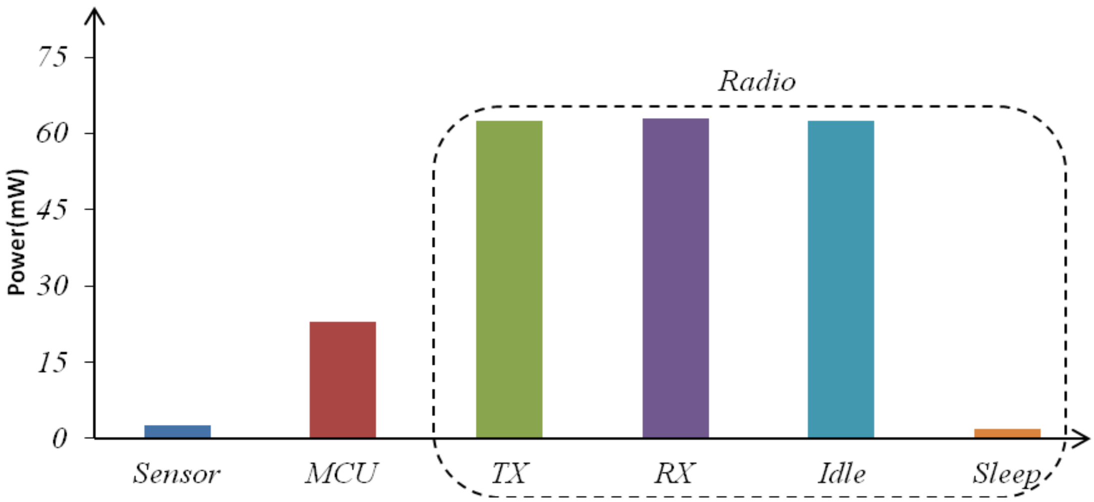


Figure 1.3: The breakdown of power consumption of a MicaZ node[11]

1.1.2 Applications of WSNs

In order to understand the significance of WSNs, this section briefly describes the applications which include the areas where WSNs are or can be distributed[1, 4, 12-14]:

- ***Health monitoring*****:** there are many ways in which WSNs can be exploited in health-care services. For example, it can be used to monitor the patients, health diagnostics, drug administration in hospitals, tele-monitoring of human physiological data, etc. The patient's signs also can be tracked using wearable sensors (blood pressure or heart monitoring) which can be stored using different devices (PDA or homePC) or it can be forwarded directly to doctors.
- ***Environmental monitoring:*** WSNs can be used in some environmental applications like fire detection, tracking of movements of birds, animals or insects, detection of chemical and biological agents. As an example, a huge number of tiny temperature sensor nodes can be distributed from an aircraft over a remote forest. As soon as fire starts, the sensor nodes can forward the location of the fire to the base station before the fire spreads to a larger area of the land. Also, the sensor nodes can complement the remote sensing from satellite as they can observe the environment at higher spatio-temporal resolution.
- ***Military and security:*** the applications of WSNs in military are varied in which the sensors can be attached to soldiers, vehicles or equipments to gather information about their situation and position, which will support to plan their activities in the battle field. In other ways also, the sensors can help by gathering information like intensity, radiation level and type of chemical agents if biological attacks are used. In addition, by using seismic, acoustic or video sensors reconnaissance of enemy terrain or forces can be discovered. Besides, the sensor nodes also can be used as a remote border security to monitor any possible intrusion, which contributes to reducing manpower cost and enhances the security level.
- ***Industrial safety:*** the sensor nodes also can be used to monitor buildings, bridges and highways. In the same way, the sensors can be used in factories to monitor machines together with air pollution and fire monitoring.

- ***Other applications:*** there are many different possible applications in which the sensors can be used, for example, home automation, environmental control in office spaces, detecting car thefts, vehicle tracking and smart environment. Also, there are a huge number of research paths ranging from hardware to designing energy-efficient routing and MAC layer protocols.

1.1.3 Requirements and Challenges

As mentioned in section 1.1.1 routing protocol is crucial for WSNs implementation. A good communication protocol is achieved through proper identification for the parameters required for the routing protocol to fulfill its functionality[15-18].

- **Scalability:** a large number of nodes may be used for different applications of WSNs. Based on the applications; a large scale of sensor nodes can be distributed to collect the data required. As the number of nodes increases, the network the scalability issue emerges. To handle a large number of sensor nodes in the sensor field, hierarchical network structure has been proven to provide better scalability than the flat ones[15, 17]. In hierarchical network topology, clustering is utilized to divide the sensor nodes into clusters, processing the data within the clusters, in which the resource efficiency can be improved and the longevity of the sensors can be guaranteed. A routing protocol should support huge number of nodes in the network by providing the services which are needed from the sensor nodes.
- **Ease of deployment:** based on the applications, the sensor networks may contain a large number of nodes, hundreds or thousands, and they may be distributed in remote or dangerous environments utilizing a plane flying, in which it may be impossible otherwise. So the sensor nodes should be small and inexpensive to throw them over a remote area to extract information from the environment.
- **System lifetime/Low energy use:** as the sensors may be deployed in remote areas, replacing the battery of the sensors are almost impossible. Therefore the lifetime of the sensor nodes are based on battery life. WSNs nodes should fulfill the task given and function as long as possible; which different from nodes to others depending on the applications, which are required from WSNs

to achieve it. Energy-efficient routing protocols are needed to guarantee the longevity of WSNs lifetime.

- **Latency:** in some applications the user needs to get the data as quickly as possible and, long delays due to data aggregation processing or communications between the sensor nodes are sometimes unacceptable. In addition, some real-time applications require an immediate response from the sensor field. Therefore, the routing protocol should be aware of latency to deliver the information to the user within a short time.
- **Self-configuration:** especially in remote environment, since the sensor nodes are unattended, manual configuration is not feasible. Therefore, the sensors should be able to self configured to some changes, i.e. that might happen in the network, like node failure and node addition.
- **Querying ability:** a user may be interested to send a query to an individual or a group of nodes in a particular region to get information. Depending on the amount of data fusion performed, it may not be feasible to send a large amount of data across the network. Instead, local cluster heads can be used to collect and aggregate the data from a given area, and finally forward them to the user through multihop cluster heads.

1.1.4 Mobility in WSNs

Based on the support offered by wireless communication to mobile devices, mobility in WSNs can appear in three forms[18, 19]:

- **Target (event) Mobility:** this is used in application in which more than one sensor node will participate to detect the movements or mobility of an event particularly in tracking applications. As many sensors are involved in detecting the events movements, they must wake up during the observation of an event, and go back to sleep mode later.
- **Node Mobility:** based on the applications, the sensor nodes themselves can be mobile too. For example in livestock surveillance, the sensor nodes attached to cattle can be mobile. But on the other hand, the sensor nodes are frequently challenged to self-configure to the changes in position, speed and energy consumption while maintaining the desired function in the network

- **Sink Mobility:** the special case of node mobility can be when the sink node moves and disseminates the interest message, and gathers the data from the sensor field. The mobile sink includes external devices like PDAs, laptops or gateways when the user requests for the information from the sensor network using these devices. The user can interact with any node or only with specific nodes in a particular region of the network, and request for required information using the devices.

1.1.5 Comparison of WSNs to Mobile Ad Hoc Networks

A new design consideration should be introduced for an emerging infrastructure-less wireless communications; Mobile Ad Hoc Networks (MANETs) and WSNs[20]. The routing between the sensor nodes in WSNs is more complicated in contrast to the traditional and infrastructure-based wireless communications in which the fixed base station manages the communication between the mobile nodes. In addition, WSN has a unique characteristics and constraints when compared it to MANETs.

There are some similarities between MANETs and WSNs. For example, in both MANETs and WSNs the communications among the nodes are unreliable as the networks are infrastructure-less; there is no base station manages the communication between the nodes. Additionally, the nodes are self-configured and unattended by human-being, and they support multihop communications between the nodes. On the other hand, WSNs has some differences when compared it to MANETs. WSNs is very constrained in energy, most of the time, based on the applications, sensor nodes are used in severe environments and once deployed, it is very difficult to replace their battery. Additionally, in WSNs the nodes are distributed in large numbers and densely deployed and data-centric routing method is used in which the identity of the sensor which sends the data is not important. Thus, sensor nodes collaborate with each other to achieve one goal; sometimes tens of nodes will participate to answer user's question[15, 18, 20, 21].

1.2 Problem Statement

One of the major challenges in the implementation of WSNs is to prolong the lifetime of the energy source in the sensor nodes. This can be achieved through designing efficient power supply, using solar cells to scavenge the energy and designing energy-efficient routing protocol. Energy-efficient routing protocol enables WSNs to stay in operation for a long time by managing communication between the sensor nodes and the sink. In addition the routing protocol can handle a large number of sensor nodes in energy-efficient manner utilizing multihop communications among the sensors[22]. The most common routing protocol architectures are Flat, Hierarchical, and Location-based architecture. Flat architecture creates a significant overhead which degrades network's performance and the lifetime of the sensor nodes [23]. On the other hand, the Hierarchical architecture provides a better solution for the aforementioned energy issue.

Four existing WSNs routing protocols, flat routing protocol, DD [24], hierarchical routing LEACH[25], Location-based Greedy Perimeter Stateless Routing (GPSR) [26] and on-demand Ad-Hoc On-Demand Distance Vector (AODV)[27], are typically implemented. LEACH, which is clustering technique, is more energy efficient than the others. However, it assumes single-hop communication from the sink node which is not scalable for large scale sensor networks. In addition, LEACH also assumes that each node has data to send to the base station according to the set time schedule from each cluster heads to its member, which is not suitable for applications in which the sink node interested in information from some specific region, for example, query-driven techniques. DD is the most well-known recognized, robust and scalable and, based on query-driven data delivery technique WSNs routing protocol. In WSNs applications the query-based routing protocols are used in the areas, in which the user sends a query to specific region of the sensors to get the answer. However, in the original DD[24], the authors declared that the MAC layer which has been used in their work is not completely satisfactory, since a Time Division Multiple Access (TDMA)-style Media Access Control (MAC) is more efficient and is necessary for long-lived sensor nodes. Thus, the design of routing protocols can be associated with a TDMA-style MAC address layer to reduce idle-energy consumption.

Thus, designing an energy-efficient, scalable and multihop communication supportive routing protocols which associated with a TDMA scheduling MAC are needed to save the energy, increase the lifetime and, to handle a huge number of nodes in the network.

Many studies have shown the improvements for the performance of DD routing protocol[23, 28-31] using different approaches in order to reduce the energy consumption, and guarantee the longevity of WSNs. However, they utilize the same, 802.11 MAC layer which have been used in original DD and some of them assume that the energy consumption for *idle* state to be zero, despite it is stated in DD that, an 802.11 radio's idle power consumption is close to the transmission one. Moreover, the protocols which have deigned to improve DD's do not support any mobility of the sensor nodes. In spite of mobile sensor nodes can be exploited to detect the sources which might not be covered by the stationary nodes. In addition, the mobile sensor nodes can be utilized to compensate for the lack of sensor nodes and to enhance the coverage of the network as well [32].

1.3 Objectives

The main objective is to design and implement energy-efficient routing protocol. The work will focus on designing and implementing a query-driven, cluster-based data diffusion routing protocol associated with a TDMA-style MAC layer, to prolong the lifetime of sensor nodes by saving energy. In addition, implementing and creating a multihop communications between cluster heads to handle a large number of sensor nodes. Another aspect is the use of mobile nodes to enhance the lack of stationary nodes and to compensate the coverage.

The methodologies have been followed to achieve the objectives of this thesis can be summarized as:

- Comparison of four well-known existing WSNs routing protocols by contrasting their features
- Proposal of query-driven, cluster-based data diffusion routing protocol associated with a TDMA-style MAC
- Extension of a query-driven, cluster-based data diffusion routing protocol associated with a TDMA-style MAC for mobile sensors
- Validation of the results through simulations

1.4 Thesis organization

The thesis consists of six chapters. The first chapter gives the introduction and background of WSNs. Also, the problem statement, the methodology that has been used and the objectives are described in details in the same chapter.

Chapter 2 reviews the works that have been done so far in the area. Details of the routing protocols that have been proposed for WSNs and the main features of the clustering protocols are also discussed.

Chapter 3 presents the methodology of the proposed method. This chapter covers the overview of CDD, cluster head electing phase, cluster formation and path set-up phase, interest packet propagation phase, data messages transmission phase and cluster maintenance phase in details.

Chapter 4 discusses the simulator that has been used for this work. In this chapter the main components of the WSNs: sink node, sensor node and target node are presented. In addition, the performances and results of the static wireless sensor nodes for the proposed method are compared with the original DD.

Chapter 5 continues with the discussion of the simulation based study by using different metrics, and analyzes the results obtained to show the performance of the proposed method after consideration of sink's (user's) mobility and the combination of mobile and static sensor nodes.

Chapter 6 concludes and recommends some future work that can be extended in the area.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

WSNs routing protocols functionalities differ based on the application and network architectures[5]. The challenge of WSNs is designing an energy-efficient routing protocol to prolong WSNs lifetime in terms of energy and data efficiency. Thus, saving sensor nodes' energy is desired while the sensor achieving its functionality.

2.2 Routing Methodology in WSNs

In WSNs, the routing method has to be simple which does not spend much memory and computational power. Moreover, it should minimize the communications among the sensor nodes to reduce the energy consumption. Typically a sensor network consists of tens, hundreds and sometimes thousands of nodes which are densely deployed in the interest area. One or more sink(s) will be provided from which the

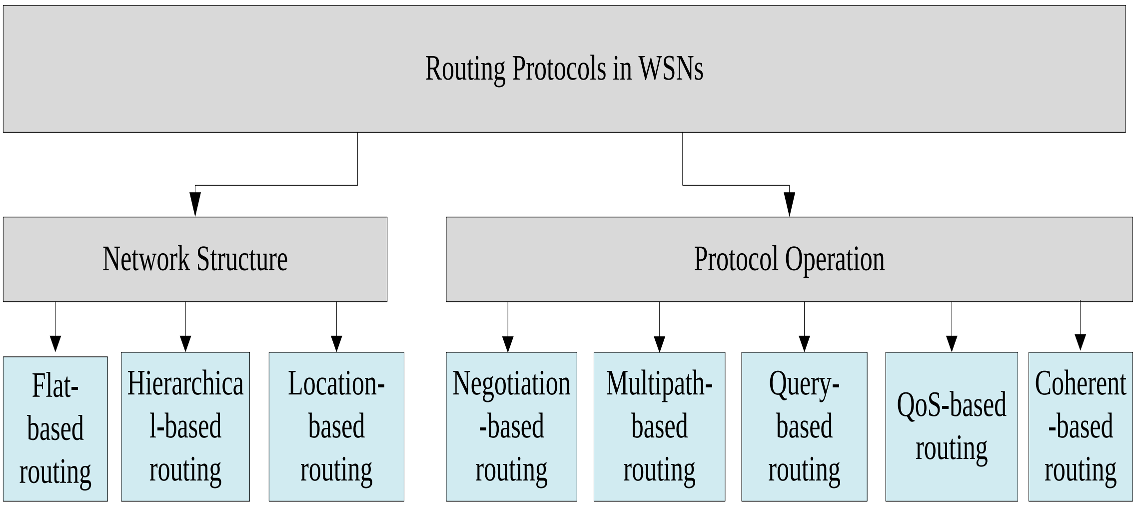


Figure 2.1 Routing protocols in WSNs: a taxonomy[5]

user can get the information through it. Whenever the sensor node detects the requirement data, it can send the data to the sink node via single-hop or multihop transmission between the source node and the base station. In general, wireless sensor routing protocol can be classified as flat-based, hierarchical-based and location-based routing according to network structure. In addition, they can be categorized based on protocol operation into negotiation-based, multipath-based, query-based, QoS-based and coherent-based routing. Figure 2.1[5] shows the taxonomy of WSNs routing protocol based on the two architectures mentioned above. According to[5] multihop communication can be categorized into two parts, flat-based and hierarchical-based network architectures.

2.2.1 Network Structure

Depending on network structure, WSNs routing protocols classified as a flat-based routing, hierarchical-based routing and location-based routing.

- **Flat-based Routing**

In the sensor nodes environment, a flat of single-hop long-distance communication between the sink node and the sensor nodes can be used as shown in Figure 2.2.

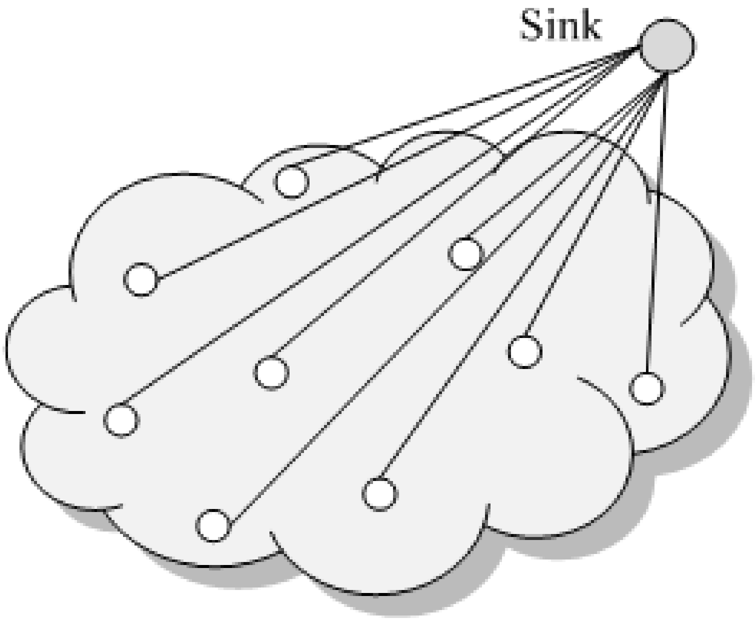


Figure 2.2: Single-Hop Architecture[33]

However, it is costly because the energy used to transmit data over a long distance will increase and, leads to high energy consumption among the sensor nodes as the energy utilized to forward the packets is much higher than that used for computation and sensing. This, consequently leads to a reduction of the WSNs lifetime [33].

Additionally, in flat-based routing architecture, the distributed sensor nodes in the sensor field play the same function and they are equal. Generally, data gathering in wireless sensors is fulfilled in data-centric manner; described in more details in section 2.4, in which the sink node broadcasts the query in the sensor field and gets the answer from some sensing region. In this case, the intermediate nodes between the source and the sink nodes are used as routers to deliver the data to the sink. The flat structure is not scalable for large networks in which the nodes number grows and it is more efficient for small networks[5, 33]. Figure 2.3 shows the multihop flat-based network structure.

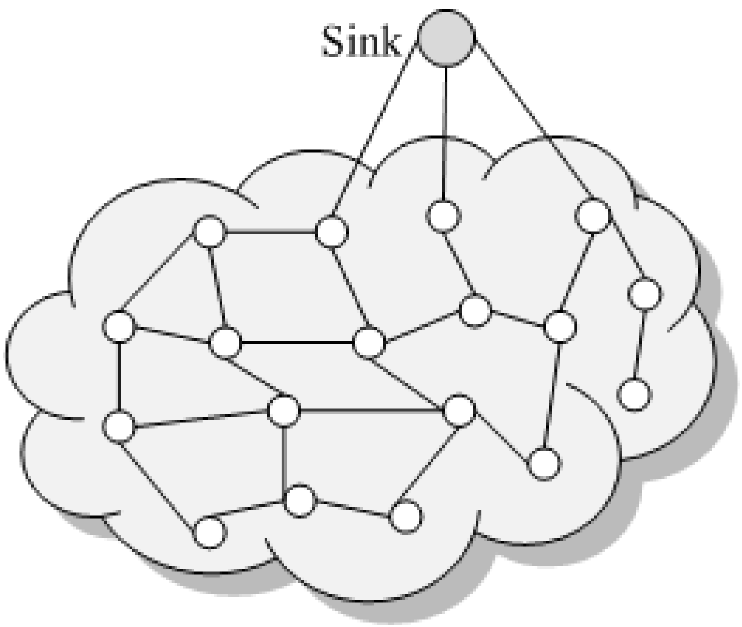


Figure 2.3: Multihop Flat Network Structure[33]

- **Hierarchical-based Routing**

In a hierarchical, also known as a cluster-based, routing approach, the sensor nodes are grouped together to create a cluster. One of the nodes acts as the leader of the cluster. If the sensor node detects the event, it will sends the data to corresponding cluster head and the cluster head will perform data aggregation before the data are forwarded to the sink node through multihop cluster heads or the single hop ones. The hierarchical methods have a great improvement over the flat-based approaches, for wireless sensor nodes; reducing the energy consumption and prolonging WSNs lifetime as well. In addition, hierarchical architectures are more scalable for large networks than the flat one as they can be utilized to cover a larger area of sensor nodes[5, 33]. Figure 2.4 illustrates single-hop clustering while Figure 2.5 shows multihop clustering structure.

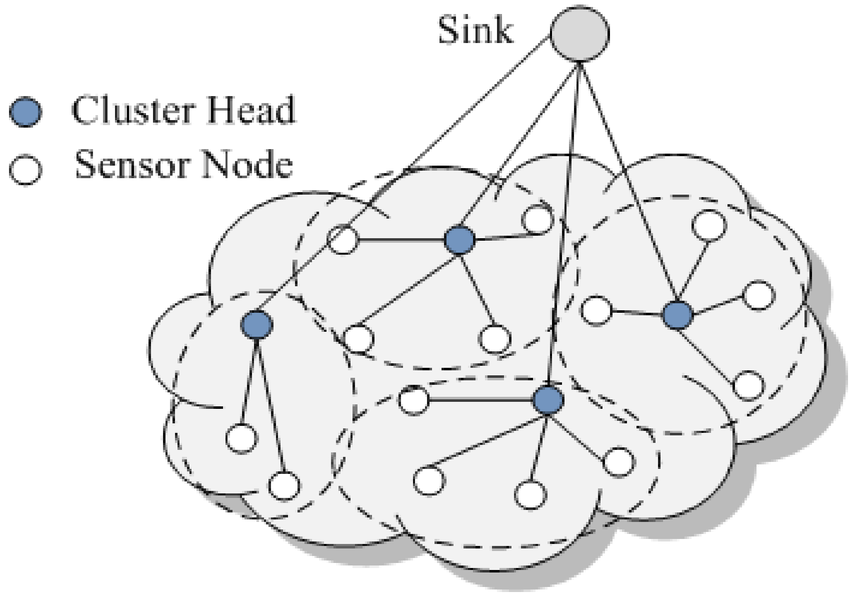


Figure 2.4: Single-Hop Clustering Architecture[33]

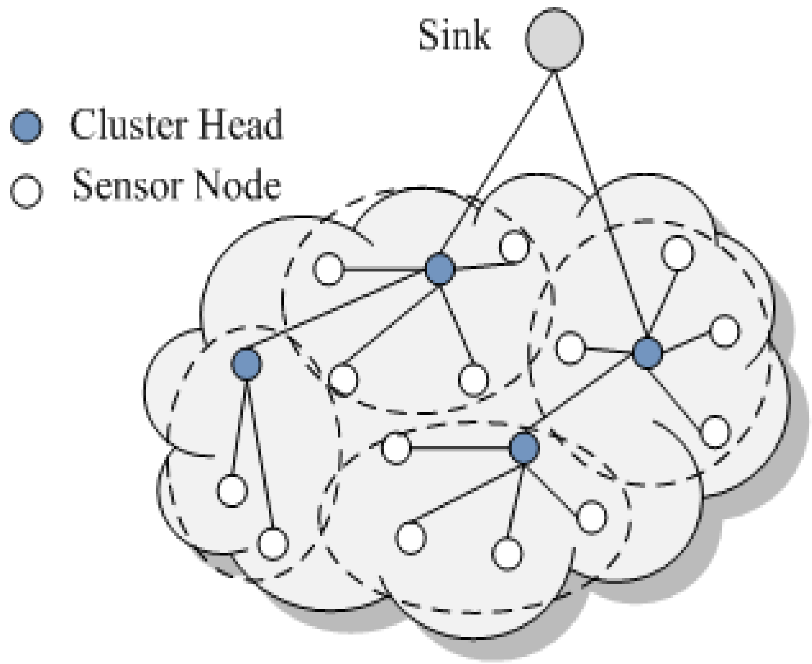


Figure 2.5: Multihop Clustering Architecture[33]

- **Location-based Routing**

In location-based structure, the nodes are addressed by their location. By utilizing the strength of the received signal the distance between the nodes can be estimated. In addition, the sensor nodes may be equipped with a small low-power GPS receiver through which the location of the sensor can be obtained directly.

2.2.2 Protocol Operation

According to the operation of the protocol, wireless sensor routing protocols can be classified as follows[5]:

- **Negotiation-based routing**

In negotiation based, before forwarding the actual data, negotiation will takes place between the source and the sink. This is to reduce redundant data transmissions as well as reducing the energy consumption.

- **Multipath-based routing**

Multiple/alternate paths between the source and the destination will be provided for utilization whenever the main path fails, in order to enhance the performance of the network.

- **Query-based routing**

In query-based routing protocols, the sink node initiates the query message (sensing task) which may include the location of the part of the sensors in which the sink node interested in to get answer from. Most of the time, the query is described in high-level language, for example, the sink may initiates the query: "*Are there moving vehicles in battle space region 1?*" The query will be broadcasted from the sink node through the network, and whenever the sensor node detects a matching data for the query, it sends the data back to the sink node.

- **QoS based routing**

In QoS-based routing, the protocol is concerned about fulfilling QoS metrics (delay, energy, bandwidth, etc) when the messages are forwarded to the sink node. QoS focuses on reducing the energy consumption while delivering good quality data.

- **Coherent-based routing**

In coherent-based routing, by utilizing various types of data processing techniques, the node sends the data to aggregators for further processing. The nodes in the sensor field cooperate with each other to process the data.

2.3 Clustering Routing Protocols for WSNs

The use of clustering in routing protocols for WSNs has been studied in many research works. In those studies, clustering methods have been shown to make great improvements and have helped to decrease the amount of energy consumption and overhead data, which are normally generated during relay of information from the sensor nodes to the base station[34, 35].

2.3.1 Heuristic Schemes

2.3.1.1 Linked Cluster Algorithm (LCA)

The literature studies have shown that LCA is one of the first heuristic algorithms developed. It was primarily developed for wired sensors, and has been used in Wireless Sensor Networks. LCA concentrates on maximizing the network connectivity. It consists of two steps: cluster formation and cluster linkage. The cluster formation process is, the first cluster head is elected is randomly, (node 4, in cluster 1 in Fig. 2.6). Based on the communication range of the node, a circle is drawn, and then every node within the range joins the cluster head. From the nodes which do not belongs to cluster 1, another node is elected as a cluster head (node 13, in cluster 2), and same as before a circle is drawn around a new cluster head. At this time, every node in communication range of node 13 and do not member of cluster head 4, joins the cluster head as member. The process is repeated until all the nodes is covered and belongs to one of the clusters. In LCA every node has its own elected cluster head, which acts as the controller, and at least belongs to one cluster. In LCA a large routing overlay is produced as it induces a great number of cluster heads to ensure a maximum inter-luster connectivity. After the cluster is formed, three roles will be assigned for each node. A node will be either gateway (node 5), cluster head (node 4), or an ordinary node (node 3) as shown in Figure 2.6. The cluster heads are connected to each other through the gateways to form the backbone, which will be used for inter-cluster communication[36-41]. The following data structures will be maintained by each node as described in Figure 2.6:

- *Heads_one_hop_away:* is a list recording those cluster heads that are connected to a node (one-hop away)

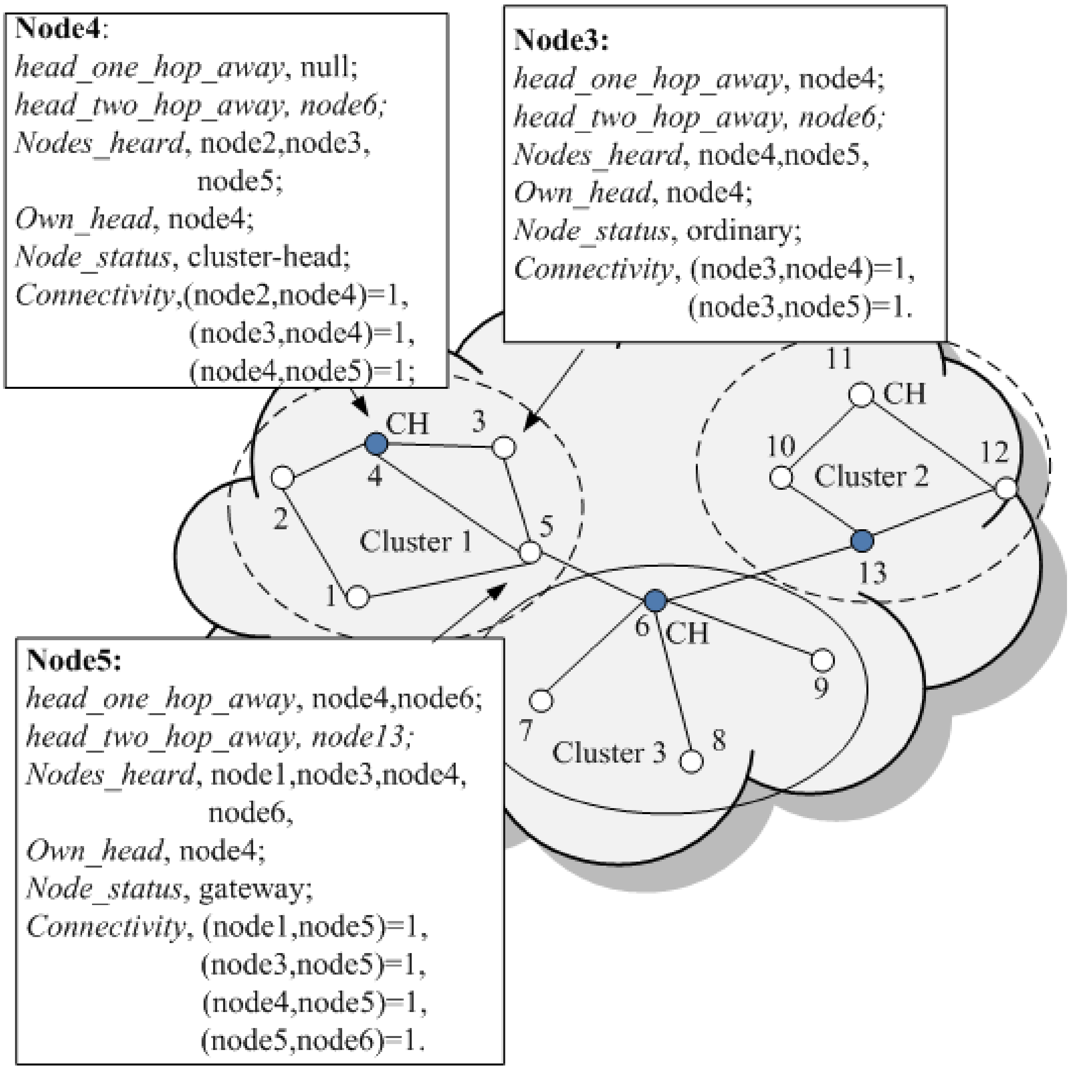


Figure 2.6: Clustering by the LCA[33, 37, 38].

- *Heads_two_hops_away:* is a list of the cluster heads that are not directly connected but connected to the neighbors of a node (two-hops away)
- *Nodes_heard*: is a list that includes all neighbor nodes
- *Own_head*: is the identity of the cluster head of the given node
- *Node_status*: indicates the status either (ordinary node, cluster head or gateway)
- *Connectivity:* is a matrix having binary entries. A value of 1 at the (i, j) position shows that there is a link between nodes i and j, 0 is for no connectivity

2.3.1.2 Linked Cluster Algorithm 2 (LCA 2)

LCA 2 was developed to omit the election of an unnecessary number of cluster heads, similar to LCA. In this algorithm, the idea of a node being covered and non-covered was introduced. When one of its neighbors is a cluster head, a node is considered covered. The election of cluster heads begins with the node that has the lowest ID among non-covered neighbors [37, 40]. As we can see form Figure 2.7, for cluster 1 node 1 is elected as a cluster head, node 10 and node 6 are elected, for cluster 2 and cluster 3 respectively.

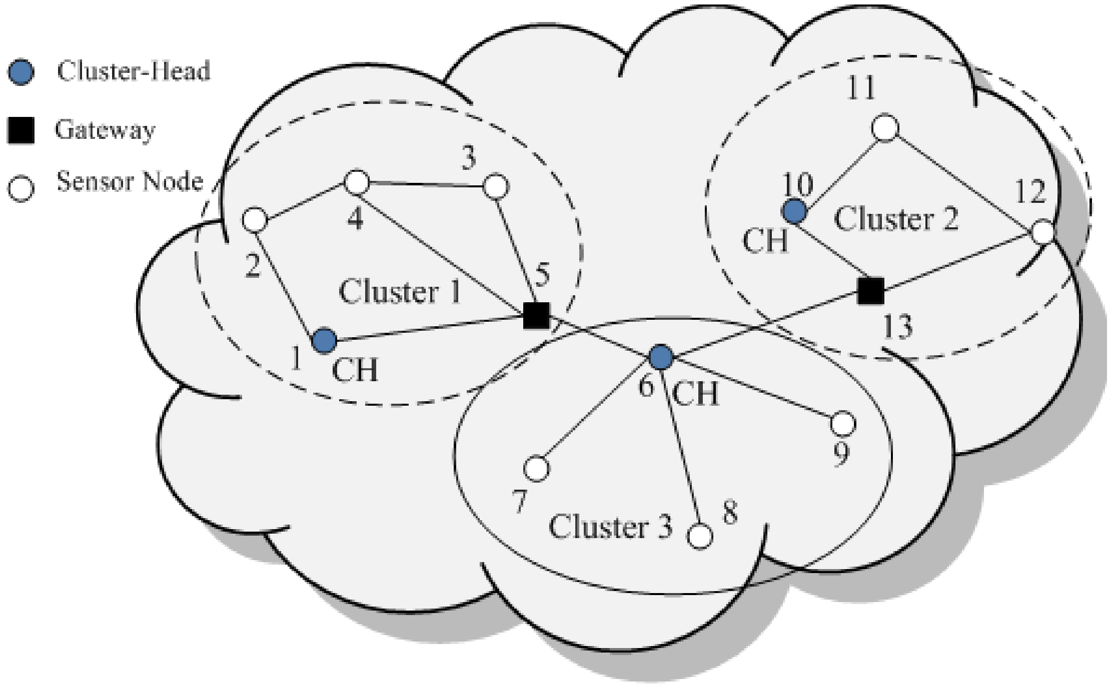


Figure 2.7: Example of cluster formation (Lowest-ID) [33, 37]

The drawbacks of both LCA and LCA2 are, the cluster head load is not evenly distributed. Moreover, in LCA as the nodes have to disseminate for their all neighbors about *nodes_heard*, this creates a high message overhead. Additionally, in LCA the mobility of the nodes as well as the energy is not considered [33].

2.3.1.3 Highest-Connectivity Cluster Algorithm

Highest Connectivity Cluster Algorithm is similar to LCA. In this algorithm, every node broadcasts to the surrounding nodes the number of neighbors which the node is connected to. This makes the connectivity of a node as the factor to be considered, instead of looking at the ID number. The node that connects to the most number of nodes (nodes with the highest connectivity) is elected as the cluster head [40]. Figure 2.8 shows, cluster formation process which is applied to the same network structure shown in Figure 2.7. In cluster 1 either node4 or node5 can be a cluster head, as both of the nodes have equal number of nodes as neighbors. For cluster 2 and 3, node 13 and nodes 6 are elected as the cluster heads, respectively. In Highest-Connectivity cluster, a large number of messages between the sensor nodes are exchanged, which creates a higher message overhead while comparing it the Lowest-ID [33].

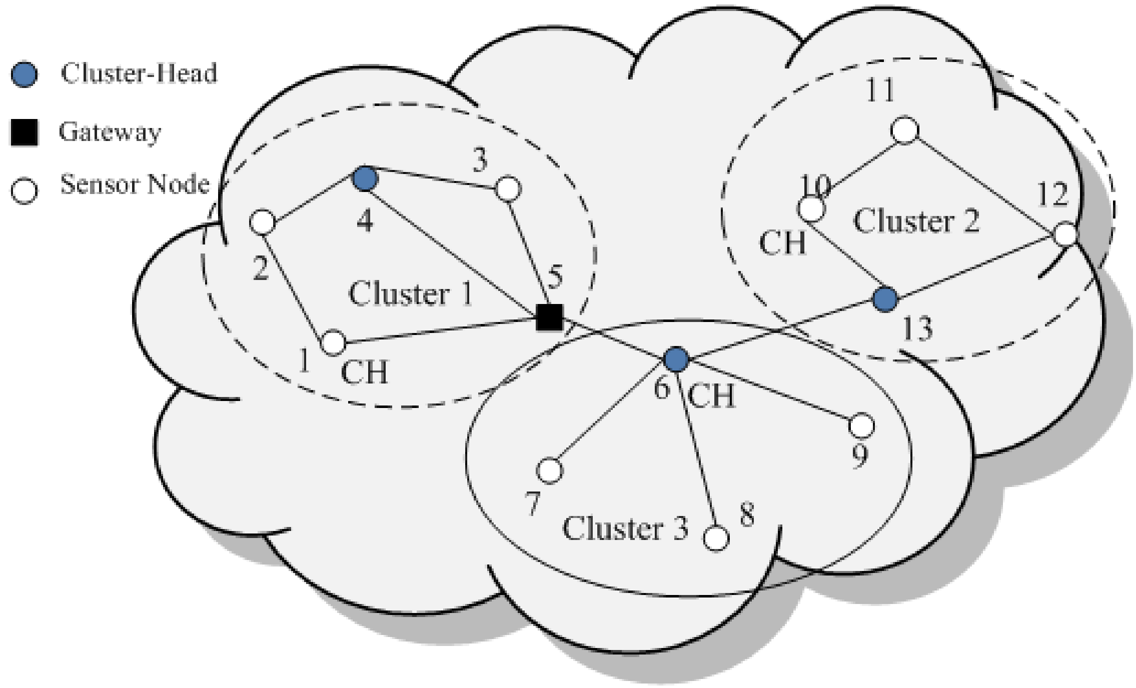


Figure 2.8: Example of cluster formation (Highest-connectivity)[33, 37, 40]

2.3.1.4 MAX-MIN D-Cluster Algorithm

In this algorithm, a new distributed cluster head election procedure has been proposed. Here a node must not be more than d (d is a value selected for the heuristic) hops distant from the cluster heads. The criteria of selecting a cluster head is developed by allowing each node to initiate $2d$ rounds of flooding, from which the results are recorded. Each node then follows an easy set of rules to find their respective cluster head. The first d rounds are used to propagate the largest node IDs and are called *floodmax*. The second d rounds of flooding occur after the completion of the first round. This round is used to let the smaller node IDs to reclaim some of their territory and is called *floodmin*. Each node then assesses the logged entries following the rules listed below [41]:

Rule 1: Each node checks to see if it has received its own ID in the 2nd d rounds of flooding. If it has, then it can claim itself as the cluster head and skip the remaining rules. Otherwise it proceeds to Rule 2.

Rule 2: Each node looks for node pairs. Once this is achieved, it selects the minimum node pair to the cluster head. If a node pair does not exist, they carry on to Rule 3.

Rule 3: elects the largest node ID in the 1st d rounds of flooding as the cluster head for this node.

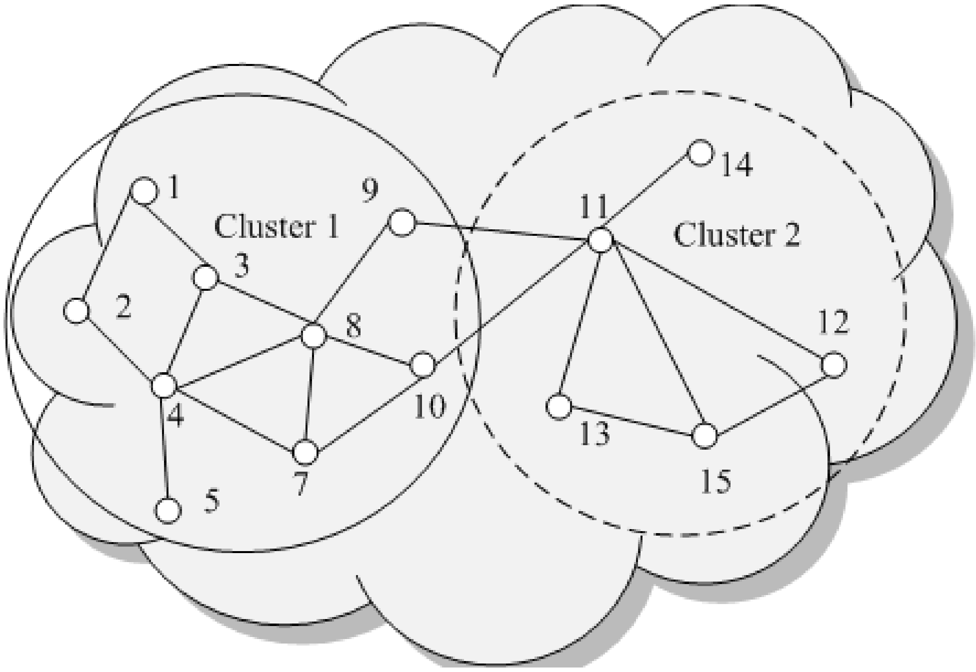
After the node has finished following the rules, it needs to determine whether it is a gateway node or not. This is achieved by broadcasting to its neighbors a listing of its elected cluster head. A node can figure out whether it is a gateway or not after getting responses from all of its neighbors. Communication between each node and the cluster head starts when the gateway node is found. This is fulfilled by sending a message inward from the gateway node. This message includes its node ID, all neighboring gateway nodes and their connected cluster heads.

The correctness of this algorithm depends on the following two assumptions:

Assumption 1: During the *floodmin* and *floodmax*, no node ID will be propagated further than $d - hops$ from the originating node itself (definition of flooding).

Assumption 2: All nodes that survive the *floodmax* elect themselves as cluster heads.

These assumptions have been proved by the authors in[41].



Node	1	2	3	4	5	7	8	9	10	11	12	13	14	15
floodmax1	3	4	8	8	5	10	10	11	11	14	15	15	14	15
floodmax2	8	8	10	10	8	11	11	15	15	15	15	15	15	15
floodmin1	8	8	10	10	8	8	11	15	15	15	11	11	11	11
floodmin2	8	8	8	8	8	8	11	8	8	11	11	11	11	11
Result	8	8	8	8	8	8	8	8	8	11	11	11	11	11

Figure 2.9: Clustering by the max – min 2 - clustering algorithm[33]

2.3.2 Weighted Schemes

2.3.2.1 Weighted Clustering Algorithm (WCA)

This algorithm is a non-periodic procedure for the cluster head election. It is needed every time a reconfiguration of the network topology is unavoidable. During the first cluster head election, this algorithm attempts to find a long-lasting architecture. If a sensor loses the connection with any cluster head, an election procedure is needed to find a new clustering topology. The fundamentals of this algorithm are combined metrics that takes into account several system parameters such as: the ideal node degree; transmission power; mobility; and the remaining energy of the nodes. Based on the specific application, these parameters can be used as a metric to elect a cluster heads. One of the most essential aspects of this algorithm is that it is fully distributed, which means that all nodes in the mobile network share the same responsibility by acting as cluster heads [42].

Cluster head election procedure is based upon a global parameter that is called *combined weight*, which is explained by[42].

$$W_v = w_1 \Delta_v + w_2 D_v + w_3 M_v + w_4 P_v \qquad \text{Eq. 2.1}$$

Where w_1, w_2, w_3, w_4 are the weighing factors for the corresponding system parameters. The component $w_1 \Delta_v$ helps MAC to function efficiently as it is always preferable to determine the maximum number of nodes that the cluster head can handle in its cluster. D_v represents the sum of the distances between each node and all of its neighbors, and is mostly related to energy consumption.

M_v represents the mobility of the nodes. In order to have more constant cluster head architecture, it has been accepted that the cluster head must move very slowly. From this perspective, a node that moves slowly is always preferable to be the cluster head. P_v is directly related to the energy available in a node. The combined weight is updated periodically by each node and broadcasted across the network. The node with minimum W_v is elected as the cluster head. A node should be ignored for the next cluster head election, if it has been chosen as the cluster head earlier. This is because it may have expended a large amount of energy. The algorithm provides selection of the weighing factors (w_1, w_2, w_3, w_4) based on the system requirements.

2.3.3 Hierarchical Schemes

2.3.3.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

Low Energy Adaptive Clustering Hierarchy (LEACH) is the first clustering routing protocol proposed for WSNs [43].The main characteristics of the LEACH protocol is splitting the whole network into clusters and randomly selecting Cluster-Head (CH) for each cluster, depending on their energy load. For every round, 5 % of the nodes becomes a CH. LEACH protocol is broken into rounds, each round starts with the set up phase (cluster formation) and steady-state phase(data transmission). Details of the LEACH protocol are as follows:

a. *Setup Phase*

In this phase as the clusters are being created, each node chooses random number between 0 and 1 to decide whether or not to be a cluster head for the current round, and it compares the chosen number with the threshold given by the equation:

$$T(n) = \begin{cases} \dfrac{P}{1 - P * (r \bmod (1/P))} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \qquad \text{Eq. 2.2}$$

Where n is node ID, p is the desired percentage of cluster-heads in the network (that is, 5%), r is the current round and G is the set of nodes that have not been a cluster head in the last $(1/p)$ rounds. If the chosen random number is less than the calculated threshold $T(n)$ then the node is selected to be a cluster head for the current round.

b. *Cluster Formation*

The cluster formation starts by the elected cluster heads advertising their roles to the rest of the nodes in the network. After the cluster heads have been selected, each cluster head broadcasts an advertisement to other nodes. Based on the strength of the received signal from the cluster heads, each node then decides which cluster head it wants to join and, it uses CSMA MAC protocol to send the message back to the cluster head. The cluster head creates and broadcasts a TDMA schedule for all members in its cluster. The TDMA schedule is used by each node to send data to its cluster head avoids data collision among data messages, and helps to turn the radio off for each non-cluster head node at all times except during its transmission time. Figure 2.3 shows the details of cluster formation phase (cluster set-up and schedule creation).

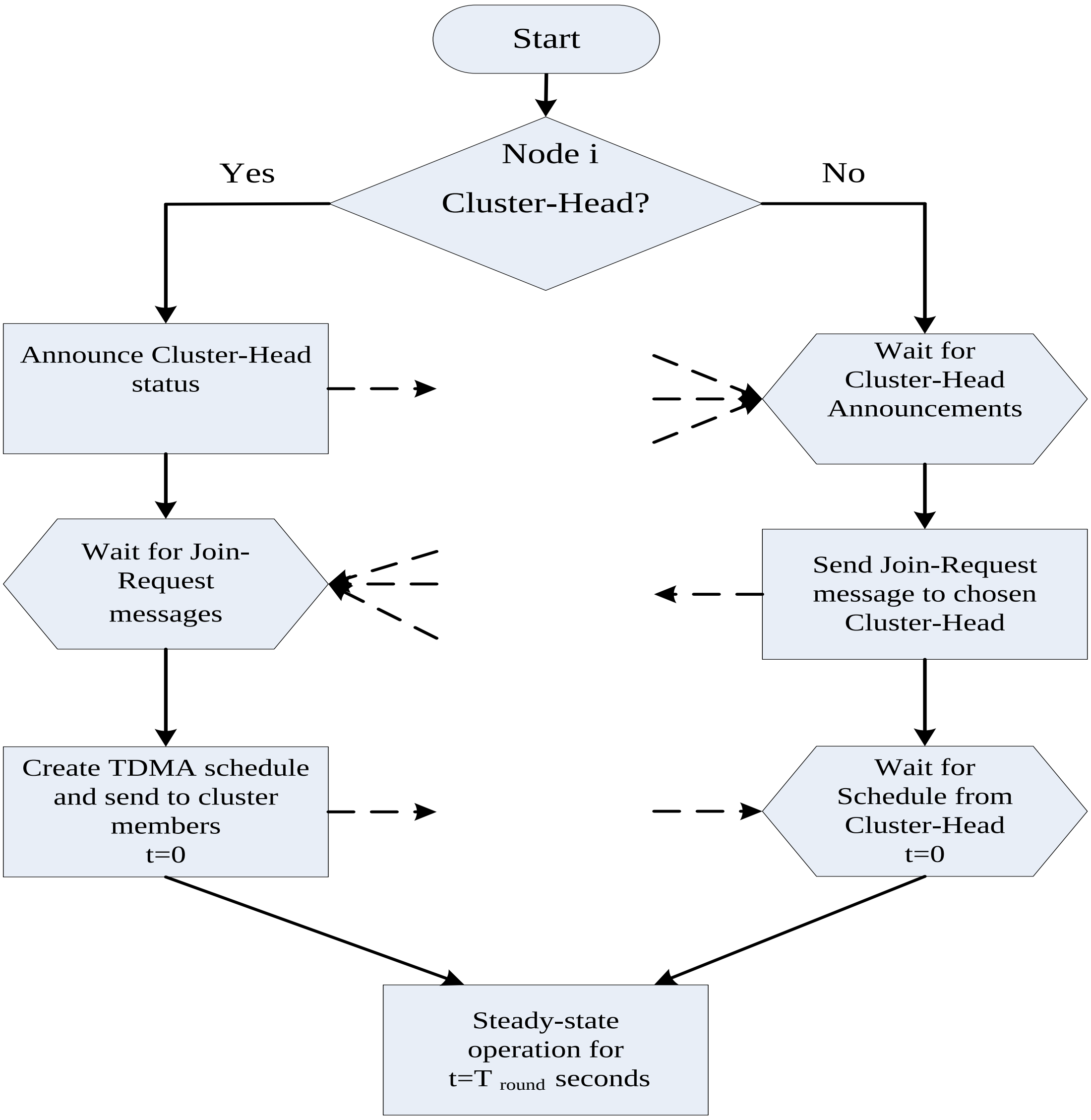


Figure 2.10: Flow of set-up phase [25]

2.3.3.2 LEACH-C: LEACH-Centralized

LEACH-C (LEACH-Centralized) is a variation of LEACH, which uses a centralized clustering algorithm for formation of the clusters. Better clusters can be produced by dispersing the cluster-head nodes throughout the network. The assumption and characteristics of this protocol during the steady-state phase is the same as the original LEACH. In LEACH-C the nodes send their current location and energy level to the base station at the end of cluster head change period. For this reason, the cluster formation, appointment of fixed number of nodes as cluster head and even distribution of the number of nodes in each cluster can be optimized by the base station that has global network view. Computation of average energy for each node is possible via the base station, and whichever nodes that have energy below this average cannot be the cluster head for the current round. The simulated annealing algorithm is run by the base station to select the cluster heads and the associated clusters with the remaining sensor nodes. Once the cluster heads and the associated clusters are found, the base station broadcasts a message that contains cluster head ID for each node. If a node's cluster head ID matches its own ID, the node becomes a cluster head. Otherwise, the node determines its turn in TDMA schedule and goes to sleep until it is time to transmit data. The data transmission phase for LEACH-C remains the same as in original LEACH [43].

2.3.3.3 LEACH-F: LEACH with Fixed Cluster, Rotating Cluster-Head

LEACH-F (Fixed Cluster, Rotating Cluster-Head) is another variation of LEACH protocol which uses LEACH-C characteristics to set-up clusters and select cluster head nodes for the first round. The main advantage of LEACH-F is once the cluster formation has been completed, there is no set-up overhead at the beginning of each round. When the first round is complete the clusters are fixed, only the cluster head nodes are rotated. The steady-state phase for LEACH-F remains the same as in original LEACH protocol. Due to the fixed clusters, LEACH-F is not practical for dynamic systems, because it does not allow new nodes to be added to the system and does not adjust its behavior based on dying nodes [16].

2.3.3.4 Two-Level Hierarchy LEACH (TL-LEACH)

This is a proposed extension to LEACH algorithm. It has two levels of cluster heads, primary and secondary, in addition to the other easy sensing nodes. The primary cluster head in each cluster communicates with the *secondaries*, where the corresponding *secondaries* communicate with the nodes in their sub-cluster. TL-LEACH data-diffusion is performed as in LEACH. Also, communication inside a cluster is still listed using TDMA time-slots. In this algorithm, by conducting the same method as in LEACH, the creation of a round consists of first choosing the primary and secondary cluster heads that have less prior probability of being promoted to a primary cluster head than that of a secondary node. Communication of data from the source node to the sink is attained in two steps:

1) Secondary nodes obtain data from the nodes in their respective clusters. Data-fusion can be performed at this level.
2) Primary nodes obtain data from their respective secondary clusters. Data-fusion can be executed at the primary cluster head level.

The main benefit of this algorithm is it efficiently reduces the total energy usage. This is achieved with the two-level structure, where the amount of nodes that need to transmit to the base station is reduced[44].

2.3.3.5 Energy Efficient Clustering Scheme (EECS)

EECS is a clustering algorithm in which cluster head candidates compete for the ability to be promoted to cluster head for a given round. This competition includes candidates broadcasting their remaining energy to the neighboring candidates. A node becomes a cluster head in case it does not find a node with more residual energy. In this algorithm, the formation of cluster is different as compared to that of LEACH. Formation of clusters in LEACH is based on the minimum distances of nodes to their corresponding cluster head. EECS prolongs this algorithm by dynamic sizing of clusters according to the cluster distance from the base station. This gives rise to an algorithm that deals with the problem that clusters at a greater range from the base station need more energy for transmission than those that are nearer. This upgrades the distribution of energy throughout the network, thus giving better resource usage and longer network lifetime[45].

2.3.3.6 Hybrid Energy-Efficient Distributed Clustering (HEED)

HEED is a multihop clustering algorithm for WSNs that concentrates on efficient clustering by appropriately selecting cluster heads based on the physical distance between nodes [46].

The major objectives of HEED are to:

- distribute energy consumption to prolong network lifetime,
- minimize energy during the cluster head selection phase,
- minimize the overhead control of the network.

Cluster heads are found according to two essential parameters:

1) The residual energy.
2) Intra-cluster communication.

2.3.4 Grid Schemes

2.3.4.1 Power-Efficient GAthering in Sensor Information System (PEGASIS)

PEGASIS is a data-collecting algorithm that creates the concept that energy restoration can result from nodes not directly making clusters. This algorithm gives the concept that if nodes form a chain from source to sink, only one node will be transmitting to the base station in any given transmission time-frame. The occurrence of data-fusion is at every node in the sensor network, permitting all relevant information to permeate across the network. Moreover, the average transmission range needed by a node to relay information can be lesser than in LEACH, resulting in energy improvement versus the hierarchical clustering approach[47].

2.4 Query Processing and Data Aggregation

A user requests a sink node or some sensor nodes in specific region of the network for relevant information in a query format, which is also known as a spatio-temporal query. Based on the routing method implemented after the source node receives the query message, it forwards the data to the sink through multihop intermediate nodes. There are two types of routing techniques utilized: address-centric (AC) and data-centric (DC) routing:

Address-Centric: In AC routing, if the source sensor node, in the location which is included in the query message, has relevant information about the query recently sent from the sink node, it replies individually to the sink through the optimal path which is selected by the nodes. However, AC routing protocols do not support data aggregation and consumes a great amount of energy to forward data to the sink node. Thus, AC routing protocols are not energy-efficient and reduce the lifetime of the sensor node [33, 48].

Data-Centric: In DC routing protocols, in contrast to AC routing method, the sensor node can perform data aggregation to reduce the redundancy of data and saves the energy as well. The intermediate or the cluster head can aggregate the data from different multiple source nodes and forwards toward the sink node. In this way, utilizing DC routing method can save the energy and prolong sensor's lifetime[33, 48]. Directed Diffusion (DD) is the most well-known data-centric routing protocol with which we compare our proposed method. The next section gives details of DD and the mechanisms it exploits to forward the query, selects the best path and retrieves the data back from specified region of the sensor field.

2.4.1 **Directed Diffusion**

2.4.1.1 Naming

As described in DD, tasks are represented or named using attribute-value pair [24]. The query message which is generated from the user which will be sent to the sensor in specified region is named using the attribute-value pair. The sink may generate the following query and send to the sensor nodes:

Type=Wheeled vehicle //Detect Vehicle location
Interval=20 ms //send events every 20 ms
Duration=10 seconds //send data till this time
Rectangle= [-100, 100, 200, 400]

The task describes the data (matching for the attributes) which is interested by the user, so it is called an *interest*. The data sent in response to this query is also named using the same scheme, attribute-value pair. For example the sensor that detects the wheeled vehicle or an animal might respond like,

Type=Wheeled vehicle //type of vehicle seen

Instance=truck //instance of this type

Location= [125, 220] //node location

Intensity= 0.6 //signal amplitude measure

Confidence= 0.85 //confidence in match

Time Stamp= 01:20:40 //Event Generation Time

2.4.1.2 Interest Propagation

DD routing protocol is constructed from four parts, *Interest dissemination, gradients, reinforcement* and *data propagation.* The interest message is initiated and broadcasted by the sink node to the rest of sensor nodes in the network. Every sensor nodes receives the interest message stores in it's the interest and the neighbors from which the interest is received in the interest cache. Each sensor node is task-aware and any node matches the specified region will reply with the relevant data and the node will be the source for the recently sent interest. Each source node will build the gradients towards the sink nodes through the sensor nodes from which the interest message has been received.

- **Interest Cache**

The interest cache is used to store the recently sent interest which contains necessary information for routing the data back to the sink node. Each node broadcasts the received interest to all of its neighbors which helps the nodes to activate the sensing information. Every time the node receives new interest it checks its cache, if the interest is already exists or if it is the refreshment. In case if it is the refreshment, the node updates the necessary information. For each interest in an interest cache there is a gradient which represents the direction to which the data should flow. After the specified time for each interest elapsed, the interest will be removed from cache.

- **Refreshment of Interest**

Since broadcasting the interest is not a reliable transmission, the sink node periodically refreshes the interest message by re-sending the same interest to the sensor nodes by increasing timestamp attribute[24]. Directed diffusion[24] disseminates the queries across the entire networks to retrieve information from specific region of the sensor field. The named interest which is generated by the sink node is represented using attribute values. The following task may be sent from the base station: "In every time *Interval* (ms) for the next *Duration* (Sec), send me the

estimation location of any four-legged animal in sub-region *Rect* of the sensor field".
The interest message can be presented in the following format:

Type = four - legged animal // detect animal location
Interval = 20 ms //send back events every 20 ms
Duration = 10 seconds // … for the next 10 seconds
Rect = [− 100, 100, 200, 400] // from sensors within rectangle

Or, for instance, if the sink node wants to receive the temperature from a particular region of a rectangular during some specific time the interest message may appear: *type = temperature, interval = 10 ms, duration= 40 s, and Rect = [-200,200,400,600].* Each node that receives the interest message will build the gradient towards the sensor node from which it receives the query of the interest message. The gradient expresses the direction in which the data are to be forwarded. By increasing a timestamp attribute, the sink node rebroadcasts the same interest message to guarantee reliability. In each node, an interest cache is maintained which contains the following items [24, 33]:

- Timestamp of the last received matching interest.
- A gradient for each neighbor showing the data rate requested by the neighbors.
- Lifetime of the interest message.

As we can see from Figure 2.11, the sink initiates and broadcasts the interest message to the whole network.

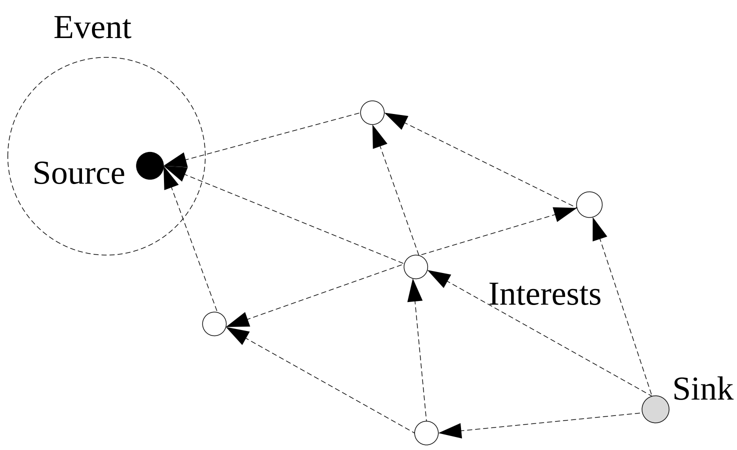


Figure 2.11: Interest propagation for directed diffusion[49]

2.4.1.3 Gradient Establishment

The gradient field shows a value and a direction to which the data will be forwarded to the corresponding neighbor, from which the interest message is received. In DD the gradient value is the data rate. Figure 2.12 presents the set up the gradients among the sensor nodes.

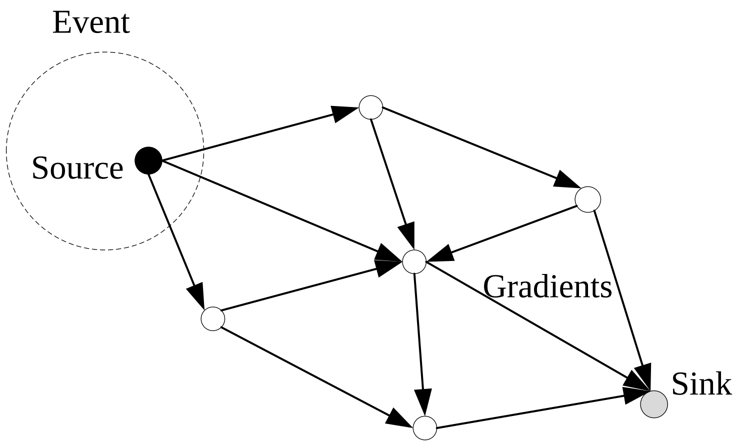


Figure 2.12: Initial gradients setup for directed diffusion[49]

2.4.1.4 Positive Reinforcement and Negative Reinforcement

In DD, the sink repeatedly broadcasts an interest for low-rate event, which is called *exploratory* events. These events are intended to build the path between the sensor nodes, which detect the matching data for recently sent interest, and the sink node. The match data for the exploratory events might be forwarded (possibly along multiple paths) towards the originator of the interest message which is the sink. When the sink node receives the exploratory data from many sensors; which are neighbors of the sink, it reinforces at least one of the neighbors based on the criterion of the application, for example, the lowest delay or lowest energy. The neighbors also take their turn to reinforce their neighbors according to the same criteria which has been used from the sink node. The real data from the source node to the sink node will be forwarded only along the reinforced path between them. Figure 2.13 presents the third stage of DD, which is reinforcing the best path among multiple paths. The rest of the gradients or the paths will be negatively reinforced to suppress loops or duplicate paths.

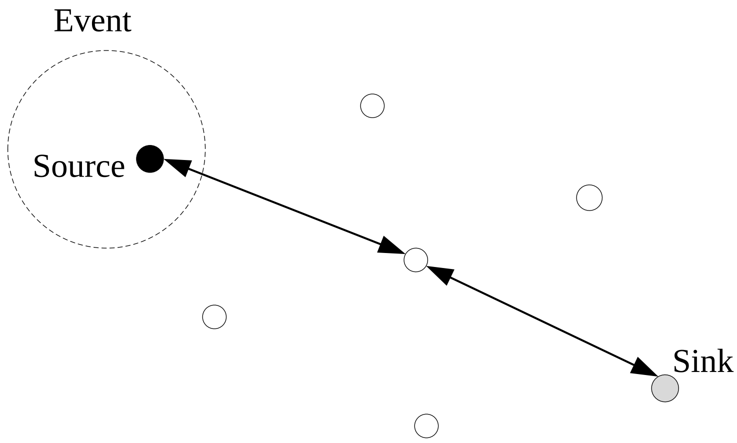


Figure 2.13: Path reinforcement for directed diffusion [49]

2.4.1.5 Data Propagation

- **Data Cache**

In DD like an interest cache there is a data cache which is used to suppress duplicated data messages which avoid loops of the data message. Cache is can be utilized in application-specific scenarios, for in-network data processing and data aggregation. After the specified time by the originator of the interest elapsed the interest message will be removed from the cache.

Every sensor node which detects a target searches insides an interest cache for a matching interest entry. Based on the applications, the matching criterion will varies. For example, in DD the entry matches a target if the type is the same and the rect in which the sink node interested in is encompasses the estimated target's location. After the sensor node completes the matching process based on the interest message, the data will be generated and sent over the reinforced path towards the sink. The data that will be generated depend on the interest message and can be described as the followings[24]:

Type = four-legged animal // detect animal's location

Instance = elephant // instance of this type

Location = [125, 220] // estimated location

Intensity = 0.6 // signal amplitude measure

Confidence = 0.85 // confidence in the estimate

Timestamp = 01:20:40 // event generate time

In response to the interest message, the data which are also named using attribute value are generated, and eventually they are propagated towards the sink node over the gradients being established during the propagation of the interest message. Figure 2.14 shows the propagation a data message from the source node to the sink node along the reinforced path.

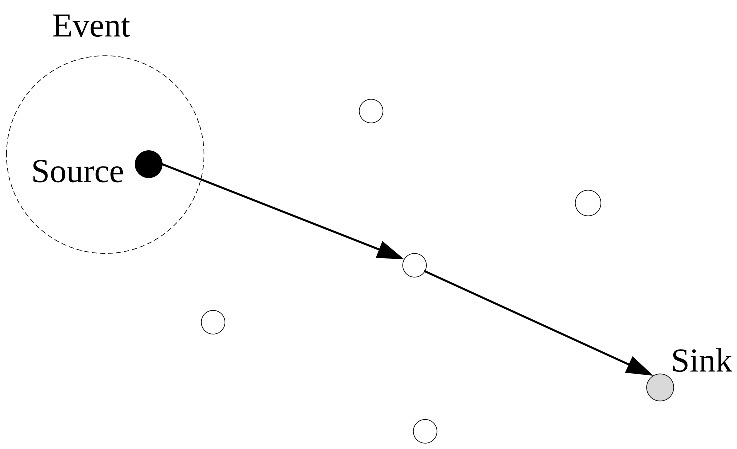


Figure 2.14: Data delivery along reinforced path for directed diffusion[49]

Some of the features for DD routing protocol in base of WSNs are:

- Data is named utilizing attribute-value pairs
- In-network processing, data aggregation and propagation by the cooperation of neighbor nodes
- Data-centric
- Scalable
- Robust
- Energy-efficiency
- Multi-path capability

- Based on query-driven data delivery model

2.4.2 PCDD [28] and ECPC [29]

As enhancement of the original DD, a passive clustering[28] which utilizes a mechanism of building the clustering on demand to reduce the overhead due to interest propagation flooding and set-up time. This method has been proposed by using tree, which will be formed during the propagation of the exploratory data packets, between the source and the sink. This is used to forward the data to the destination only by using some parts of the tree to reduce the energy consumption and the delivery delay between the source and the sink. However, once the sensor nodes' energy is depleted in some part of the network, the tree also needs to be updated, which needs more energy. In addition it has limitation for mobility and ignores energy consumption for "*Idle*" and the use of *First declaration wins*, which is used to select the cluster heads, may partition part of the network as the selected cluster head might has low energy. ECPC[29] is an enhancement of PCDD which considers distance between and the remaining energy for cluster head election and forming the clusters, there improves the routing approach over PCDD. However, ECPC creates extra-overhead while forming clusters and it is very limited in supporting mobile nodes.

2.4.3 EPCDD [30] & ELPC [31]

The paper presented in[30, 31] argue that, the concentration of PCDD and ECPC [29] on set of nodes to perform the flooding may lead to a different in energy consumption level among the sensor nodes. They declare that this may generate a partition in some region of sensor filed. To control the problem, EPCDD has been proposed to use the energy consumed rate for the selection of the flooding nodes. For ELPC they propose energy level for each node.

2.4.4 DDLSC [23]

The authors of [23] proposed the delay-energy aware routing protocol as enhancement over DD. The paper focuses on real-time applications, which need shortest delay. DDLSC enables tradeoffs between delay and energy consumption. DDLSC builds a virtual backbone from the elected cluster heads for multi-hop communications. The authors stated that DDLSC improves delay and energy while comparing it to other data-centric routing protocols. However, the use of long transmission range between the cluster heads and the nodes as well as ignoring the idle energy consumption leads DDLSC to consume more energy while transmitting the data between the cluster heads or between the cluster member and the cluster heads.

2.5 Performance Analysis and Comparison of Four Routing Protocols

This section presents the performance analysis for four existing routing protocols using different metrics. As discussed in section 2.2, WSNs routing protocols can be classified into: hierarchical-based, flat-based and location-based according to network structure. The performance analyses of WSNs routing protocols that have done in this work includes hierarchical-based routing protocol, LEACH, flat-based routing protocol, DD, location-based routing protocol, Greedy Perimeter Stateless Routing (GPSR)[26] and Ad-Hoc On-Demand Distance Vector (AODV)[27].

GPSR is a geographic routing in which the nodes utilize the location of the sensor nodes to deliver data to the destination. In GPSR each node saves its own location and the locations of all of its one-hop neighbors. Two kinds of routing algorithms have been developed in GPSR: greedy forwarding and perimeter forwarding. In greedy forwarding approach, whenever the node receives a data packet for the destination, it selects the closest node to the destination including itself and forwards the data packet. The greedy approach stops working when there is no node close to the destination than the node which has the packet itself. In this case, GPSR switches to perimeter mode in which the node applies the right-hand rule after the planner sub-graph network connectivity is computed[26].

AODV is a reactive routing protocol which computes the route between the source and the destination on-demand, and it is specifically designed for Mobile Ad Hoc Networks (MANET) to avoid the overheads of proactive routing algorithms. AODV does not have built-in energy efficient technique. However, it is successfully integrated with energy-efficient protocols, thus it can be applied on energy-constrained wireless sensor nodes in different applications [50].

The simulation was conducted utilizing J-Sim simulator and three different metrics: average dissipated energy, average delivery latency and packet delivery ratio, have been chosen to conduct the simulation and to evaluate the routing protocols. Table 2.1 shows the comparison of the four protocols, the main features and limitation of each protocol.

Table 2.1: Performance Analysis of Routing Protocols

	DD [24]	LEACH[25]	GPSR[26]	AODV[27]
Network Architecture	Flat	Hierarchical	Location-based	On-demand/flat
Energy-Efficiency	Limited	Maximum	Non-energy-aware	Limited
Mobility	Limited	FixedBS	Possible	Possible
Main Features	I. Best path using reinforcements II. In-Network Data-Processing	I. Clustering II. Data Aggregation III. Saves Energy by even distribution	Delivers more data packets	I. Reactive/on-demand protocol II. Less Over-head
Limitations/drawbacks	I.Interest broad casting/flooding & exploratory data II. Consumes great energy	I. Consider s single hop communication II. Assumes each node has data to forward to BS	Highest energy consumption & Delivery latency	Energy-efficient strategy is not built-in

According to the performance analysis that we have conducted, it is found that LEACH routing protocol shows the best of the four protocols in terms of energy saving and delivery latency. The using of clustering technique and performing aggregation for the data packet at the cluster heads, helps in reducing the energy consumption by decreasing the message exchanges between the cluster heads and the base station. In this way, LEACH can achieve longer lifetime for the sensor nodes. However, LEACH assumes that each node has data to send to the base station according to the set time schedule from each cluster heads to its member, which is not suitable for applications in which the sink node interested in information from some specific region, for example, query-driven techniques. Also, its assumption of one-hop routing between the cluster heads and the base station reduces its functionality for large scale sensor networks.

DD, which is known as robust and scalable routing protocol paradigm in WSNs, can select the best path from multiple paths using the reinforcement strategy. Moreover it has some novel features: data-centric routing technique, in-network data processing and aggregation and data caching. These features can be exploited in dynamic networks in energy-efficient manner and robust data propagation while at the same time minimizing the per-node configuration that is features of present sensor networks[24].

2.6 Summary

There are various types of routing protocols that can be classified into Network Structure and Protocol Operation. In general, Cluster-Based routing protocols which classified under Hierarchical Network structure have shown a good performance to support energy-efficient implementation. In addition, this routing protocol can be scalable to support different number of sensor nodes deployment in the network. However, LEACH which is one type of Cluster Based routing protocol has found not to be scalable.

In general, Flat Network structure is found not to be energy-efficient and scalable. However, DD routing protocol which is classified under Flat Network Structure was found to be the most scalable, robust and the best recognized routing algorithm for routing in WSNs.

CHAPTER 3

CLUSTER-BASED DATA DIFFUSION ROUTING PROTOCOL FOR LARGE-SCALE WIRELESS SENSOR NETWORKS

3.1 Introduction

Wireless sensor networks has various applications: environmental monitoring (forest fire detection), detecting and tracking of objects (animals), health care, etc. These networks also require, based on the applications, low latency and long-lived sensor nodes. The sensors which are deployed for specific-application should fulfil its functionality, prolonging networks lifetime in terms energy saving and data efficiency. In this chapter, the proposed method - Cluster-Based Data Diffusion Routing Protocol for Large-Scale WSNs - is presented and discussed. The implementation of CDD is consists of various procedures: Cluster Head election process, Cluster Formation and Route Setup, Interest Propagation, Data Transmission and Cluster Maintenance.

3.2 Cluster-Based Data Diffusion Routing Protocol Overview

The proposed CDD routing protocol includes the following phases:

I. *Cluster Head Election Phase*: after the sensor nodes are distributed in the network, self-election of the sensor node as a *Cluster-Head* process is achieved by exploiting the equation specified in LEACH protocol;

II. *Cluster Formation and Route Setup phase:* in this step after the cluster heads are elected, each cluster head broadcasts its role to the nodes using fixed signal strength. Based on the transmission range of the node, each node calculates the location in which it belongs to and sends the joining message to it. Each cluster head create TDMA time schedule for each node in its cluster and broadcasts to them;

III. *Interest Propagation Phase:* the originated interest message by the sink node is forwarded to the elected Cluster-Heads and it reaches the ordinary nodes through their Cluster-Heads;

IV. *Data Transmission Phase*: the nodes check if they have data that matches the recently sent interest and forward it to their corresponding Cluster-Head and the data finally reaches the base station through multi-hop Cluster-Heads; and

V. *Cluster Maintenance phase:* finally, after a specific number of rounds the first two phases are repeated to rotate the cluster heads among the sensor nodes. Consequently, the sink node forwards the interest message only to the cluster heads and the other procedures are repeated again.

3.2.1 Assumptions

In the proposed method, some assumptions have been made to show the capabilities of the node.

Assumption 1: All the sensor nodes in the network are homogeneous

Most of the earlier works assume that WSNs platform is homogeneous in which all of the nodes have identical capabilities of computational capacity, communication and limited energy budget, while the heterogeneous network may include sensors with advanced processing and communication capabilities.

Assumption 2: All sensors in the network are location-aware

To know their location in the network, the sensors can use a Global Position System (GPS) if we are working with "high-level resources" devices, or the sensors can use GPS-free location scheme (i.e. location based on triangulation) if we are working with "low-level resources" devices. More information about location schemes used in WSN can be found in[51].

Assumption 3: The sink node is immovable

For the first part of this work the sink node is considered stationary and is placed at the top right corner of the simulation, however, in the extension part, which will be discussed in chapter 5 of this work, the sink node will be mobile.

Assumption 4: Every node in the network has the same initial energy

Assumption 5: Propagation channel is symmetric

The energy consumption for the data transmission between two sensor nodes for both sides is equal

Assumption 6: The base station can talk to every node in the network

At the sink node is not energy-constrained, it can reach each sensor node in the network to send the interest message

3.3 Cluster Head Election Phase

In this section we explain how the nodes are going be elected as a cluster head. The procedures used in LEACH routing protocol have been adopted and followed to elect the cluster heads in each round for the proposed CDD routing protocol. In the conventional clustering protocols, once the sensor is elected as a cluster head, it may deplete it energy because of organizing sensor nodes in the cluster and exchanging a great number of messages for a long time without replacing it by another sensor node to act as the cluster head. Consequently, many sensors lifetime will be ended when the cluster head drains its energy. In LEACH protocol randomized cluster head rotation among the sensor nodes has been included to reduce the energy consumption of the sensor nodes as well as prolong networks lifetime. This assures that, the energy consumption will be evenly distributed among all the sensor nodes, so that no nodes will drain their energy before the others [16].

3.3.1 Cluster Head Election Criteria

In LEACH protocol the nodes organize themselves in clusters as one of them acting as the leader of the cluster in each round[1]. In cluster head election phase, once the nodes are deployed in the field each node chooses a random number between 0 and 1

[1] round: the time which includes organization of cluster heads, transferring data from each node to the cluster heads and then to the sink node

to decide whether or not to be a cluster head for the current round and it compares the chosen number with the threshold $T(n)$ given by equation 3.1.

$$T(n) = \begin{cases} \frac{P}{1 - P * (r \bmod (1/P))} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \qquad \text{Eq. 3.1}$$

Where n is node's ID, p is the desired percentage of cluster-heads in the network (that is, 5%), r is the current round and G is the set of nodes that have not been a cluster head in the last $(1/p)$ rounds. If the chosen random number is less than the calculated threshold $T(n)$ the node is selected to be a cluster head for the current round.

3.4 Cluster Formation and Route Setup Phase

The cluster formation starts by the elected cluster heads advertising their role to the rest of the nodes in the network. After the cluster heads have been selected, each cluster head broadcasts an advertisement to other nodes. Based on the strength of the received signal from the cluster heads each node decides which cluster head it wants to join and it uses CSMA MAC protocol to send the message back to the cluster head, telling the cluster head "I am joining you for this round". Figure 3.1 shows the details of cluster formation phase.

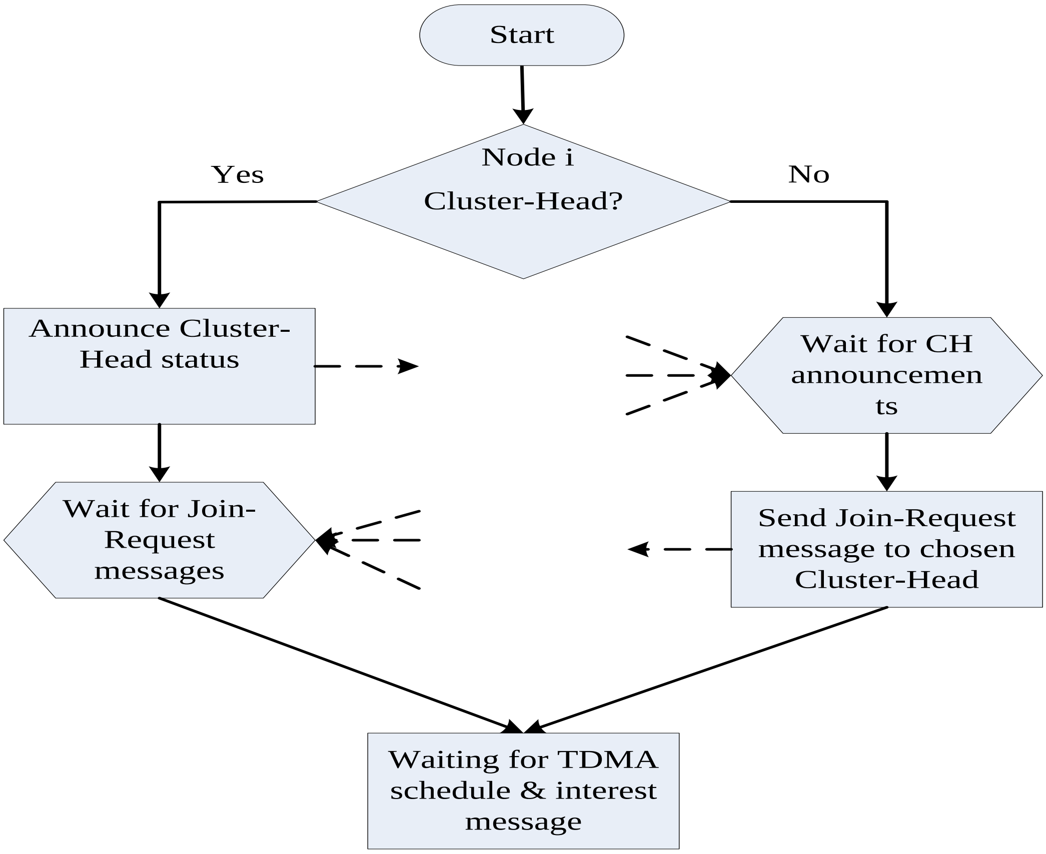


Figure 3.1: Flow graph for cluster setup adopted from[25]

The TDMA-style MAC layer energy-efficient which uses sleep mode to reduce the energy consumption of the idle mode is shown in Figure 3.2 below. As it is stated in LEACH routing protocol that, after the cluster heads are elected, the nodes should communicate with their cluster heads in an energy-efficient manner. This can be accomplished by using the TDMA protocol, in which the nodes can go to the sleep state when there are no data to be forwarded to the cluster heads. Moreover, using TDMA avoids the collision of data which might happen inside the cluster among the nodes which are belongs to the same cluster. In this way the energy and time can be saved.

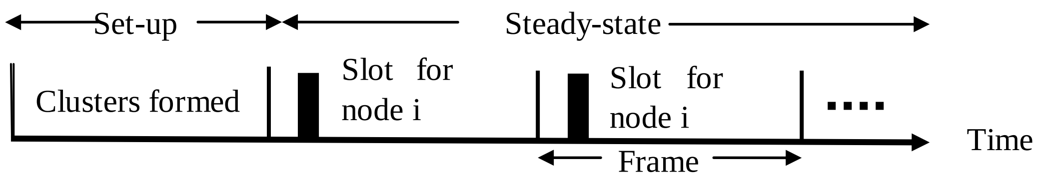


Figure 3.2: Time-line showing the operation using TDMA schedule of each node[25]

3.4.1 Cluster Head Advertisement

If the node has been elected as the cluster head for the current round, it should advertise it role for the rest the sensor node in the network. Each node receives the advertisement message from more than one cluster head selects the closest cluster head to its position and sends a join-request message, Figure 3.1. Therefore, the energy used for the exchanged message between the cluster heads and the non-cluster head nodes will be reduced.

3.4.2 Path Set-up

The path setup between the cluster head and each node has been accomplished after each node selects and sends join-request message to the closest cluster head from the list of cluster heads have been elected in section 3.2.1. The next stage after the sensor nodes select the best cluster head and send join-request message, the cluster head creates TDMA time schedule for its members in the cluster and broadcasts to them. TDMA time schedule will be used whenever the sensor nodes receive the interest message from the cluster head it compares with the sensed data, if match occurred it sends the data back to its cluster head using its TDMA time. If the node does not has match data for the recently sent interest message it ignore it and does not send data to the cluster head. Consequently, to build the multi-hop communication between the cluster heads, from currently elected cluster heads list each cluster head selects the closest cluster head to its position towards the sink node. Figure 3.3 presents the schematic graph after the cluster heads are elected and each node joined the closest cluster head to its position.

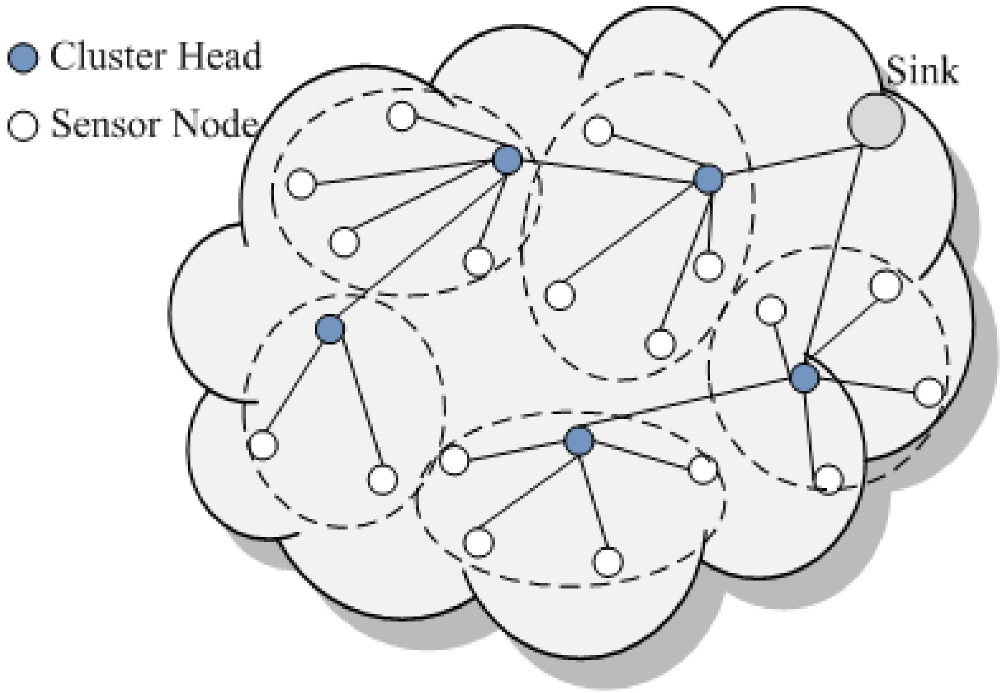


Figure 3.3 CDD multihop clustering

3.5 Interest Propagation Phase

Interest is a group of attribute-value pairs which contains a description of what a user wants. The interest message is defined and injected from the sink node to the network. The original Directed Diffusion protocol uses flooding the interest message to the whole network that is highly reduces the network lifetime as a great number of nodes should be involved in transmission of the interest through the network. Instead of broadcasting the interest message through the whole network as it is in the original DD protocol, in the proposed method the sink node sends the interest message only to the elected cluster heads. The interest message arrives at the ordinary nodes through their cluster heads.

3.5.1 Interest in the Sink Node

The sink node initiates the interest message and sends (unicast) to the elected cluster heads for the current round. In the Directed Diffusion, tasks are defined using attribute-value pairs [20]. An example of the animal tracking task is[24]:

Type = four-legged animal // detect animal’s location

Interval = 10 ms // send back events every 10 ms

Duration = 10 minutes // for the next 10 minutes

Rect = [-100, 100, 200, 400] // from sensor nodes within rectangle

To represent the sub-region, a rectangle is selected and based on some coordinate system (in practice, GPS coordinate system can be utilized)[24].

3.5.2 Interest in the Cluster Heads

Each node in the network including cluster heads and ordinary nodes maintains an interest cache, when each item in the cache coincides to a different interest. Two interests are different from each other as specified in the original Directed Diffusion protocol, if their *type* is differs, or their *rect* attributes are disjoint. Otherwise, the two interests *match* each other.

When the **cluster head** receives the interest from the **sink**, it verifies if the interest is already seen, as it is shown in Figure 3.4.

- If the interest exists in the interest cache, it is a refresh for the recently sent interest and, updates the timestamp and the duration fields. Then,
    - The path is established if it is the source node for the interest message
    - If it is not the source node, it sends the interest message to its member list in the cluster and to the neighbor cluster heads

If the interest does not exist in the interest cache, it creates an interest entry for the received interest and the gradient entry for the sender of the interest. It establishes the path towards the sink if it is a source node. Otherwise, it sends the interest to its member list in the cluster and to its neighbor of cluster heads.

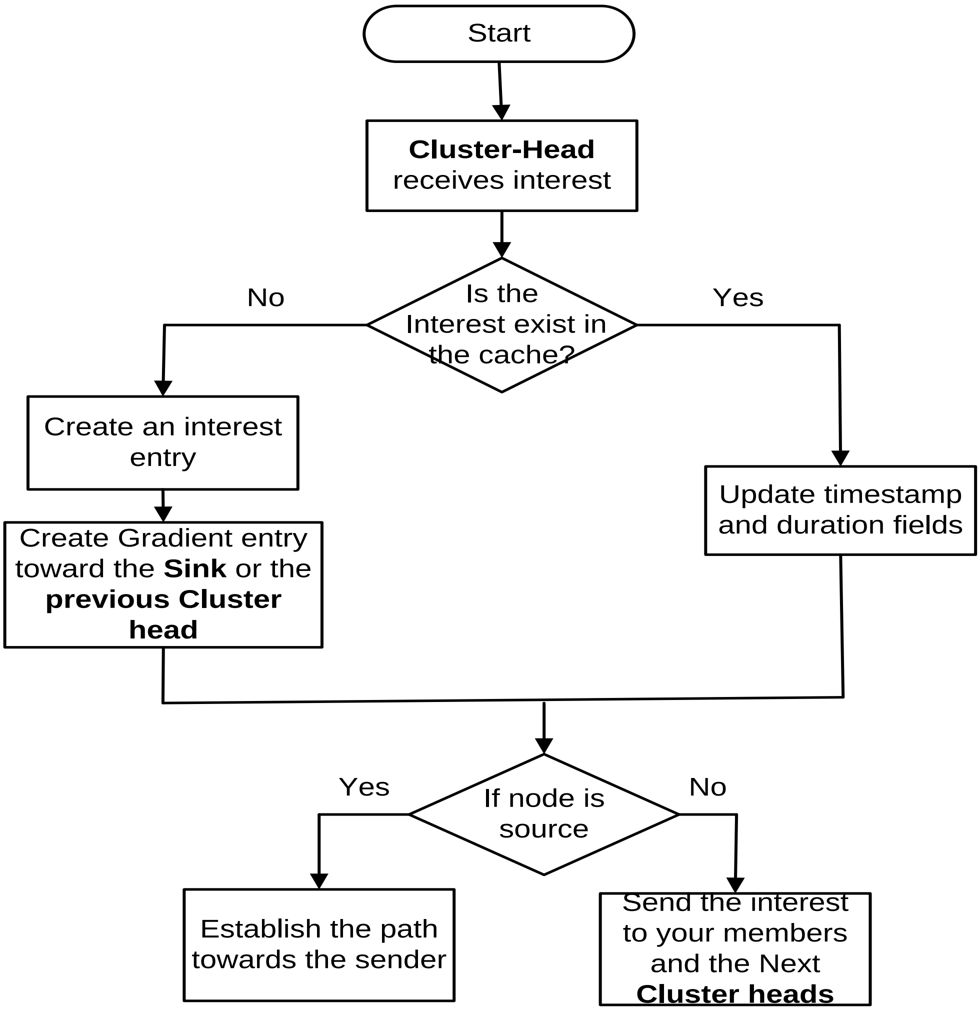


Figure 3.4: Cluster head receives the interest from the sink or previous cluster head

3.5.3 Interest in the Ordinary Nodes

In the ordinary nodes as it is shown in Figure 3.5, when the **ordinary node** receives an interest from the corresponding **cluster head**, the nodes affirms if the interest is exist in its interest cache or not.

- In case the interest exists, the incoming interest is the refresh for the existing interest entry, then
  - The node updates timestamp and duration fields and checks if it is the source node for recently sent the interest.

If it is not the source node, it ignores the interest message. In case it is a source node it establishes the path towards the cluster head from it receives the interest message.

If the interest does not exist in the interest cache, it creates an interest entry for the received interest and the gradient entry for the sender of the interest. It establishes the path towards the cluster head if it is a source node. Otherwise, it ignores the interest message.

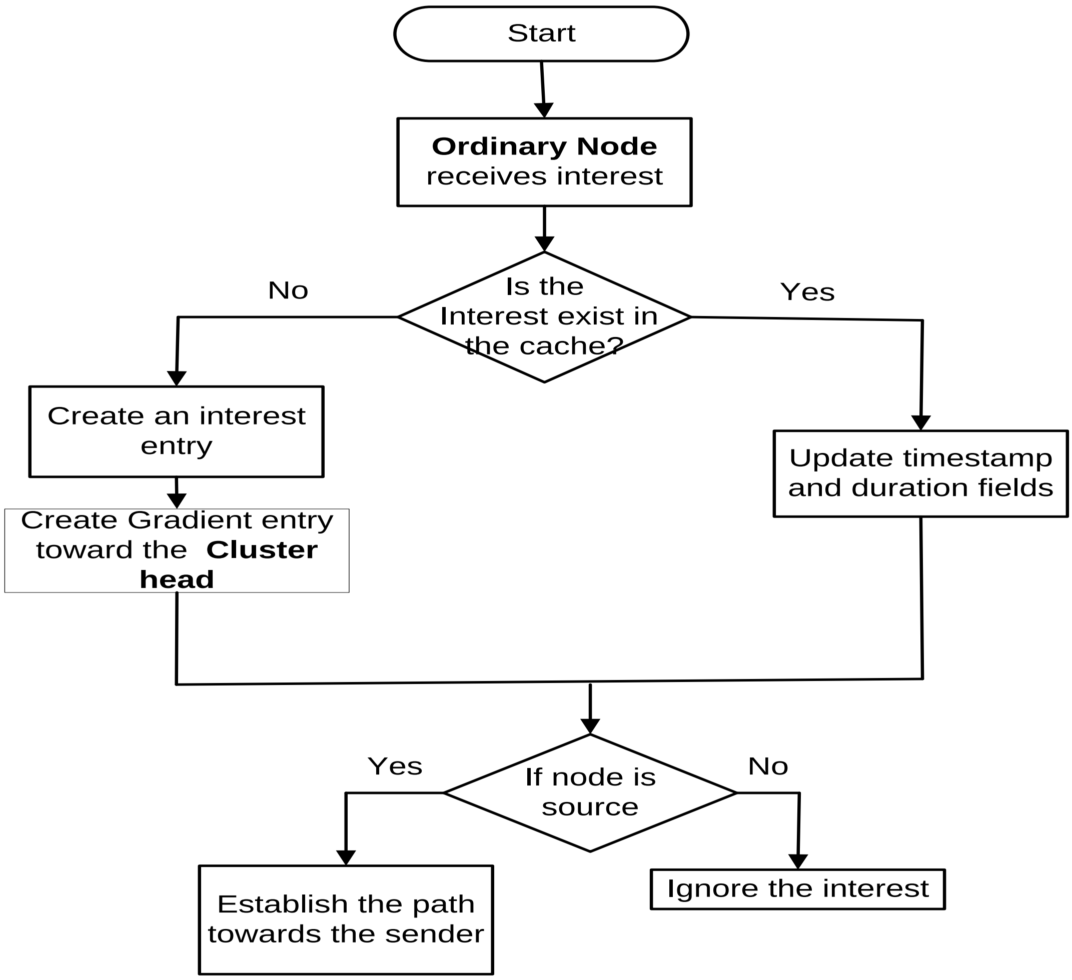


Figure 3.5: Ordinary node receives the interest from its cluster head

Periodically from the sink node the same interest message is re-send, by adding a timestamp attribute, to the cluster heads, to assure the robustness of the network.

3.6 Data Transmission Phase

The *data* is the collected information by the sensor nodes which matches an interest or request of a user and named by using attribute-value pairs. As it is stated in the original DD, the thesis does not present the algorithms which describe the recognitions of target in details. Instead, in this algorithm the nodes match between the sampled waveforms against a library of pre-sampled, which is stored waveforms. This is based on the observation that the acoustic or seismic footprint of a wheeled vehicle varies from that of human being, for example. During the matching process, in the case of some differences between the sampled waveforms and the stored waveform, the protocol gives some degree of confidence[49].

Once the required data is detected, the entries in the interest cache at each of the nodes are utilized to route the data back to the sink. For example, a sensor node that detects an animal might generate the data below[24, 49]:

type = four-legged animal // detect animal's location
instance = elephant // instance of this type
location = [125, 220] // estimated location
intensity = 0.6 // signal amplitude measure
confidence = 0.85 // confidence in the estimate
timestamp = 01:20:40 // event generate time

3.6.1 From Ordinary Nodes to Cluster Heads

If the ordinary node detects the data, it searches in its interest cache for the matching interest entry. If matching interest is not found, it discards the data. If it finds matching interest, it checks the data cache. In case that a match data exist in the data cache it discards the data as it is already forwarded to the cluster head. Otherwise, the node sends it to the respective cluster head and adds to its data cache entry.

3.6.2 From Cluster Heads to Sink

At the cluster heads, if it detects the target or receives the data from the ordinary nodes, it checks its interest cache for the matching interest entry. If it is not found, it discards the data. Otherwise, it checks its data cache entry if the data is already forwarded to the neighbor cluster head or to the sink. In case the data is not forwarded to the neighbor cluster head or the sink, it adds the data to the data cache and forwards to the neighbor cluster head or the sink based on the location of the current cluster head. At the cluster heads data aggregation can be performed to reduce the amount of data transmitted to the sink and the energy consumption. The example is shown in Figure 3.6.

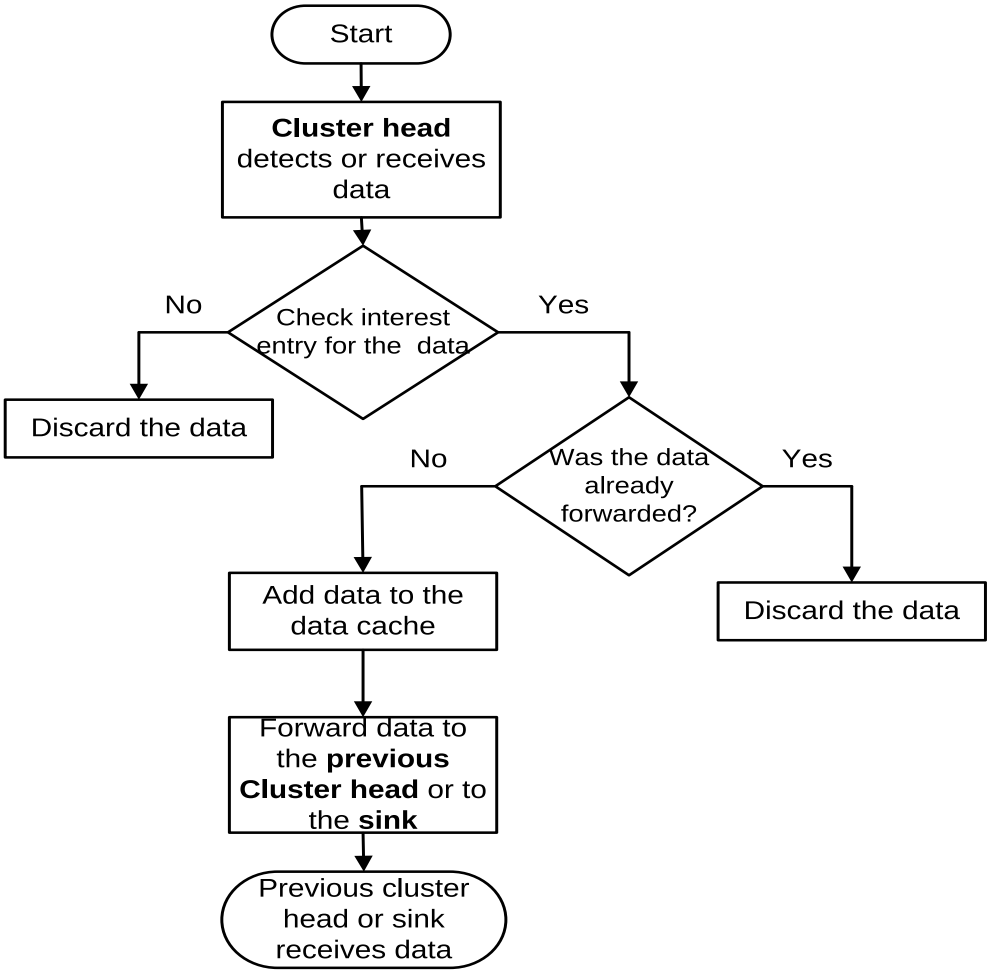


Figure 3.6: Cluster head detects or receives data

The sample of message sequence which has been exchanged among the sink node, cluster heads and ordinary nodes is shown in Figure 3.7. Before the sink node propagates the interest packet, the election of cluster heads takes place. Followed by, the advertisement message from the cluster heads and join-request message from the non-cluster heads nodes are happened. After the cluster is setup and the sink knows the cluster heads, the sink unicastly forwards the interest packet only to the cluster heads. The ordinary nodes receives the interest packet from their respective cluster heads and, if match data obtained using the same path in reverse back for the interest packet the data will be transmitted towards the sink node along multi-hop cluster heads.

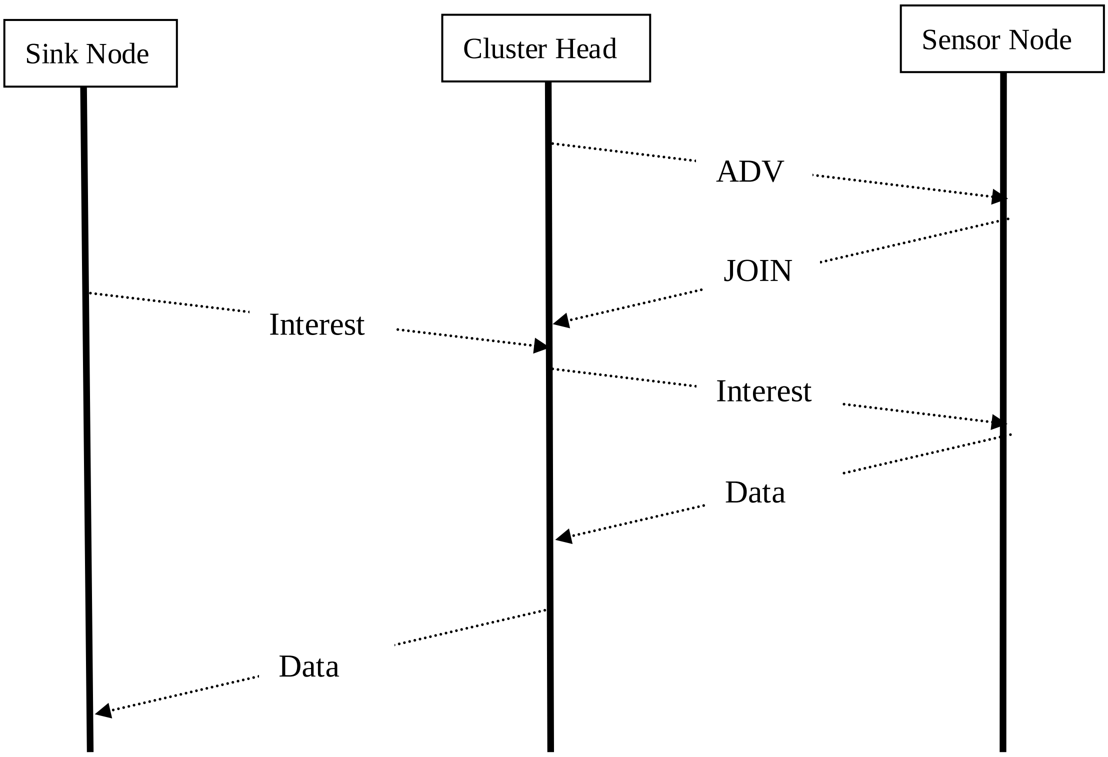


Figure 3.7: Message sequences between sink, cluster head and sensor nodes

3.7 Cluster Maintenance Phase

If the energy dissipation, during clustering and cluster maintenance, is reduced, the more energy can be used to achieve long lifetime for the sensor network. To ensure uniform energy dissipation of the sensor nodes, it is very important to rotate the cluster heads among sensor nodes as the cluster head's duties require higher energy consumption. In the proposed method as in LEACH routing protocol after the initial formation of the clusters, to ensure the good functionality for the WSNs, maintenance function after a period of time by electing new cluster heads which are not participated in the last round can be performed. Rotation of cluster heads guarantee evenly distributed of energy dissipation among the sensor nodes and increase the network lifetime.

3.8 Summary

The proposed method-cluster-based data diffusion routing protocol for large-scale wireless sensor network was discussed. In this method, cluster head is determined based on a pre-calculated eligibility threshold value. The cluster head will advertise itself to the rest of the network. Each sensor nodes measure the advertisement signal strength and decide for connectivity. In addition, the cluster head will also receive an interest message from the sink node which then disseminated to the member of the cluster. By utilizing the route which is established between the members of the cluster head and the cluster heads themselves, how the data retrieved from the source nodes, is explained. Finally, is found that rotating the cluster head role among the sensor nodes from time to time helps in increasing WSNs lifetime by reducing the burden over the cluster heads.

CHAPTER 4

SIMULATION RESULTS FOR STATIC NODES

This chapter describes the simulation models to observe and evaluate the results and, the elements which comprise the simulator. We first describe the components of the simulator. The second part of the chapter provides the metrics: average dissipated energy, packet delivery latency and packet deliver ratio, and the simulation results as well. These metrics are utilized to investigate the proposed protocol by using three different methodologies: fixed density, fixed area and different number of sources. The simulations to evaluate the performance of the proposed protocol have been performed using J-Sim[52-58].

4.1 J-Sim Network Simulator

J-Sim is an open source that is developed entirely in Java and is component based compositional network simulation environment. The J-Sim framework has been implemented on top of component-based software architecture, called *autonomous component architecture* (ACA). On the top of ACA, a generalized packet-based network simulation framework, called Extensible Internetworking Framework (INET), has been laid based on common features extracted from the various layer of the protocol stack[55].

The main objective of WSNs is to monitor and sense events of interest in particular environment. When the *sensor nodes* detect an event of interest (e.g. change in the acoustic sound, seismic or temperature) that is generated by the *target nodes*, send reports to *sink (user) nodes* (either periodically or on demand)[55].

From the perspective of network simulation, a WSN typically consists of three types of nodes: sensor nodes (that sense and detect the events of interest), target nodes (that generates events of interest) and sink node (that utilize and consume the sensors information). Figure 4.1 shows the model of typical WSNs in which the target nodes only can generate the events and communicate with the sensor nodes through the sensor channel. The sensor nodes have the features of receiving the events from the target nodes through sensor channel and receiving and transmitting using wireless channel with other sensor nodes or sink. The sink node also has the characteristics of receiving and transmitting through wireless channel[52].

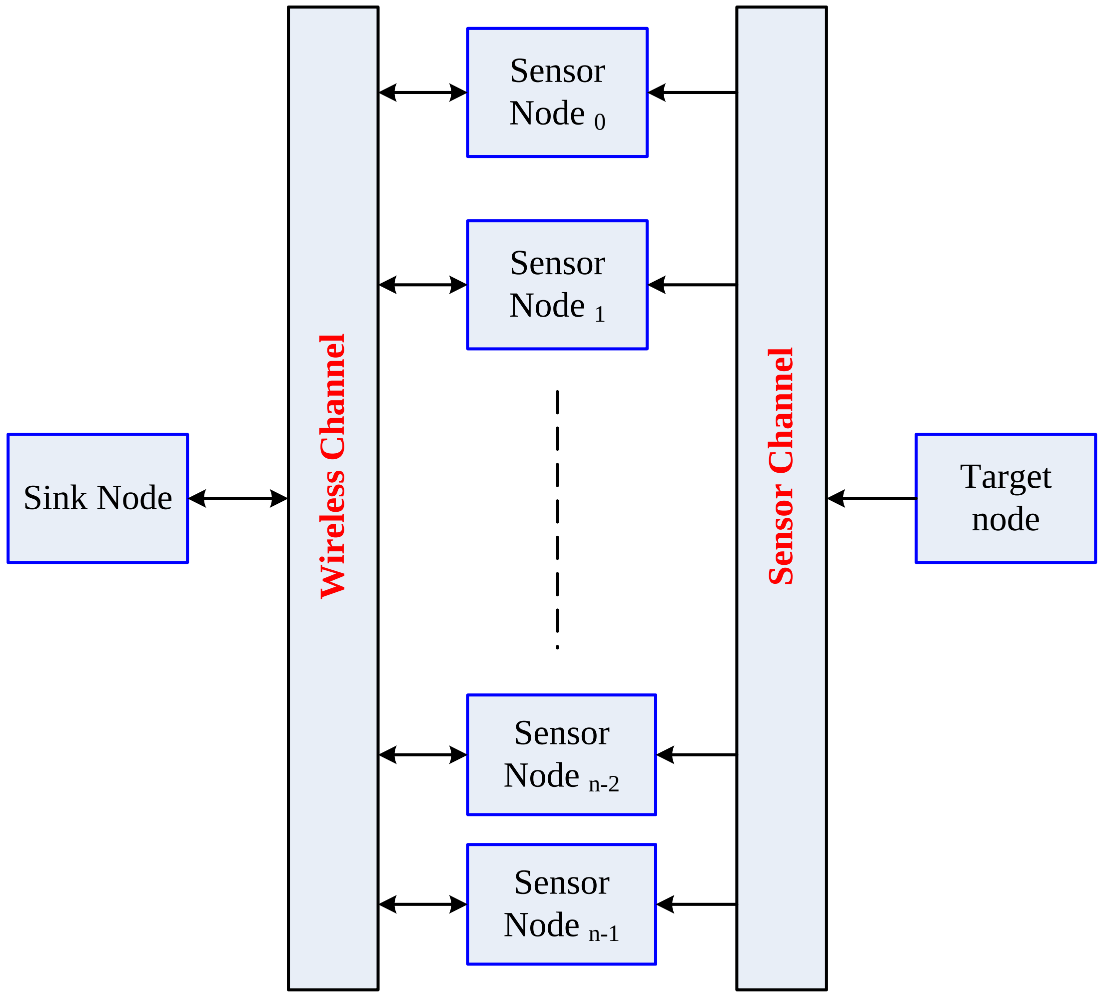


Figure 4.1: The model of a typical WSNs environment[52]

4.1.1 Characteristics of J-Sim simulator:

There are many desirable features for J-Sim simulator when comparing with other network simulators. Some of the features are [53, 55, 57]:

- Unlike ns-2[2], classes/methods/fields in Java need not be explicitly exported in order to be accessed in the Tcl environment. Instead, all the public classes/methods/ fields in Java can be accessed (naturally) in the Tcl environment.
- The combination of developing J-Sim entirely Java with the autonomous component architecture, makes J-Sim a truly platform-neutral, extensible and reusable environment.
- J-Sim allows integration with different script languages like Perl, Tcl, or Python by providing script interface.
- It is a dual-language simulation environment in which classes are written in Java and “glued” together using Tcl/java.
- The J-Sim simulator achieves better scalability in terms of the experiment setup time and the simulation completion time than the well known network simulator ns-2 and Scalable Simulation Framework Network Models (SSFNET) [55, 57].

4.1.2 Components of J-Sim Simulator Framework

This part presents the component of the J-Sim simulator in general. Specifically the main components which forms the WSNs, target node, sensor node and sink node.

4.1.2.1 Target Node

As shown in the Figure 4.2 in order to realize the target node several classes have been implemented in J-Sim. One of the classes of the target node is *TargetAgent* class which implements the target agent layer. *TargetAgent* periodically generates the stimuli (signals) and passes them to the lower layer to be propagated over the *SensorChannel*. The second main component of a target node is *SensorPhy* class which implements the sensor physical layer. The functionality of SensorPhy is different whether it exists in the protocol stack of a target node or a sensor node. If it exists at the target node, it receives the signals generated by TargetAgent, query *SensorMobilityModel* to get the up-to-date location of the target node, and transmit

[2]Ns-2 is a Network Simulator, which uses dual-language; C++ and Tcl, it can found at ,http://www.isi.edu/nsnam/ns/.

the signals including the location information to the sensor channel component (SensorChannel). As shown in Figure 4.2 the target node can only send (but not receive) data packets over the sensor channel[55].

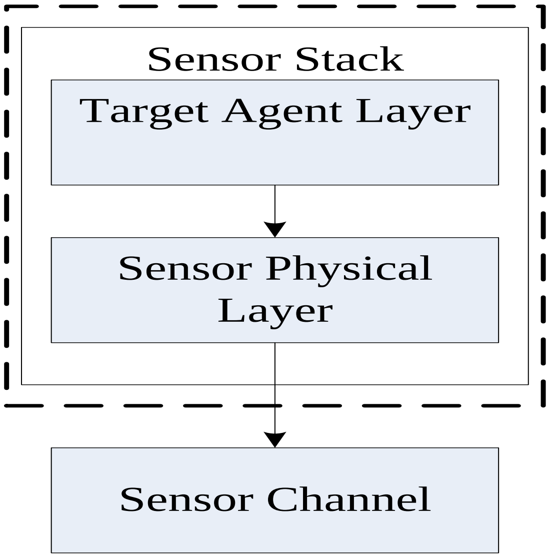


Figure 4.2: Internal view of a target node (dashed line)[52]

4.1.2.2 Sensor Node

The type of signal propagation between target node and sensor node is different from the signal propagation signal between sensor node and sink node. Therefore, two different propagation models are included: *sensor propagation model* and *wireless propagation model*. The sensor node is equipped by:

- *Sensor protocol stack* which enables the sensor node to detect the signals generated by targets over the *sensor channel*
- *Wireless protocol stack*, which enables the sensor node to deliver the data sensed from target nodes to the sink node or other sensor nodes over the *wireless channel*

In order to realize the sensor node, Figure 4.3, several classes have been implemented in J-Sim[55].

1. *Sensor Protocol Stack* which includes the following classes:

   a) SensorPhy which implements the sensor physical layer and its role at the protocol stack of a sensor node is to receive, the signals generated by the target nodes, over the sensor channel

b) *SensorAgent* the class which implements the sensor layer and its role is to receive the signals from the lower layer (*SensorPhy*) and forwards it to the upper layer (sensor application layer)

2. *Sensor Application and Transport Layers* which include[55]:
   a) *SensorApp* this class implements the sensor application layer. At this class the application-specific data packet from SensorAgent class is received. From the SensorApp the processed data is passed down to the transport layer. The processed data then goes through the wireless protocol stack and finally be transmitted to the sink node over the wireless channel.
   b) *SensorPacket* in this class the data packet that will be transmitted over the wireless channel is implemented. SensorPacket is either to specific destination (unicast) or to all (broadcast).
   c) *WirelessAgent* this class implements the transport layer between the sensor application layer and wireless protocol stack. WirelessAgent receives the data packet from the upper layer, sensor application layer, encloses it in a SensorPacket and passes it down to the wireless protocol stack to be eventually transmitted over the sensor channel to the sink node or other sensor nodes.

3. *Wireless Protocol Stack* is built in plug-and-play fashion using the J-Sim classes that constitute the J-Sim wireless network extension.
   a) *PktDispatcher* provides the role of the IP layer in the real world, sending/delivery of the data to upper and lower layer protocols. It forwards the incoming packets to appropriate set of output ports connected either to an upper layer protocol or lower layer component
   b) *ARP* it implements the address resolution protocol (ARP)
   c) *LL* implements the link layer functions when it receives the unicast IP packets from PktDispatcher, and queries ARP (by doing ARP resolve) to find out the MAC address of the next hop to which the IP packet should be forwarded and it inserts it in the interface queue of the underlying wireless interface card . Outgoing IP and ARP packets are buffered in the *Queue* component.
   d) *Mac 802.11* which implements the IEEE 802.11 MAC protocol
   e) *WirelessPhy* it implements the roles of the physical layer of a wireless card

and it queries *WirelessPropagationModel* to determine the received signal power and delivers a data only if the received signal power is at least equal to a certain receiving threshold

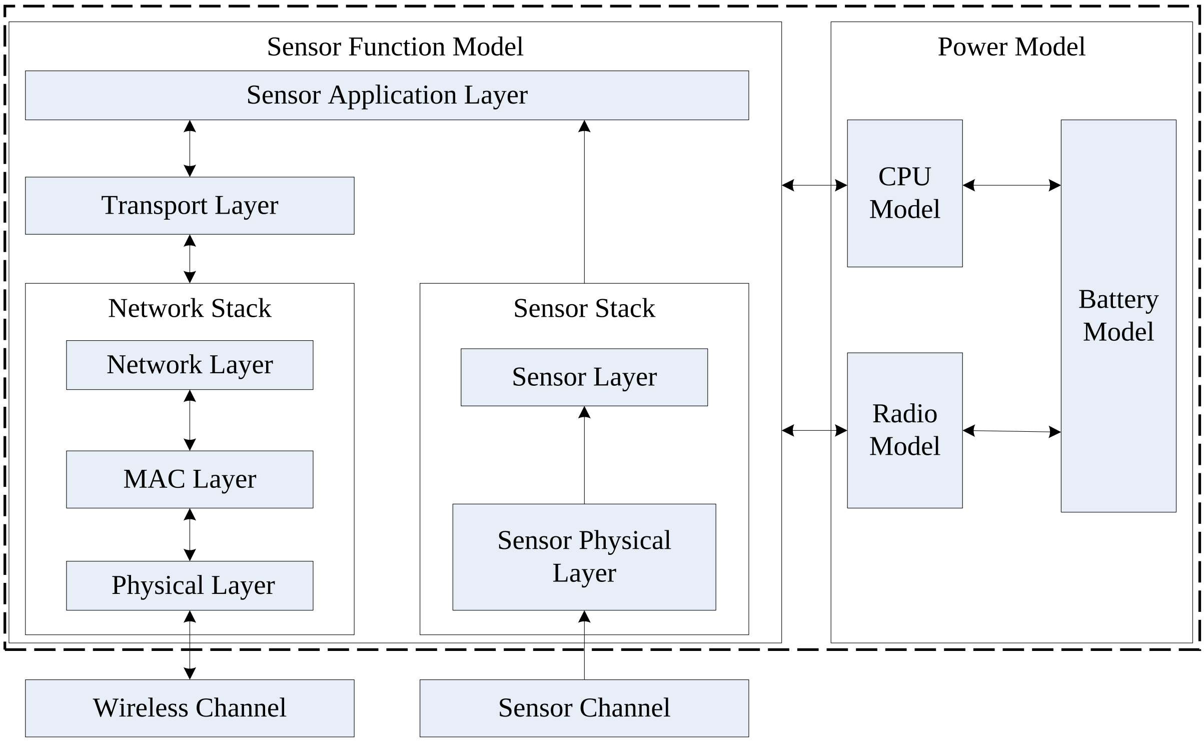


Figure 4.3: Internal view of sensor node (dashed line)[52].

As the sink might be placed at the far area from the node, the propagation of the data over the *wireless channel* is usually performed in *multihop* fashion. Due to this fact the intermediate node between the source and the sink should serve as relays (routers). This indicates that sensor nodes have to be able to both send and receive data over the *wireless channel*.

4.1.2.3 Sink Node

The sink may need to send query/interest to some geographical area of the sensor nodes and it can receive what have been sent from the sensor nodes over wireless channel. Therefore, as shown in the Figure 4.4, the sink node able to send and receive over wireless channel.

A sink node, Figure 4.4, can also be constructed in a plug-and-play fashion using sensor application layer (SensorApp), a transport layer (WirelessAgent), a physical layer (WirelessPhy) and a wireless protocol stack in general as explained in a sensor node description[55].

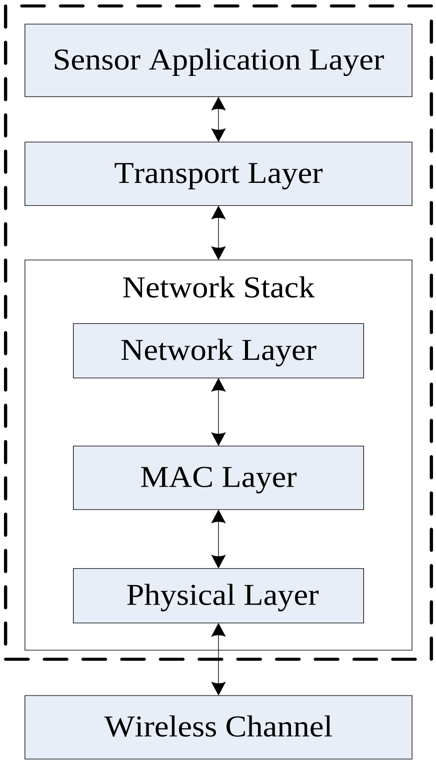


Figure 4.4: Internal view of sink node (dashed line)[52].

4.1.2.4 Sensor Channel

In order to realize a *sensor channel*, the common for the target and the senor node, the classes implemented in J-Sim are as follows: *SensorNodePositionTracker* this class is used to determine the sensor nodes, reported their location by the *SensorMobilityModel* component, are within the sensing radius area of a target node to receive the signals generated by the target nodes. The other class has been implemented is the *SensorChannel* which implements the sensor channel and its role is receiving the signal from the target node, queries *SensorNodePositionTracker* class to get the list of nodes within the sensing radius of the target node and sends the generated signal to each sensor node which included on the list. It repeats the same after a certain propagation delay.

4.1.3 Operation of the Simulator

The signals are generated periodically by the target nodes and propagated over the sensor channel. Each sensor node within the radius range of the target node receives the stimuli over the sensor channel and further process like comparing the received power with a pre-determined threshold value will be done. Sensor propagation mode is used to determine the received signal power used in the model (e.g., seismic or acoustic). The sensor node which receives the data over the sensor channel from the target node has to forward to the sink node over the wireless channel directly or through another sensor nodes using multi-hop communications. Based on the application the sensor nodes are used for, a sensor may send as soon as it detects if the used data reporting method is event-driven or time-driven. In the case of query-driven the sensor nodes cache the data which has been detected in the data cache entry and later on compare it with query which will be sent from the sink node.

In the three cases, event-driven, time-driven or query-driven the sensor nodes may do some process on the data (applying aggregation and getting average, minimum or maximum) then transmits it to the closest sensor node or to the sink directly. Any kind of in-network processing methods can be used in the sensor application layer. the sensor application and transport layer is used as the coordination between the sensor protocol stack and the wireless protocol stack.

In our proposed CDD method as it is query-driven, if the sensor node receives the signal it cache in the data cache entry for further use of it. Whenever the cluster head sends the interest message and the sensor node receives it, it compares with the query that has been sent from the sink node. If the node match data to the query is found with the recently sent interest message it send the data to its cluster head and the cluster head make further processing (applying aggregation approach)before it will be eventually sent to the sink node over wireless channel using multi-hop communications.

The snapshots have been taken to explain how the simulator works from the beginning of the simulation until the simulation stops. Figure 4.5 shows the snapshot of starting the simulation and displaying the TCL window. The following steps, which are described in Figures 4.6-4.7, during the creation of the sink node, sensor nodes and target nodes and assigning random locations for the sensor nodes are presented. In addition Figure 4.8 the message used when the interest or the data have been exchanged between sink, sensor and target nodes.

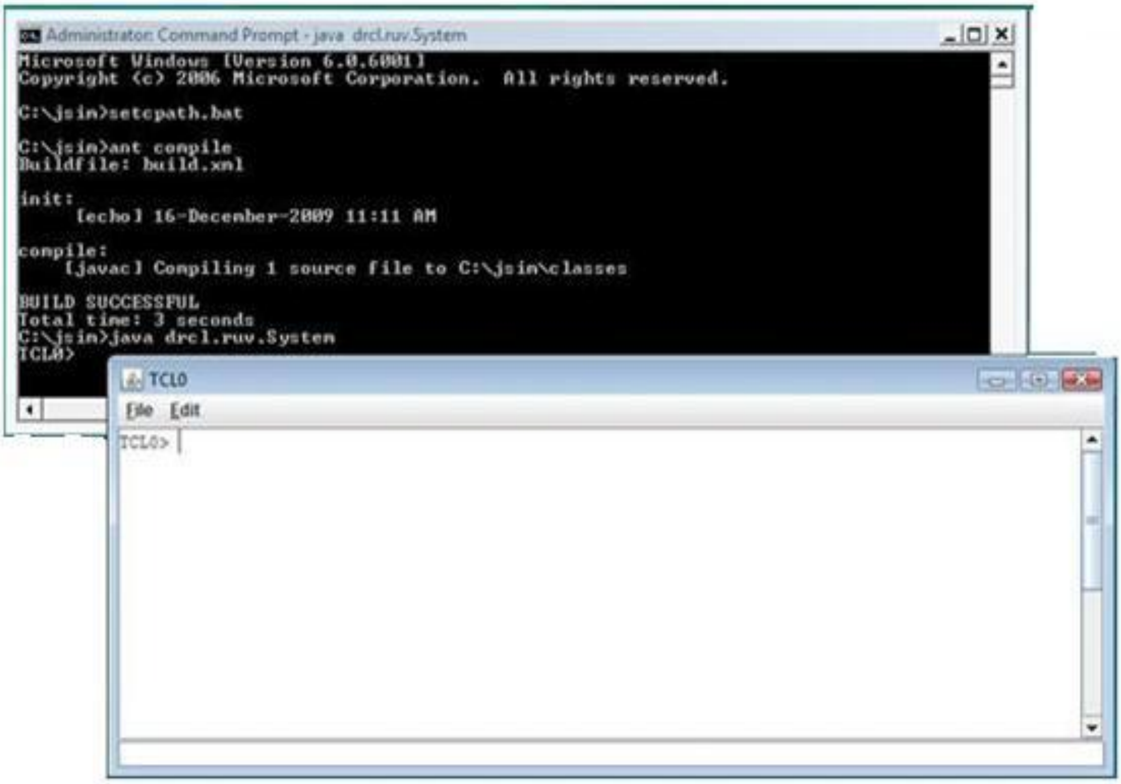


Figure 4.5: Snapshot for starting the simulation

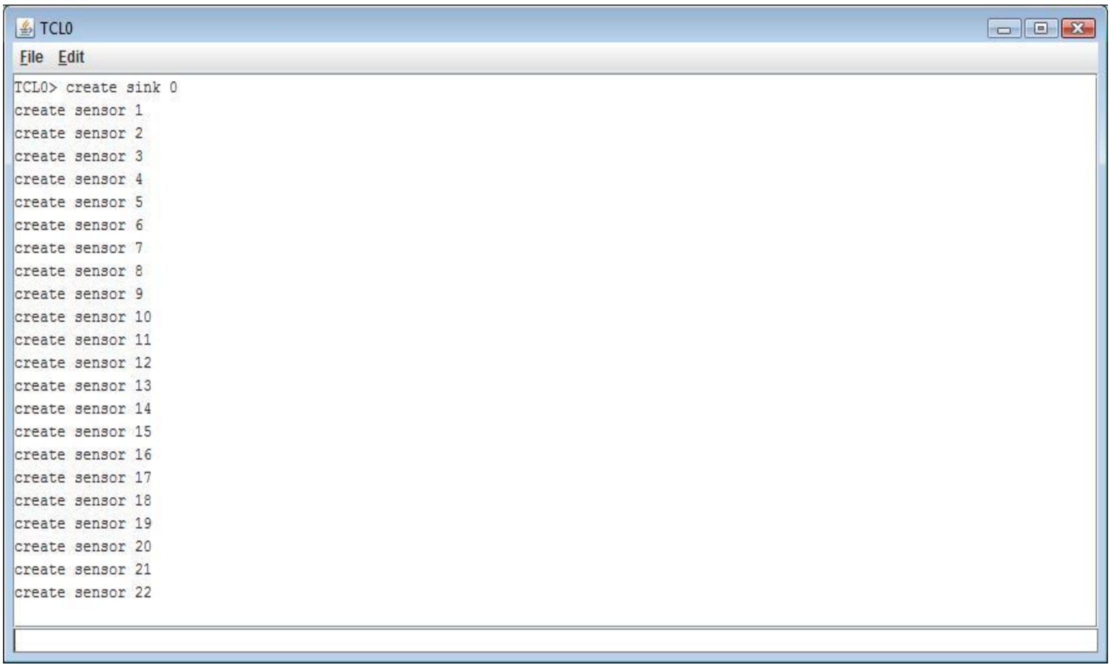


Figure 4.6: Snapshot during the creation of sink node,sensor nodes and target nodes

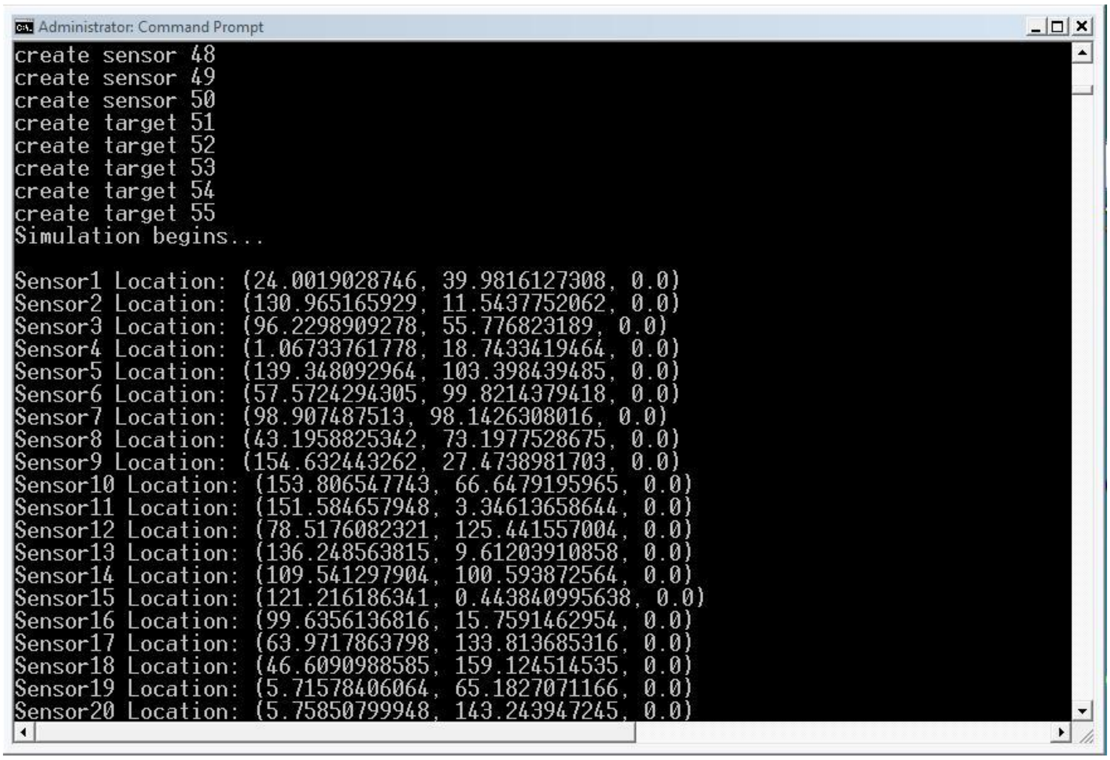


Figure 4.7: Snapshot while the sensor nodes are randomly deployed and getting random location

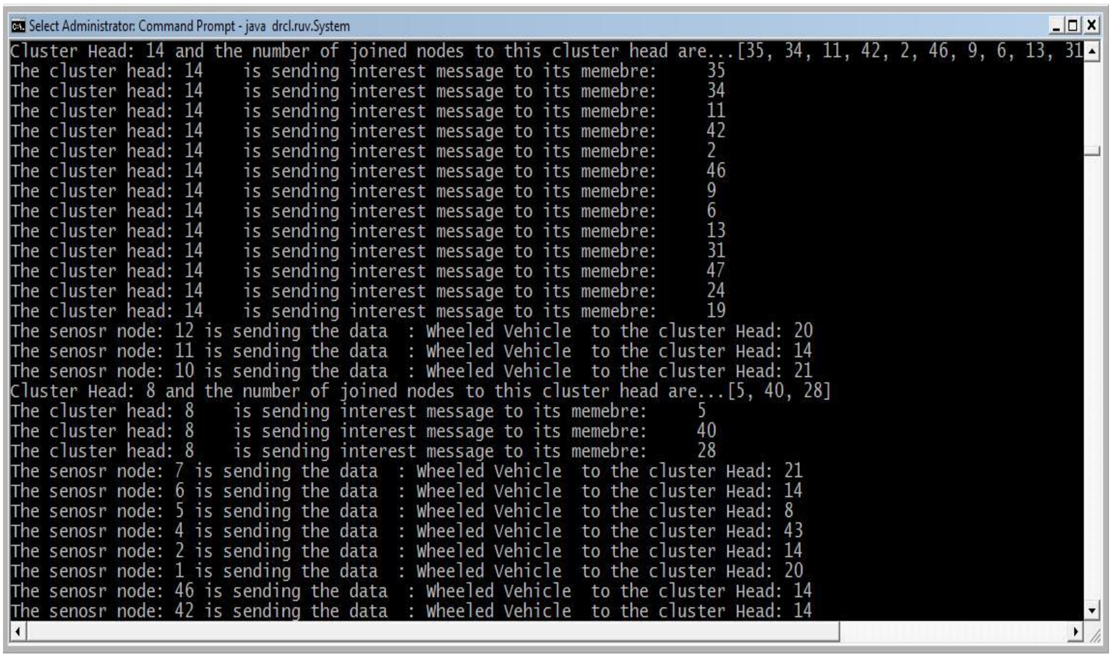


Figure 4.8: Snapshot shows some messages when running the simulation

Figures 4.9-4.11 describes the snapshots of remaining energy, average latency and packet delivery ratio respectively. The snapshots are taken when 200 sensor nodes are deployed in 200x200 m square in fixed area scenario to show as example how the simulator displays the remaining energy, latency between the source and the sink node and packet delivery to the sink. In particular case Figure 4.11 displays two types of packets. The blue color shows the total number of packets that have been sent from all nodes in the networks but the red color shows the packet that have been received by sink (base station) through aggregated multi-hop cluster heads.

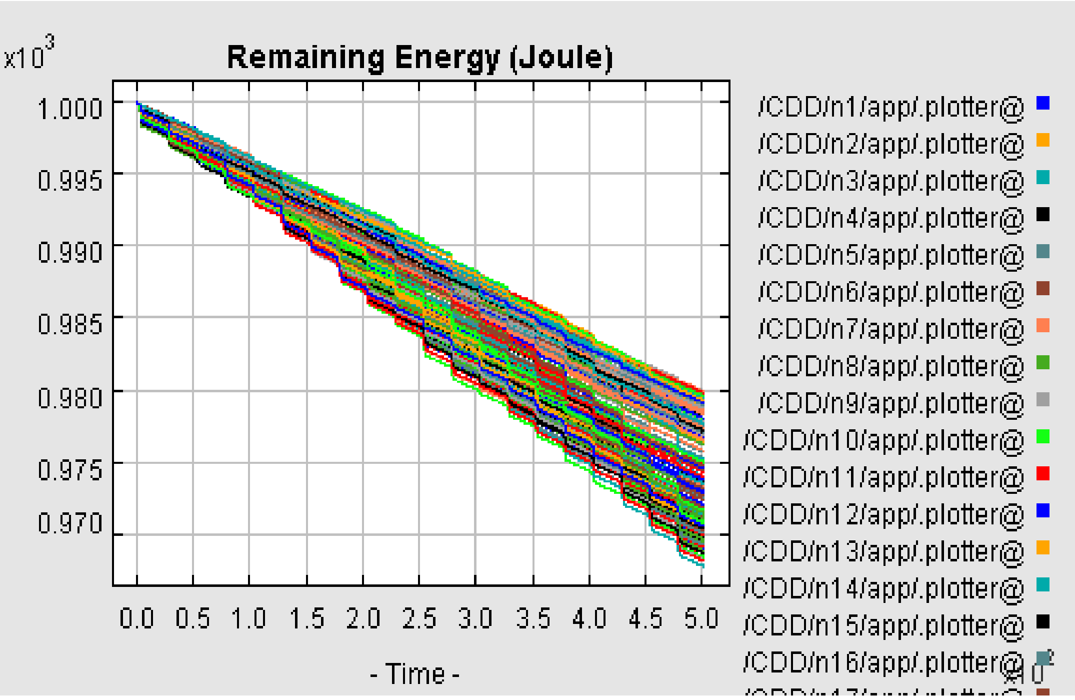


Figure 4.9: Snapshot showing the remaining energy while running 200 nodes-fixed area scenario-for 500 sec simulation time

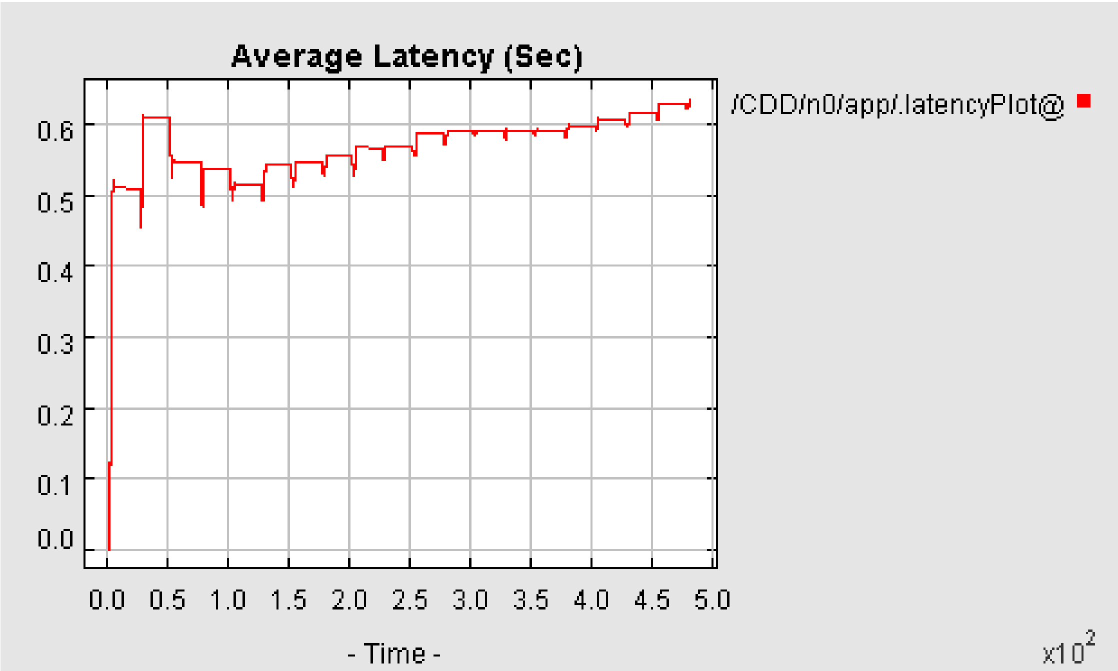


Figure 4.10: Snapshot showing the average delay while running 200 nodes-fixed area scenario-for 500 sec simulation time

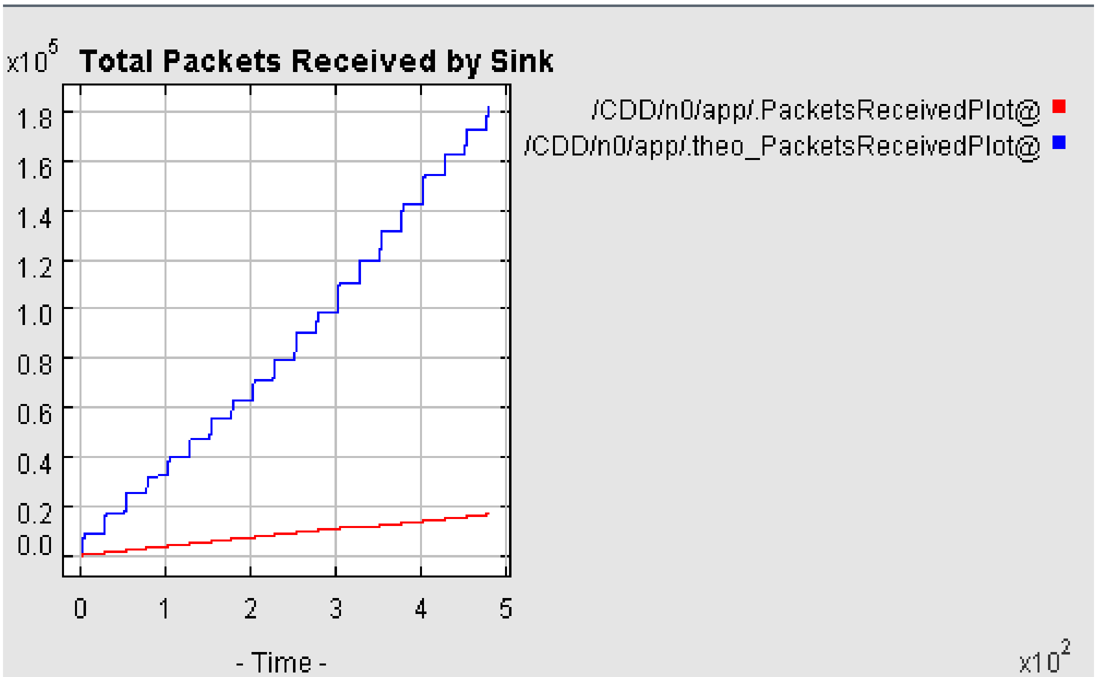


Figure 4.11: Snapshot showing the packet delivered while running 200 nodes-fixed area scenario-for 500 sec simulation time

4.2 Metrics

The main purpose of conducting the performance evaluation study is to interpret the impact of network size, density of the nodes and changes in number of sources on the proposed method while comparing it to the original directed diffusion protocol. In order to evaluate the performance of CDD protocol three metrics have been chosen. Similar metrics have been used in[49] to compare the performance of the original DD protocol with two other routing protocols.

- **Average Dissipated Energy:**

The average dissipated energy is used to calculate the ratio of the initial energy and the final energy level for each node. It is very important metric as the overall lifetime of sensor nodes based on it. The lower the energy consumption of the sensor nodes, the longer the lifetime of the sensors can be achieved, because once the sensor nodes are deployed they should live for long time before their battery is depleted, especially if the sensor nodes are distributed in remote areas. The dissipated energy can be calculated using the following equation:

$$\sum_{i=1}^{N} \frac{IE_i - RE_i}{NxT} \qquad (2)$$

Where:

N≡ total number of sensor nodes

T≡ total number of tasks

IE_i≡ initial energy of node i

RE_i≡ remaining energy of node I after the completion of the simulation

- **Average Delay:**

It measures the time taken to deliver data from the originator (source) to the receiver (sink). It's based on the applications the sensors are used for. Real-time applications need low latencies and others need data completeness regardless of the delay happens. The average delay can be calculated using the following formula:

$$\sum_{i=0}^{n} \frac{\text{Time packet received}_i - \text{Time packet sent}_i}{\text{Total number of packets received}} \qquad (3)$$

Where:

n is number nodes

- **Average Packet Delivery Ratio:**

It is the ratio of between the number of data packets arrived to the sink to the data packets sent by the sources. This metric indicates how the routing protocols either the original or the proposed, are successfully delivering the data packet to the application layer. A high value of delivery means the goodness of the protocol in delivering the interested data from the source node to the sink node.

As the nature of the sensor nodes are small and inexpensive, based on the application required they can be deployed densely. Due to the short distances utilized for commutation among the sensor nodes, diffraction is not a major factor in WSNs, which is low-power and short-range communications[11]. Moreover, a high signal to noise ratio (SNR) can be obtained since the distance between the sensors is very short [24, 49].

As it is stated in the original DD,to simplify the results obtained, because of the sensor networks are operated in non-congested system which is far from the overload, the sensors we have conducted in our simulation are do not experience congestion. However, as the authors of DD declared, there are some approaches to deal with the congestion like (in-network data rate down conversion and aggressive data quality reduction through aggregation)[24, 49], but these methods are beyond the scope of our study.

4.3 Methodology and Simulation Results

This section explains our experimental methodology which includes generating the sensor nodes, selecting the radio range and the deployment process of the nodes over the simulation area. It also discusses the simulation results obtained for variant scenarios using the metrics mentioned in section 4.2. First, the section starts by analyzing the effects of network size on the performance of the proposed protocol compare to the original one. The following part of the section verifies the impact of the network density over the fixed area on the both of the protocols. Beside the effect of increasing the source nodes on the routing protocol is discussed which constitute the third part of the section. The last part presents the table contains comparison of CDD protocol with existing DD routing protocols.

4.3.1 Performance Analysis with Fixed Density

To study the performance of CDD as the function the network size which helps for testing the scalability of the protocol, different networks have been generated. The generated networks are ranged from 50 to 250 increasing by 50 nodes. In the first scenario, the randomly generated 50 nodes are placed in 160m by 160m square where the radio range of the node is 40m. The rest the nodes have been generated as the average density of the sensor nodes in the network remains constant. The simulation scenario is organized from one sink and five sources. The sink placed at the top right corner of the simulation area, where the sources deployed in 70m by 70m square of the bottom left side of the simulation field.

The most challenge for WSNs is designing an energy-efficient routing protocol of which helps in decreasing the energy consumption of the sensor nodes individually or in group that will gains in prolonging the network lifetime. As sensor nodes are energy constrained, once they are deployed the battery is irreplaceable because of drastic environments in which the sensors are distributed. This makes energy the most important part of the senor, since the amount of their energy is very limited. By decreasing the energy consumption of the sensor nodes the longevity of the network lifetime can be achieved.

In directed diffusion protocol when the sink node sends an interest message to get answer from the sensor nodes in some geographical area, it diffuses the interest to whole network which consumes a huge amount of energy which reduces the lifetime of the sensor nodes. The evaluation of the energy consumption is achieved based on the metric Joules/Node/Distinct event. As Figure 4.12 shows when the number of nodes increases in the network the energy consumption for the routing protocols and for the metric used is almost constant. This is because the average density of the nodes in the network is fixed. However, one can clearly observers that there is a great differences of energy consumption for the proposed routing protocol when it is compared to the original Directed Diffusion and PCDD protocols. The proposed protocol gives better energy saving than an original one and PCDD, because when the sink interests in finding out some information from some geographical area, it doesn't disseminate the interest message to the whole network. Instead, the interest message is forwarded unicastly to the cluster leaders then it reaches the ordinary nodes through their leaders. In addition, whenever the sensor nodes have the answer for the interest recently sent, they will reply to their cluster heads at which the aggregation of the data can be performed, before it will be sent finally to the sink node. Using the cluster heads, for transmitting the interest and retrieving the data back from the source nodes, reduce the redundancy and the energy consumption. In addition, CDD considers saving of the *idle* energy, by using a TDMA-style MAC, which has been ignored in many of the proposed routing protocols to enhance the original DD.

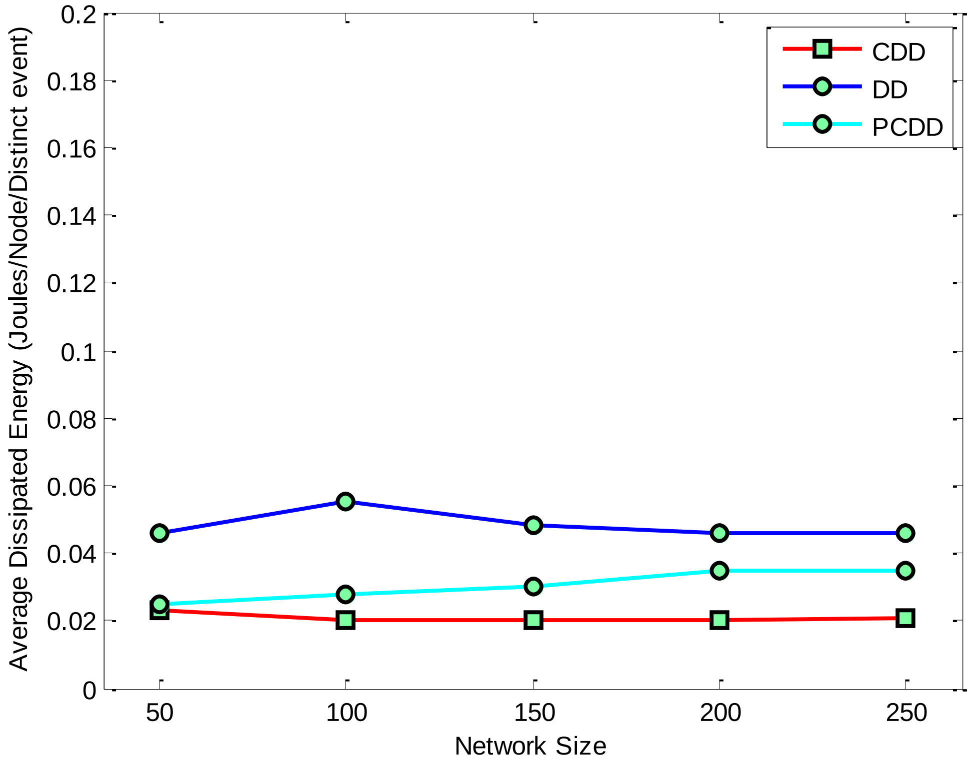


Figure 4.12: Energy Efficiency – fixed density scenario

The average delay for the three protocols has been plotted in Figure 4.13. Using the cluster heads to manage the routing minimize the packet travelling delay. In DD as the number of sensor nodes increase in network the delay also increases. This is because the DD uses retransmissions the messages to build the best route between the source and the sink, while in CDD the route between cluster heads and their members is built as the node elects the best cluster head. Moreover, the cluster head selects the closest cluster head, to its position, as the neighbor towards the sink from the list of cluster heads and, the route is constructed. In this way CDD reduces the delay between the source node and the sink. However, CDD suffers from the delaywhile comparing it because of the cluster heads need some time to aggartegte the data and send it the base station. PCDD shows the lowest of the three routing protocols, but this is at the cost of energy, as the nodes consume the highest energy while comparing to CDD sensor nodes.

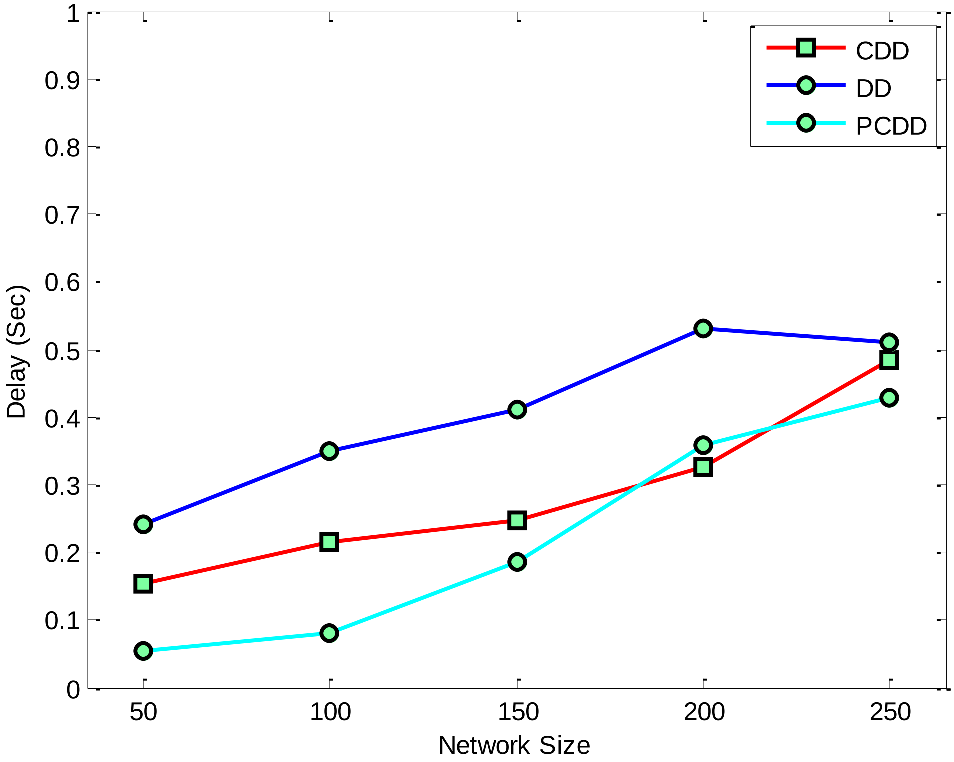


Figure 4.13: Average Delay – fixed density scenario

Figure 4.14 shows the packet delivery ratio of CDD when it is compared with DD protocol. The amount of delivered packets in DD declines as the number of nodes increased in the network. The packet delivery ratio of CDD and PCDD is almost 100% regardless of number of nodes in the network. This is achieved by using the clustering method as the data is organized to be sent from the closest nodes; to the targets, to the corresponding cluster head and finally reaches the sink through multihop cluster leaders.

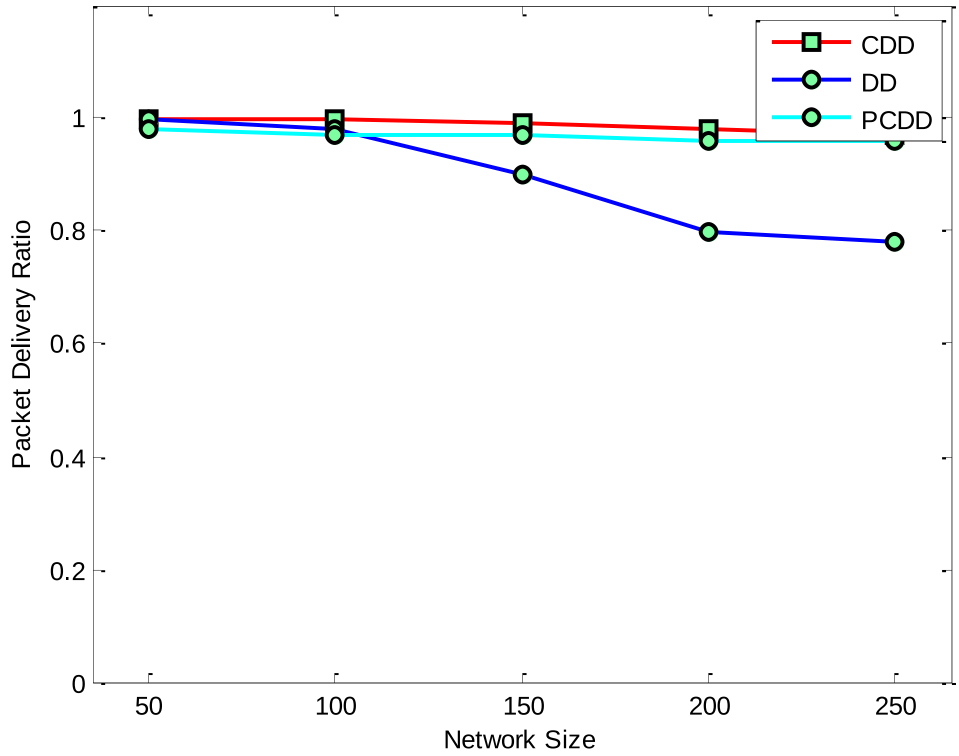


Figure 4.14: Packet Delivery Ratio – fixed density scenario

4.3.2 Performance Analysis with Fixed Area

In this section, the performance of proposed routing protocol is analyzed when the number of sensor nodes varied in a fixed area. Three metrics: average dissipated energy, average delay and packet delivery ratio are used to analyze the performance of CDD with the original DD protocol and PCDD.

Five different sensor fields ranging from 50 to 250 nodes have been generated and deployed in fixed area which is 200m by 200m square. The radio range for each node is 40m and the sensors are assumed to be randomly distributed over the sensor field. In the sensor field there are five sources deployed randomly in the area of 80m by 80m square at bottom left part of the sensor field. The sink is placed at top right part of the simulation field.

The three metrics have been used to evaluate the performance of CDD compared with DD and PCDD routing protocols as the number of nodes increases in a fixed area. The energy consumption of the protocols is plotted in Figure 4.15. For the

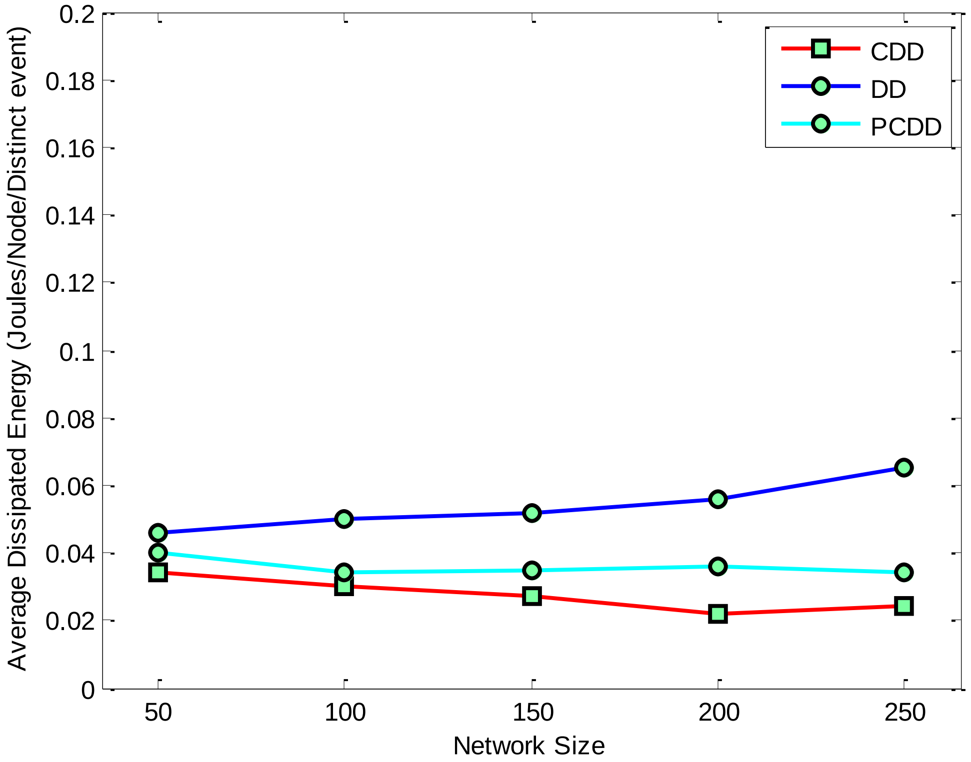


Figure 4.15: Energy Efficiency – fixed area scenario

CDD routing protocol, as the number of nodes keep increases, the average dissipated energy which is calculated based on metric joules/node/received event shows a slight decreasing trend. This is because, the number of nodes that will be in one cluster increases and the distance between the cluster leader and the ordinary nodes is become shorter that reduces the amount of energy which will be used to the send the packet. Also the distances between the cluster heads are reduced that helps to reduce the energy consumption. The packet delivery delay for the denser scenario is represented in Figure 4.16. CDD is giving low latency than DD, but PCDD has the lowest delay as number of nodes increased in the network. Although in hierarchical approaches it needs sometime for the cluster heads to aggregate the data from the members and pass it to the next cluster head or the base station, CDD and PCDD are still showing better than the flat DD routing protocol. For the denser network the DD protocol shows a good performance as well, which appears in a packet delivery ratio, Figure 4.17. It's almost delivering close to 100% for the whole simulations have been conducted in denser network, same as CDD and PCDD.

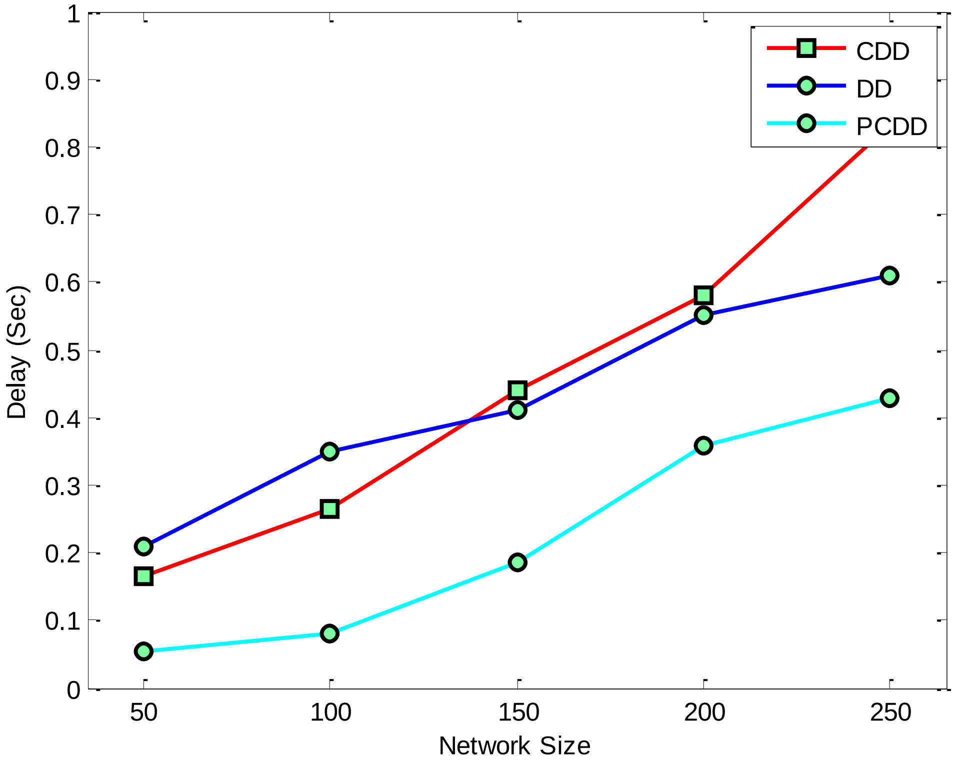


Figure 4.16: Average Delay – fixed area scenario

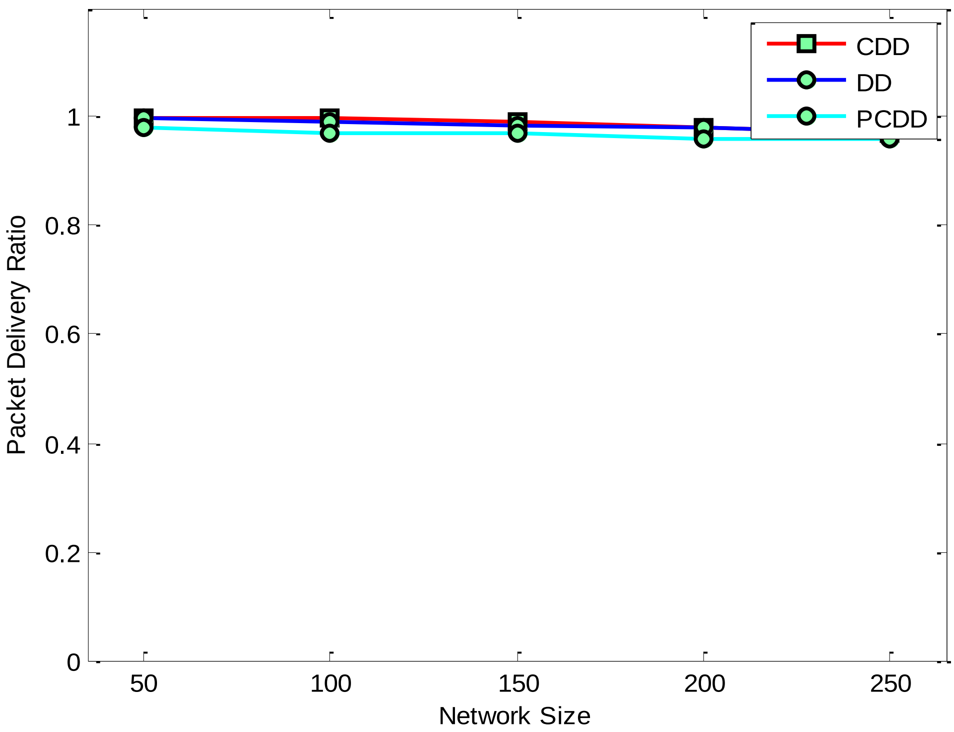


Figure 4.17: Packet Delivery Ratio – fixed area scenario

4.3.3 Impact of Different Number of Sources

In this section 100 nodes are randomly deployed in the 225m by 225m square of the sensor field. This is organized with different number of sources (targets) ranging from 1 to 6 deployed at the left bottom part of the sensor field. The experiments are conducted to analyze the effect of number of sources those who generate the data on the performance of CDD when it is compared to DD and PCDD routing protocol under different metrics.

The number of sources has a huge effect on the energy consumption of the protocols. PCDD, CDD and DD routing protocols energy consumption reduces as the number of sources incremented. It can be obviously seen from Figure 4.18 the energy consumption for DD almost improved by 70% when comparing one source in the network with the six sources. However, the CDD protocol shows considerable improvement over DD and PCDD in the energy consumption as the number of sources is increased in the network.

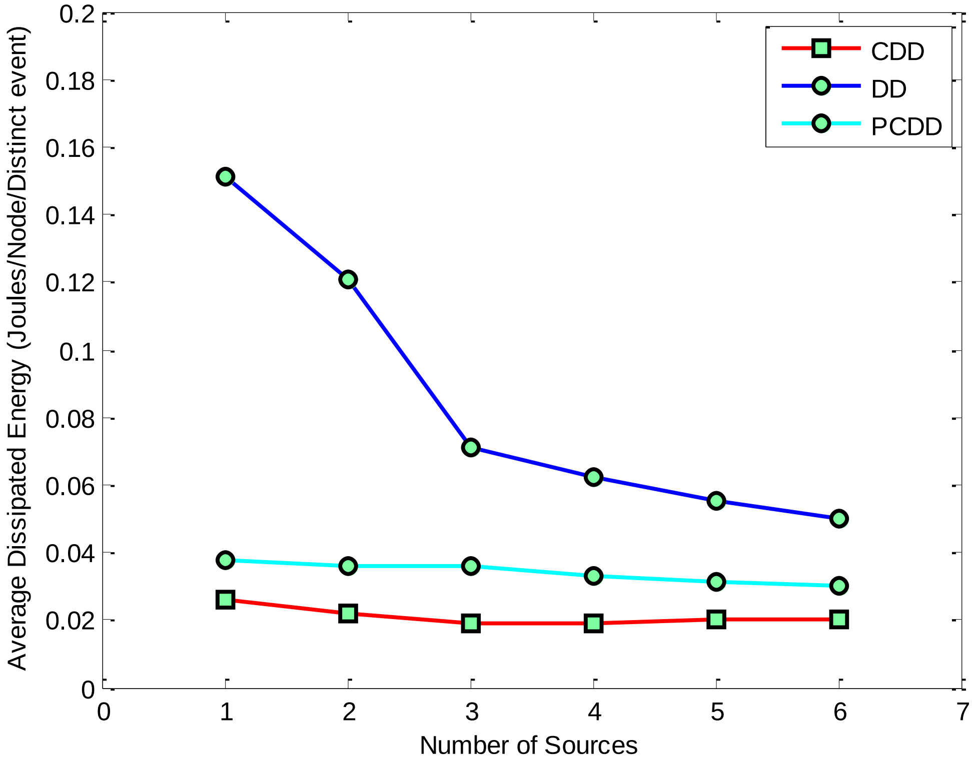


Figure 4.18: Energy Efficiency - impact of number of sources

As observed in Figure 4.19 for CDD routing protocol the average packet delivery delay between the source and the sink is independent of the number of sources. However, in DD protocol, the delay is piece-wise linear, when the number of sources between 1-4, it is almost independent of the sources. However, as the number of sources goes beyond 4, the delay becomes very significant. This is because as the number of sources becomes larger in the network, many transmissions will take place between the sources and the sink to build the best route. For the packet delivery ratio as appeared in Figure 4.20, PCDD, CDD and DD routing protocols show a good performance and they are delivering almost 100% regardless of the number of sources.

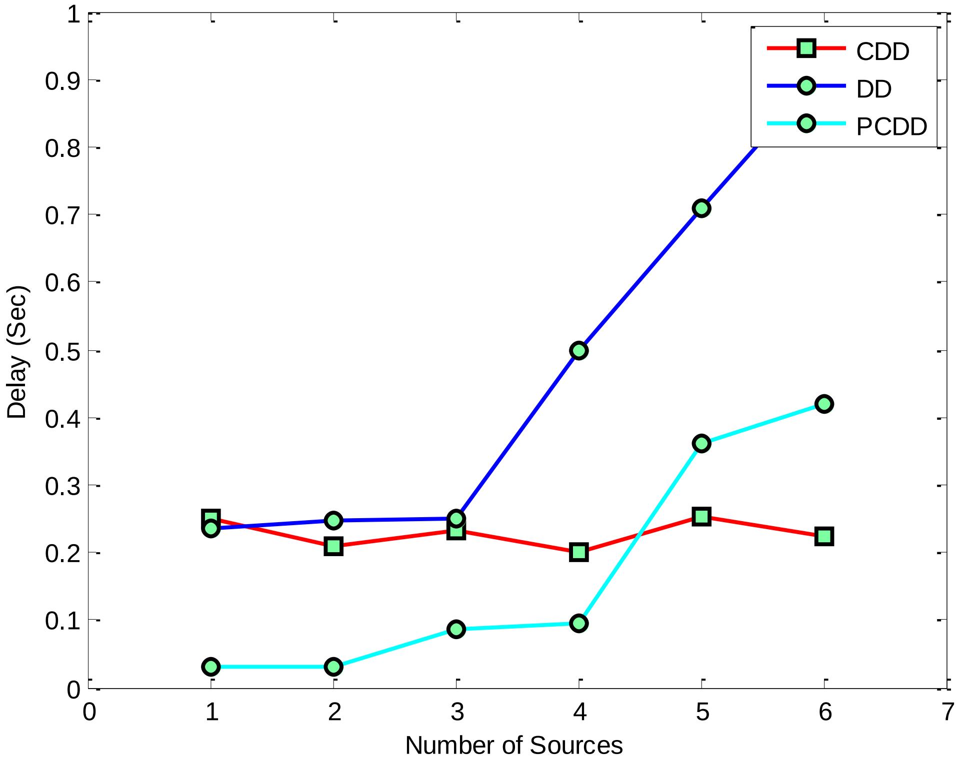


Figure 4.19: Average Delay - impact of number of sources

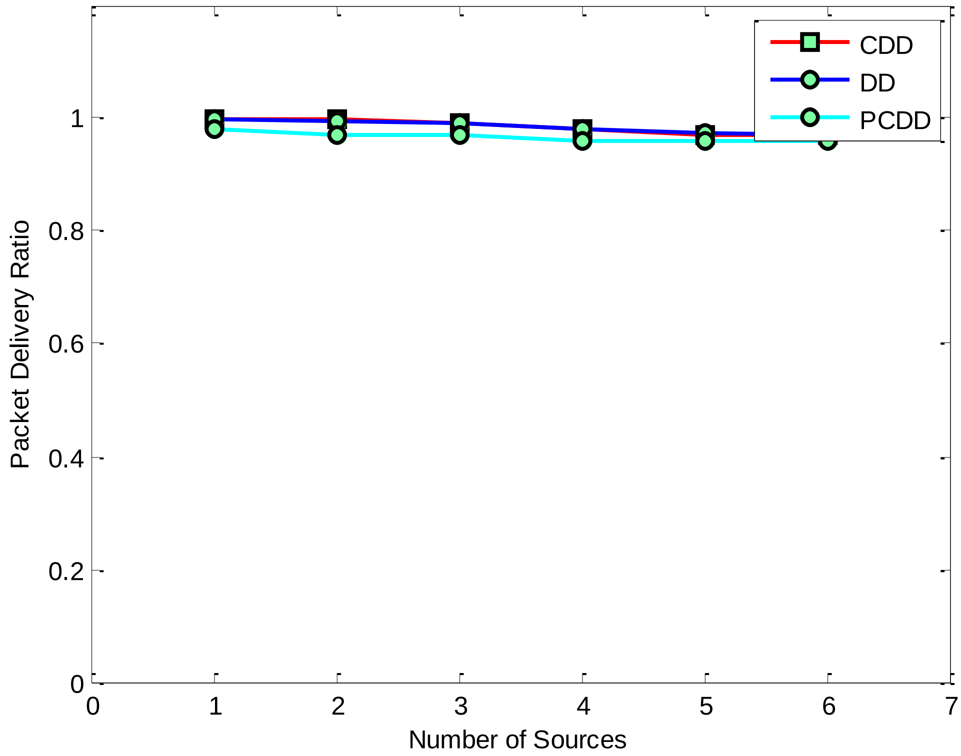


Figure 4.20: Packet Delivery Ratio - impact of number of sources

The improvement of CDD in saving the energy is between 50% and 63.64% while comparing it to DD and, 8% and 42.8% compared to PCDD, for the fixed density scenario. For the fixed area scenario the improvement is up to 29.4% and at least 15% while comparing CDD to PCDD and up to 63.1% and at least 26.1% while comparing it to DD. The use of a TDMA-MAC style to reduce the *idle* energy consumption, may create some latency because the node may stay in sleep mode, this is the tradeoff delay for saving energy. Because of that CDD suffers from some latency while comparing to PCDD In terms of packet delivery latency CDD shows a great improvement over DD. However, DDLSC has the lowest delivery delay among the three routing protocols. The results are expected since PCDD assumes long transmission ranges between the cluster head in which the delay has been reduced.

4.4 Evaluating CDD

The original DD routing protocol performance is improved by utilizing different hierarchical techniques; combining DD with different topology control like directed diffusion with passive clustering (PCDD)[28] or Directed Diffusion Based on Link-Stabilizing Clustering (DDLSC)[23]. Additionally, in [29] Energy Conserving Passive Clustering (ECPC), in [30] Enhanced Passive Clustering Directed Diffusion (EPCDD) and in [31] Energy Level-based Passive Clustering (ELPC) have been proposed focusing on the improvements of PCDD utilizing same approach in[28]. Table 4.1 gives the features and limitations of DD related protocols.

PCDD uses *first declaration wins* method when it creates its cluster heads. This may partition the sensor network as nodes with low energy might win the cluster head role. In addition, it ignores the idle energy consumption and does not support mobile sensor nodes. ECPC is also a passive clustering but, it considers residual energy and restricts number of cluster heads in the network. ECPC shows better than DD and PCDD in terms of energy saving and increasing the lifetime. However, it has a great overhead while creating the clusters, very limited in mobility and utilizes the same MAC layer which is used in PCDD. Other passive clustering routing protocols are EPCDD and ELPC, for the same author. In EPCDD the author combine DD with passive clustering to improve the energy consumption and increase WSNs lifetime. ELPC is the energy-level based passive clustering and they show improvements over PCDD and DD in terms of energy and network lifetime. However, neither EPCDD nor ELPC consider idle energy consumption. Additionally, none of them support mobile sensor nodes. DDLSC a good clustering and selects a cluster heads based on waiting time between the nodes. The node marks itself as unclustered and pick up a random waiting time, if it does not heard from any one the nodes within the waiting time, it declare itself as the cluster head. Then it sends the message for its neighbors and, each node receives the message will send back to become the slave of the cluster head. DDLSC shows good improvements in terms of network coverage and, energy-efficiency and low latency over DD. This is because; in DDLSC a long-range communications are used for inter-cluster which is 200m and 100m for intra-cluster communications [23]. However, as the same time it requires more energy for sensor nodes to transmit data over longer hops than utilizing shorter hops to forward them [59]. In addition, DDLSC uses the same MAC layer which has been used in DD and it does not support mobile sensor nodes.

The original directed diffusion utilizes 802.11 MAC and, the authors stated that the power consumption of 802.11 radio for idle and transmission is close to each other. Additionally, they declared that the MAC layer which has been used in their work is not completely satisfactory, since a TDMA-style MAC is more efficient and is necessary for long-lived sensor nodes. Also, TDMA schedule algorithm can reduce the idle-energy consumption as well.

With the contrary of conventional wireless networks, WSNs have to support a large number of sensor nodes associated with short range radio communications. A short-distance separation between sensor nodes can be exploited to provide multihop communication in the way that energy of the sensors can be saved. In short hops, the energy, which is required to transmit data, is nearly same as that of receiving data. Thus, the design of routing protocols should include a MAC address which includes a TDMA to make the radios off during the idle state. A time-division style MAC can be

Table 4.1: Features and limitations of DD related routing protocols

Features / Method	Network Topology	Features/advantages	Limitations/drawbacks
DD[24, 49]	Flat	It can select the best route for delivering data	Ltd. in scalability, mobility and energy. More overhead from interest/exploratory data consumes more energy.
PCDD[28]	Hierarchical-PC	Improves delay, delivery rate and energy over DD	Has limitation for mobility and ignores energy consumption for "*Idle*" and *First declaration wins* may partition part of the network.
ECPC[29]		Better than DD and PCDD in terms of Energy saving & increasing the lifetime	Extra-overhead while creating clusters and very limited in mobility.
EPCDD[30] & ELPC[31]		Enhancement over DD and PCDD, in terms of energy and network lifetime	Does not support mobile nodes, ignores *idle* energy consumption
DDLSC[23]	Hierarchical-LSC	Low latency Energy-efficient Scalable	Does not support mobility and uses long transmission range between CHs, uses same MAC layer as DD
CDD/CDDM	Hierarchical-with TDMA-MAC	Energy-efficient /TDMA, Scalable, Handle mobile nodes Limit flooding	delivery latency

utilized to periodically exchange messages and maintain local synchronization between the sensor nodes [22].

As it is stated in [22, 24, 49], designing an energy-efficient routing protocol should be associated with a TDMA-style MAC to save sensors' energy guarantee the longevity of WSNs lifetime. Although CDD does not consider clearly energy level while selecting the cluster heads, but implicitly it is there, according to Eq. 3.1 the cluster heads have been selected in the way that any node which served as the cluster head for the current round, will be excluded when the cluster heads are elected for the next round. In this way the energy evenly distributed among the sensor nodes and the burden of cluster heads will be reduced.

Finally, the overall performance of CDD outperforms DD, in terms energy-efficiency, data delivery latency and data delivery ratio. Additionally, as PCDD, ECPC, EPCDD, ELPC and DDLSC ignore the idle-energy consumption and, CDD is considering it by using TDMA schedule algorithm, it was evidence that CDD saves more energy than them. Also, nowadays most applications of WSNs require either mobility of the sink node or the sensor nodes, in which the sink node mobiles to collect the data or the sensor nodes move to compensate the lack of coverage. None of the above protocols consider sink mobility or the nodes. In the next section CDD has been extended to handle the mobility of the sink node and the combination of mobile and static nodes as well.

4.5 *CDDM*: Query-driven Cluster-based Data Diffusion Routing Protocol for Mobile Sensor networks

This section explains the extension of Cluster-based Data Diffusion routing protocols for WSNs to handle combination of mobile and fixed sensors and sink's (user's) mobility as well.

This section considers the combination of mobile and stationary nodes with sink mobility. In mixed scenarios, mobile sensor nodes can be exploited to detect the sources which might not be covered by the stationary nodes. In addition, the mobile sensor nodes can be utilized to compensate for the lack of sensor nodes and to enhance the coverage of the network as well[32]. Various applications of WSNs require the combination of both mobile and fixed sensor nodes in the same network in order to decrease the energy consumption by reducing the amount of information that requires to be exchanged between nodes [60]. The combination of mobile and stationary sensor nodes has recently been studied [61, 62]. In many applications, the sensor nodes may be mobile as they can move around using various methods like, self propelling (for e.g. via wheels, micro-rockets, or others means) or by attaching the sensor nodes to different means of transporters (e.g. robots, vehicles, animals, air, and water) [63]. However, entirely mobile sensor nodes is beyond the scope of our proposed method and it will be treated as recommendation for future work.

In WSNs the gathering of data can be followed in various ways according to the applications of the sensor nodes. There are three ways being used as data reporting from the sensor nodes. In time-driven, the sensor nodes transmit sensed data to the sink node periodically, whereas in event-driven scenario the sensor nodes start sending data to the sink node if they detect an event. The third case is the query-driven, in which the sink node broadcasts the query to the sensor nodes in order to get information from some geographical area, for example, a reading of temperature which is more than some threshold value [63, 64].

4.5.1 Motivation

There are many research works that have studied the use of mobile sinks in many different applications [65]. For example, a biologist who needs to collect data of wild life monitoring like birds, zebras and others. The biologist does not stay for the complete collecting period, for example two weeks, in the sensing area. Instead, he deploys the sensor nodes in some geographical area and after the specified time has elapsed, he would go to the sensing place and moves around the nodes, using mobile sinks like PDA, laptops and many other devices, and collects the data he needs.

The mobile information sink can move around the nodes and collects the data. For example, at the supermarkets you want to have some read-out of the sensor's stock using electronic devices throughout the supermarket. It can also be helpful to know how many products are left, and the information can be sent not only to the owner of the supermarket, but also to the company that produces the product.

Other applications that have been studied using sink's (user's) mobility are firefighters and soldiers who need to gather data from the sensor field, such as rescue in a disaster area or maneuvers in a war cluster using large-scale wireless sensor networks. The user receives data from the sink node through multihop communications by applying the cluster heads .This situation can be used for rescue operations when the mobile sink (emergency vehicle) joins the field to collect data from the sensor nodes distributed over the disaster area [66].

The same assumptions which have been used in chapter 3 for the stationary nodes and fixed sink have been adopted, except that in this section the sink (user) has been considered as a mobile device and partial mobility of the sensor nodes is also considered. Also, we assume that the sensor nodes are location-aware using GPS receiver. The cluster heads are proactively elected and every node joins the nearest cluster head to its position.

4.5.2 Cluster Head Election and Advertisement

Cluster heads are elected using the same approach used in LEACH routing protocol, in which every node competes with other sensor nodes using the threshold according to equation 3.1, which is specified in LEACH protocol. The election of the cluster heads is repeated every round to guarantee that the energy consumption is evenly distributed among the sensor nodes. As discussed in chapter 3, when the node is elected as the cluster head, it broadcasts its role using fixed signal strength to the rest of the sensor nodes in the field.

4.5.3 Join-Request and TDMA Schedule

Any nodes that receive the advertisement message from more than one cluster heads selects the closest cluster head to its position and send a join-request message to it. Once the nodes are organized in clusters, each cluster head creates a TDMA schedule for each node in its cluster and broadcasts it to them. The TDMA schedule is used by each node to send data to its cluster head, avoids data collision among data messages, and helps to turn the radio off for each non-cluster head node at all times except during its transmission time. Each node detects its data cache and when the cluster head sends the interest message, it checks if the data is matched to the interest. If it matches, it sends the data back to the cluster head; and if not after the time for the data cache has elapsed it removes the data from the cache.

4.5.4 Interest and Data Messages

Whenever a target node generates a signal, sensor nodes within the radius of the target can detect and cache it in the data cache entry for further use of it. Once the network has been set up, the mobile sink disseminates the interest message to the specified geographical region through the cluster heads. The members of each cluster head can receive the interest message through their cluster heads. As soon as the member nodes receive the interest message, they will search for a match for the received interest among the previously cached data. If a match is found, they will send the data through reversed path to their cluster head, and the sink node can receive the data of interest through multihop cluster heads.

4.5.5 Cluster Maintenance

To reduce the burden over the cluster heads, the cluster heads are elected every round using the formula as specified in LEACH protocol. And the rest of the procedures will be repeated until the simulation is complete. As the sink node is not energy constrained like the other sensor nodes, whenever the sink node broadcasts its new location it can be heard by the whole nodes in the network field, and all the nodes in the sensor field can update the sink location.

4.6 Simulation Environment and Simulation Results

Three metrics have been chosen to evaluate the performance of the CDDM: average dissipated energy, packet delivery ratio and average latency. These metrics have been used for the stationary sensor nodes to compare CDD with original DD as well as DDLSC and PCDD. These metrics are studied as a function of network density and network size.

4.6.1 Performance Evaluation with Fixed Density

To study the performance of the proposed method as a function of network size, three different sensor fields: 100, 200 and 300 sensor nodes, have been generated. The first 100 sensors are generated and randomly deployed over a square 225m by 225m field. To maintain the density of the sensor nodes in the network, other networks are generated by scaling the network size. One mobile sink was deployed and five sources (targets) have been distributed in the left bottom area of the sensor field. The power consumption for transmitting and receiving is 0.66W and 0.395W, respectively. The idle power consumption for the sensor node is 0.035W. The radio range for each sensor is 40m. The interest packet size is 36 bytes where the data packet is 64 bytes. The target nodes generate two stimuli per second. After the network setup has been completed, the sink node sends the interest message unicastly only to the cluster heads. The interest message is disseminated from the sink node every 10 sec. The speed of the sink node is between 0 and 3m/s based on the random Waypoint mobility model, and the pause time between two movements for the sink node is 10sec.

The results are obtained using J-Sim simulator as shown in Figures 4.21 through 4.23 for the fixed density. To investigate the performance of the proposed protocol and to evaluate the advantages and disadvantages of it; we compare CDDM, which assumes sink’s mobility, and combination of static and mobile sensor nodes with the well-known data centric routing protocol DD (which assumes static sink). Figure 4.21 shows the average dissipated energy using metric (Joules/Node/Distinct event) as a function of fixed density. The CDDM achieves almost constant, for the three generated sensor fields, and extremely low energy consumption in comparison to DD. This is due to the limitation of forwarding the interest message unicastly to cluster heads and retrieving the data from target nodes utilizing multihop communication between cluster heads.

Figure 4.22 compares the average delay for the three sensor network field of CDDM to those of DD. As it can be seen from the figure, in CDDM the delay decreases as the number of participated mobile sensor nodes increases in the network. Moreover, in CDDM the delay is considerably lower than in DD protocol, as the sink node can move closer to the interested area and collects the required data as the delay can be reduced accordingly.

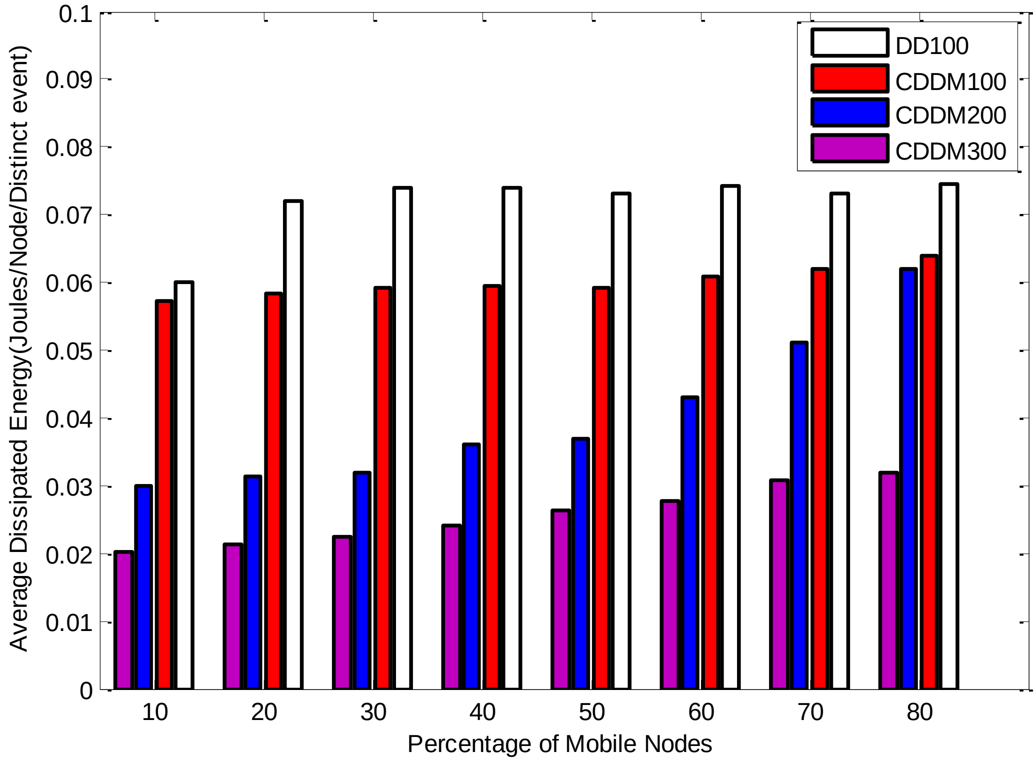


Figure 4.21: Energy Efficiency – fixed density scenario

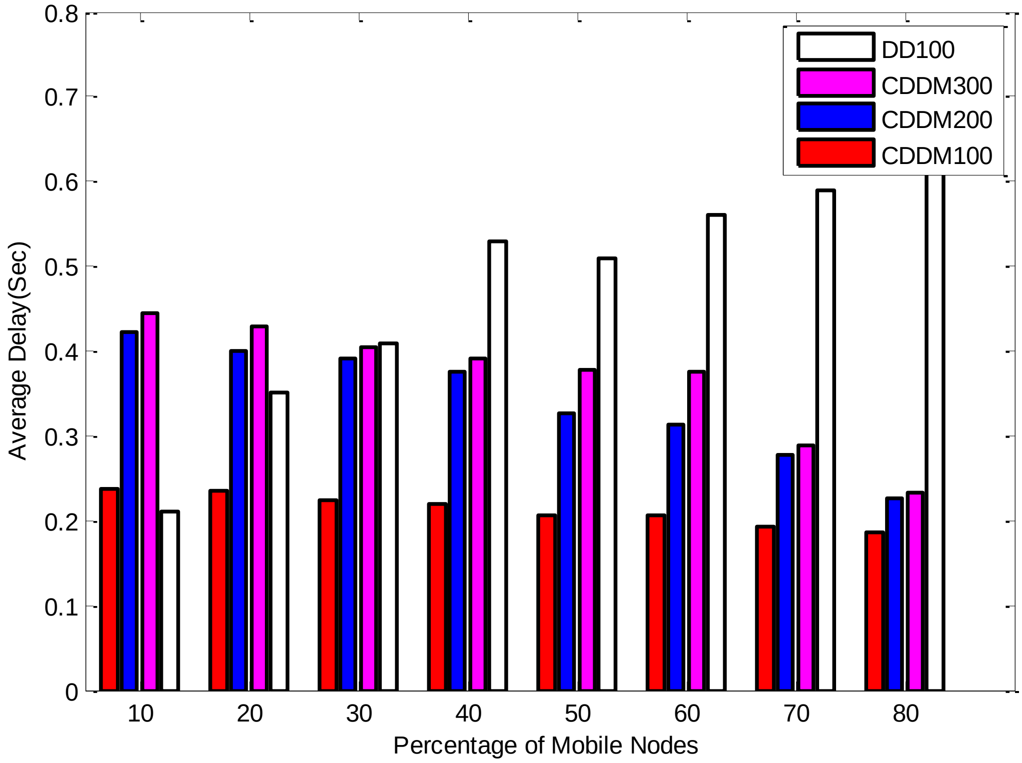


Figure 4.22: Average Delay – fixed density scenario

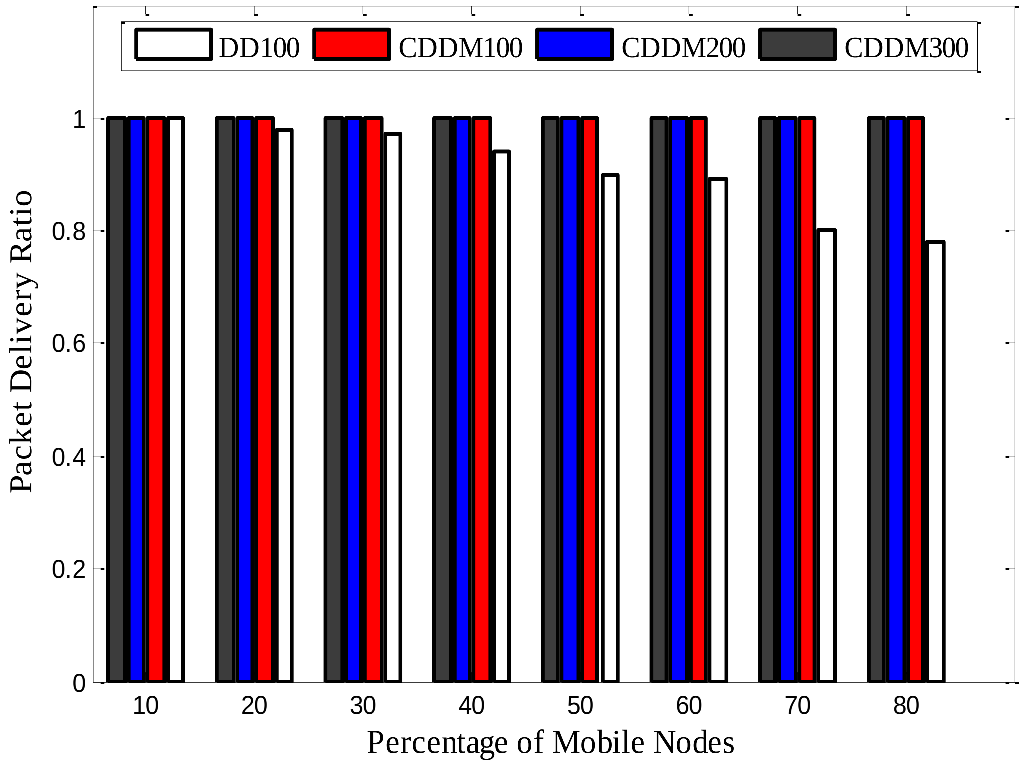


Figure 4.23: Packet Delivery Ratio – fixed density scenario

The result in Figure 4.23 illustrates the packet delivery ratio in which CDDM is delivering almost 100%. This refers to the use of mixed nodes, in which the mobile nodes compensate for the lack of sensor nodes in the sources area and enhance the coverage.

Finally, when comparing CDDM with DD, the proposed protocol outperforms DD routing protocol, and it is able to achieve low energy consumption, high packet delivery ratio and in some cases low latency.

4.6.2 Performance Evaluation with Fixed Area

To evaluate the performance of CDDM in denser network, three networks of sensor nodes 100-300 have been generated and randomly distributed over an area that measures 200m by 200m. To conduct the simulation for the generated networks, the same configurations which are used in 4.5.2 have been adopted.

Figure 4.24 shows the average dissipated energy of the CDDM as a function of network density by utilizing the metric (Joules/Node/Distinct event). As we can see, the average dissipated energy decreases particularly as the number of nodes increases from 100 to 300 in the same network field. When we compare CDDM with DD routing protocol, it achieves a better result as the number of nodes increases in the same field. As DD protocol uses flat network architecture, it needs to build long paths between the sources and the sink (which is in a fixed position). This consumes a great amount of energy which reduces the network lifetime.

Thus, it was evidence that CDDM can save energy and prolongs network lifetime. However, the saved energy is at the cost of delivery latency between the source and the sink node, which is shown in Figure 4.25. The latency between the source node and the base station keeps increasing as the number of nodes increases in the same network area. This is because of large number of sensor nodes are participating in delivering a huge amount of information about the interested area, which takes a long time particularly for the cluster heads to aggregate the data from the member nodes and pass it to another cluster head or the base station.

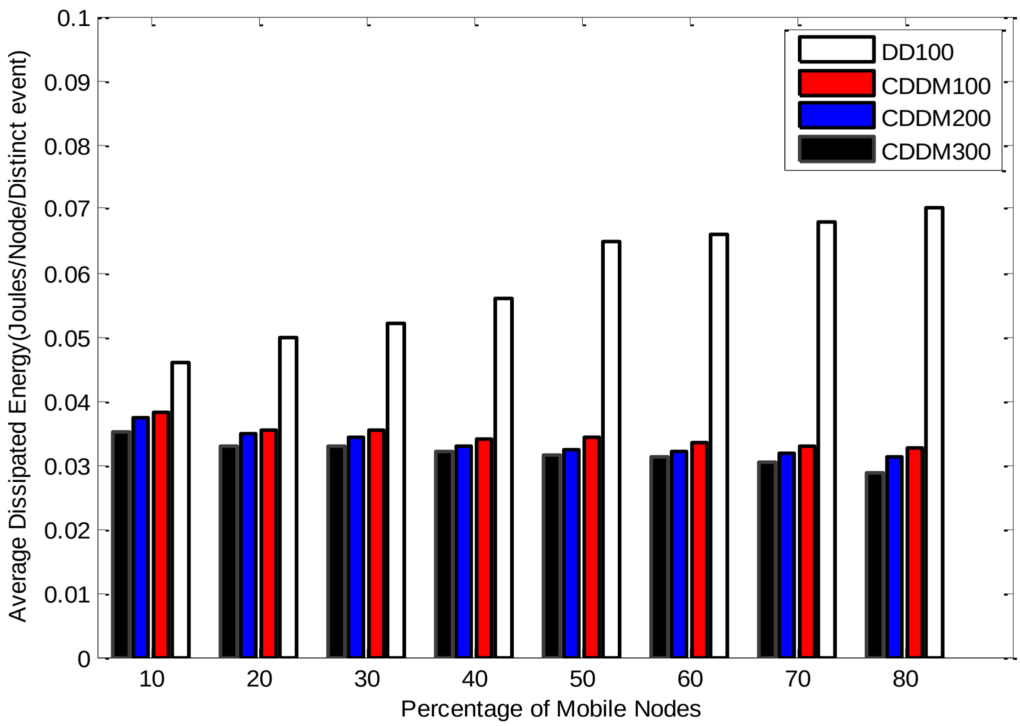


Figure 4.24: Energy Efficiency – fixed area scenario

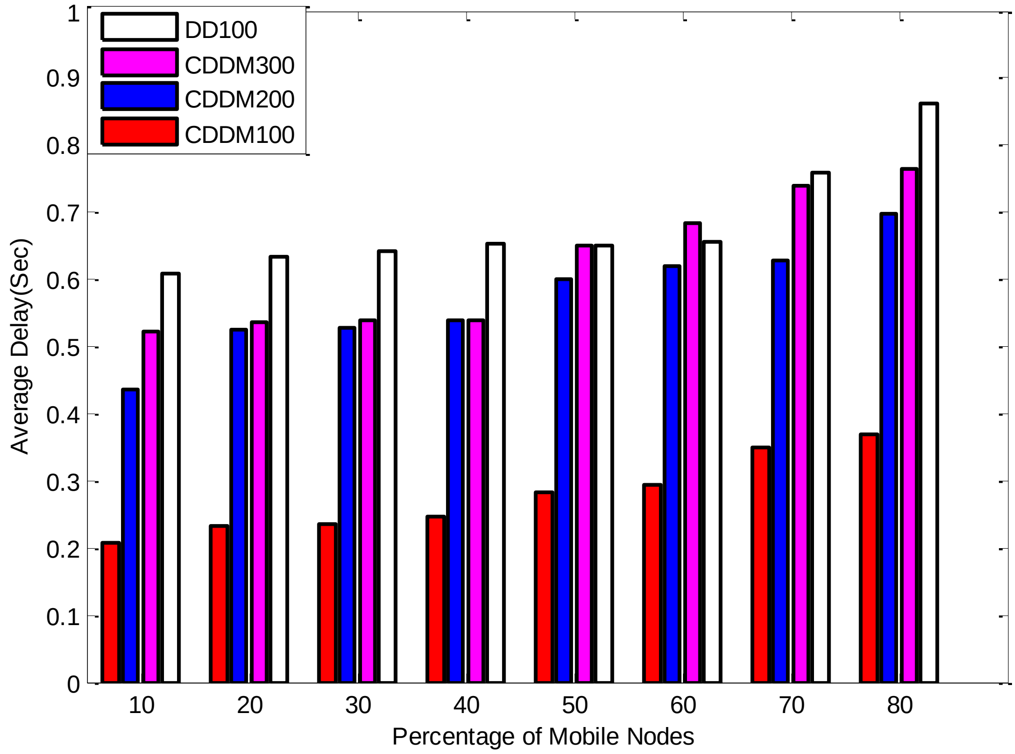


Figure 4.25: Average Delay – fixed area scenario

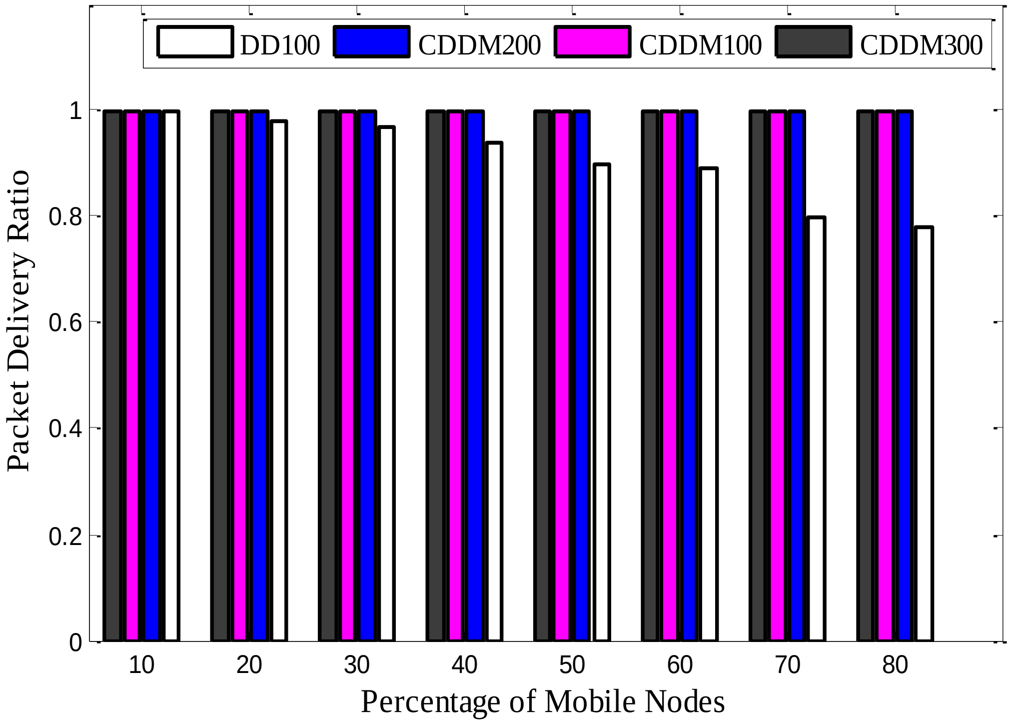


Figure 4.26: Packet Delivery Ratio – fixed area scenario

In Figure 4.26 packet delivery ratio of CDDM is plotted and compared to DD protocol. Since many nodes are collaborating to gather the data from the interested area, CDDM is delivering almost 100% in the three sensor fields.

Based on the three graphs of the fixed area scenario, in which the number of nodes increases in each cluster as the number of sensor nodes increases in the same field, CDDM has achieved lower energy consumption and close to the optimal packet delivery ratio but it has some delay.

4.7 Summary

The proposed cluster-based CDD routing protocol has shown a good performance over a flat-based DD. The evaluation of CDD using various scenarios; fixed density, fixed area and, different number of sources, have proven that CDD outperforms DD, DDLSC and PCDD in terms energy saving, delivery latency and packet delivery ratio. In addition by considering the idle-energy consumption, the overall performance of CDD is shown that, it outperforms all of the routing protocols which have been proposed to enhance the original DD. In addition, the use of combination of mobile and static sensor nodes is found to improve the coverage and compensates the lack of sensor nodes. This has been proven when CDDM, which handles the mixed of mobile and static sensor nodes and sink's mobility, has shown a good performance in terms of energy saving and delivering the required data. However, CDDM is found to have a little delivery delay in particular for the fixed area scenario simulation. In addition, it is shown aspects like node density and network size affects the behavior of the proposed protocol.

Finally, CDDM is proven to achieve low energy consumption, in comparison to DD. It is noticed that, in denser network CDDM delays increase because of large number of sensor nodes participating in delivering the required data.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

Wireless Sensor Nodes can be randomly deployed over the monitoring area for instance, on a battlefield, disaster area or in inaccessible areas to humans. The network is infrastructure-less and the nodes are self-configured and unattended. So, the most constrained part of the sensor node is the energy supply as it cannot be replaced due to difficulties to access the area where they have been deployed.

The aim of the research is to develop a query-driven cluster-based data diffusion routing protocol for large-scale wireless sensor networks, to prolong sensor nodes lifetime in terms of energy and data efficiency. The use of clustering method and multihop communications between cluster heads help to achieve the main concerns of WSNs, extending WSNs lifetime by reducing the energy consumption, delivering the required data to the end-user in lesser time delay between the sources (targets) and the destination (sink).

The study of performance analyses for four existing WSNs routing protocols by contrasting their features was done. We found that, some of these protocols were not scalable for large-scale WSNs, were not energy-efficient and were not support sensors' mobility. These investigations discussed the challenges in improving some of application-specific WSNs routing protocol that achieves energy efficiency and scalability.

The application-specific CDD routing protocol was developed with features of the original directed diffusion: data-centric routing, attribute-based naming and in-network data aggregation. In addition, in CDD clustering scheme and multihop communication was used. By using clustering technique, CDD avoided broadcasting the interest message to the whole network (interests were unicastly sent only to the cluster heads) and the exploratory data as well. Additionally, the use of multihop communications between cluster heads in CDD contributed in achieving the scalability for large scale sensor networks. CDD aggregated the data at the cluster heads and reduced the amount of data (by maintaining the information content) which were sent to the sink node. In this way CDD reduced data overhead, achieved WSNs longevity by saving the energy and fitted for large-scale WSNs. These features are crucial for energy and resource constraints WSNs.

CDD used a TDMA scheduling algorithm, to reduce the idle-energy consumption and, so the sensors can go to a sleep state as long as possible to conserve the energy. In addition, as the combination of mobile and static sensors can be exploited to improve the coverage and compensate the lack of sensor nodes, the extension of CDD to handle node's mobility was introduced.

We implemented the CDD in J-Sim simulator. We compared the performance of CDD against the original DD routing protocol in terms of: average dissipated energy, packet delivery latency and packet deliver ratio. Our evaluation showed that CDD outperforms DD and other clustering schemes; which were proposed to enhance DD performances, in terms of energy saving and handling mobile nodes.

As a conclusion, based on the results obtained by using the open source J-Sim simulator, the proposed routing protocol for WSNs has shown good performance in both stationary and low speed sensor nodes.

5.2 Future Work

The proposed method can be extended to address fully mobile sensor nodes which need new approaches in order to support full broadcasting of the interest message and to manage it for high mobility sensor nodes. In high mobile, the performance of the sensors may degrade because of frequent change of the position of sensors and the sink node. Moreover, as the sensor nodes are energy-constrained topology changes may create a significant overhead. Thus, it needs new methods, so there will be a trade-off between saving sensors' energy and delivering the required data.

In high mobility of the sink and the sensor nodes, the user may leave its current position, after the query has been sent at the time the answer for the query will submitted from the nodes. In this case the query answer must be submitted by the sensor nodes. The user either rejects the answer from the base station or it asks the base station to process the query again.

The main objective of WSNs is energy saving. We assumed in our work that all nodes are homogeneous in which all participated nodes are equally powered. However, node heterogeneity can be utilized in sensor network in which some powerful battery end nodes can be participated. This can be achieved by adding some end nodes equipped with powerful battery associated with a large number of cheap sensor nodes. They can help, by increasing link bandwidth in the more powerful battery end nodes [33].

All the simulations have been conducted only for one sink. However, multiple queries from multiple users (sinks) can be injected to the sensor field at the same time. The more challenge is to design a query-based routing protocol which supports a multiquery with one or more sensing capability [33]. In future, multiple sinks for stationary, combination of mobile and static sensor nodes and for fully mobile nodes can be investigated. Finally, many different WSNs applications can be explored and, based on the applications any changes required for the proposed method can be extended.

REFERENCES

1. Vieira, M.A.M., A.B.d. Cunha, and D.o.C.d.S. Jr. Designing Wireless Sensor Nodes. in 6th Workshop on Embedded Computer Systems: Architectures, Modeling,and Simulation (SAMOS'06). 2006. Samos, Greece.
2. Zhao, J. and A.T. Erdogan. A Novel Self-Organizing Hybrid Network Protocol for Wireless Sensor Networks. in the first NASA/ESA conference on Adaptive Hardware and Systems. 2006.
3. Crossbow Technologies Incorporated. *http://www.xbow.com*. 2007 [cited.
4. Chatterjea, S., Distributed and Self-Organizing Data Management Strategies for Wireless Sensor Networks: A Cross-Layered Approach. 2008, University of Twente.
5. Al-Karaki, J.N. and A.E. Kamal, Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications, 2004. **11**(6): p. 6 - 28.
6. Noori, M. and M. Ardakani. A probability model for lifetime of wireless sensor networks. 2007: Computing Research Repository.
7. Healy, M., T. Newe, and E. Lewis. Wireless Sensor Node Hardware: A Review. in the IEEE Conference on Sensors. 2008. Lecce, Italy.
8. Howard, A. and M.M.G. Sukhatme. From Mobile Robot Teams to Sensor/Actuator Networks: The Promise and Perils of Mobility. in First Workshop in GeoSensor Networks. 2003.
9. Cui, L., et al., A Pervasive Sensor Node Architecture. Lecture Notes in Computer Science, 2004. **3222**: p. 565-567.
10. Bischoff, R., J. Meyer, and G. Feltrin, Wireless Sensor Network Platforms. Encyclopedia of Structural Health Monitoring, ed. C. Boller, Chang, F. & Fujino, Y. 2009: Chichester, John Wiley & Sons.
11. Akyildiz, I.F. and M.C. Vuran, Wireless Sensor Networks. 2010: John Wiley & Sons.
12. Ar Milenković, C. Otto, and E. Jovanov, Wireless sensor networks for personal health monitoring: issues and an implementation. Computer

Communications 2006. **29**: p. 2521-2533.

13.Barrenetxea, G., et al. Wireless Sensor Networks for Environmental Monitoring: The SensorScope Experience. in The 20th IEEE International Zurich Seminar on Communications (IZS 2008). 2008. Zurich, Switzerland.

14.Ferro, J.M., et al. Applications of Wireless Sensor Networks. in the World Congress on Engineering. 2007. London, U.K.

15.Younis, O. and S. Fahmy. Distributed clustering for scalable, long-lived sensor networks. in The Ninth Annual ACM, International Conference on Mobile Computing and Networking (MobiCom 2003). 2003. San Diego, California, USA.

16.Heinzelman, W.B., Application-specific protocol architectures for wireless networks. 2000, Massachusetts Institute of Technology.

17.Mishra, N., Capacity and non-steady state generalizations to the Dynamic MEXCLP model for distributed sensing networks. 2003, University at Buffalo.

18.Karl, H. and A. Willig, Protocols and Architectures for Wireless Sensor Networks. 2005: Chichester: John Wiley & Sons.

19.Liu, T., et al. Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet. in the 2nd international conference on Mobile systems, applications, and services. 2004: ACM Press.

20.Q, Y., Z.Y. C, and C. L., A Study on the optimal time synchronization accuracy in wireless sensor networks. Computer Networks, 2005. **48**(4): p. 549-566.

21.Chu, X. and R. Buyya, MANET versus WSN", Sensor Network and Configuration: Fundamentals, Standards, Platforms, and Applications, ed. N.P. Mahalik. 2007: Springer-Verlag, Berlin, Germany.

22.Pottie, G.J. and W.J. Kaiser, Wireless Integrated Network Sensors. Communications of the ACM, 2000. **43**(5): p. 51 - 58

23.Zhou, Z., et al., Directed Diffusion Based on Link-Stabilizing Clustering for Wireless Sensor Networks. UBIQUITOUS INTELLIGENCE AND COMPUTING, Lecture Notes in Computer Science, 2007. **4611**: p. 485-494.

24.Intanagonwiwat, C., R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. in ACM MobiCom. 2000.

25.Rabiner, H.W., C. Anantha, and B. Hari, An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions On Wireless Communication, 2002. **1**(4).

26.Karp, B. and H.T. Kung. GPSR: Greedy Perimeter Stateless Routing for wireless networks. in the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom). 2000. Boston, MA.
27.Perkins, C.E. and E.M. Royer. Ad-Hoc On-Demand Distance Vector Routing. in the 2nd IEEE Workshop on Mobile Computing Systems and Applications. 1999. New Orleans, LA.
28.Handziski, V., et al. Improving the Energy Efficiency of Directed Diffusion Using Pervasive Clustering. in 1st European Workshop in Wireless Sensor Networks (EWSN). 2004. Berlin, Germany.
29.Md., M.-o.-R., M. Mahbub Alam, and C.S. Hong;. Energy conserving passive clustering for efficient routing in wireless sensor network. in 9th International Conference on Advanced Communication Technology. 2007.
30.Zeghilet, H. Performance improvement of passive clustering algorithm in wireless sensor networks. in the 4th International Conference on Networked Sensing Systems (INSS). 2007.
31.Zeghilet, H. Energy Efficient Cluster-based Routing in Wireless Sensor Networks. in the 14th IEEE Symposium on Computers and Communications (ISCC'09). 2009. Sousse, Tunisia.
32.Liu, B., et al. Mobility improves coverage of sensor networks. in the 6th ACM international symposium on Mobile ad hoc networking and computing. 2005. Urbana-Champaign, IL, USA.
33.Zheng, J. and A. Jamalipour, Wireless Sensor Networks: A Networking Perspective. 2009: John Wiley & Sons.
34.C., H.T. and T.T. J., An access-based clustering protocol for multihop wireless ad hoc networks. IEEE Journal on Selected Areas in Communications, 2001. **19**(7): p. 1201-1210.
35.M., J.-N. and L. I.T., A Peer-to-peer Zone-based Two-level link state routing for mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communications, Special Issue on Ad-hoc Networks, 1999. **17**(8): p. 1415-1425.
36.Baker, D.J. and A. Epheremides, The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm. IEEE Transactions on Communications, 1981. **Com-29**(11).
37.Ephremided, A., J.E. Wieselthier, and D.J. Baker, A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling. Proceedings of the

IEEE 1987. **75**(1): p. 56-73.

38.Hossain, E., R. Palit, and P. Thulasiraman, Clustering in mobile wireless ad hoc networks. Wireless Communications Systems and Networks, ed. M. Guizani. 2004: Kluwer Academic/Plenum Publishers.

39.Baker, D., A. Ephremides, and J.A. Flynn, The design and simulation of a mobile radio network with distributed control. IEEE Journal on Selected Areas in Communications, 1984. **SAC-2**(1): p. 226–237.

40.Tsigas, P., Project on Mobile Ad Hoc Networking and Clustering for the Course EDA390 Computer Communication and Distributed Systems. Manual for University Course.

41.Amis, A., et al., Max-Min D-Cluster Formation in Wireless Ad Hoc Networks. IEEE INFOCOM, 2000.

42.Chatterjee, M., S.K. Das, and D. Turgut, WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. Clustering Computing, 2002. **5**: p. 193–204.

43.Heinzelman, W.R., A. Chandrakasan, and H. Balakrishnan. Energy- Efficient Communication Protocol for Wireless Sensor Networks. in the 33th Hawaii International Conference on System Sciences. 2000.

44.Loscri, V., G. Morabito, and S. Marano. A Two-Level Hierarchy for Low-Energy Adaptive Clustering Hierarchy. in IEEE 62nd Vehicular Technology Conference. 2005.

45.Ye, M., et al. EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks. in IEEE Int’l Performance Computing and Communications Conference (IPCCC). 2005.

46.Younis, O. and S. Fahmy, HEED: A Hybrid Energy-Efficient Distributed Clustering Approach for Ad Hoc Sensor Networks. IEEE Transactions on Mobile Computing, 2004. **3**(4).

47.Lindsey, S. and C.S. Raghavendra. PEGASIS: power-efficient gathering in sensor information systems. in Aerospace conference. 2002.

48.Xie, Z., C. Hong, and L. Wang. Subnets based distributed data-centric hierarchical ant routing for sensor networks. in IEEE International Conference on Wireless Communications, Networking and Mobile Computing. 2005.

49.Intanagonwiwat, C., et al., Directed diffusion for wireless sensor networking. IEEE/ACM Transactions on Networking, 2003. **11**(1): p. 2 - 16

50.Gadallah, Y. A Comparative Study of Routing Strategies for Wireless Sensor

Networks: Are MANET Protocols Good Fit? in the 5th International Conference on Ad Hoc Networks and Wireless. 2006. Ottawa, Canada.

51.M., L., A. C., and N. Nasser. Cluster-based Routing Protocol for Mobile Sensor Networks. in The Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks. 2006. Waterloo, Ontario, Canada: ACM.

52.Sobeih, A., et al. J-Sim: A simulation environment for wireless sensor networks. in the Annual Simulation Symposium (ANSS 2005), part of the 2005 Spring Simulation Multiconference (SpringSim 2005).

53.http://j-sim.cs.uiuc.edu/. [cited.

54.http://sites.google.com/site/jsimofficial/. [cited.

55.Sobeih, A., et al., J-Sim: A simulation and emulation environment for wireless sensor networks. IEEE Wireless Communications Magazine, 2006. **13**(4): p. 104-119.

56.Tyan., H.-Y., Design, Realization and Evaluation of a Component-based Compositional Software Architecture for Network Simulation. 2002, Department of Electrical Engineering, The Ohio State University.

57.Tyan, H.-Y., A. Sobeih, and J.C. Hou, Design, realization, and evaluation of a component-based compositional network simulation environment. Conditionally accepted for publication in SIMULATION: Transactions of The Society for Modeling and Simulation International., 2008.

58.Tyan, H.-Y., A. Sobeih, and J.C. Hou. Towards composable and extensible network simulation. in the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05). 2005: IEEE Computer Society.

59.Liu, X. and P. Mohaparta, On the deployment of wireless data backhaul networks. IEEE Transactions on Wireless Communications, 2007. **6**(4): p. 1426–1435.

60.Lambrou, T.P. and C.G. Panayiotou. Distributed collaborative path planning in sensor networks with multiple mobile sensor nodes. in 17th Mediterranean Conference on Control and Automation. 2009.

61.Awwad, S.A.B., et al., Cluster Based Routing protocol for Mobile Nodes in Wireless Sensor Network. 2009 International Symposium on Collaborative Technologies and Systems, 2009: p. 233-241.

62.Kim, D.-S. and Y.-J. Chung. Self-Organization Routing Protocol Supporting Mobile Nodes for Wireless Sensor Network. in First International Multi-

Symposiums on Computer and Computational Sciences. 2006.
63.Lee, W.-C., Query Processing in Mobile Sensor Networks. unpublished.
64.Cugola, G. and M. Migliavacca. A Context and Content-Based Routing Protocol for Mobile Sensor Networks. in the 6th European Conference on Wireless Sensor Networks. 2009. Cork, Ireland.
65.Vincze, Z., et al. Adaptive sink mobility in event-driven multi-hop wireless sensor networks. in the first international conference on Integrated internet ad hoc and sensor networks. 2006. Nice, France.
66.Ye, F., et al. A two-tier data dissemination model for large-scale wireless sensor networks. in The 8th annual international conference on Mobile computing and networking. 2002: ACM.

APPENDIX A

Sample of TCL for 200 nodes-fixed area scenario.

```
# *********************************************************
#
# the script simulates a sensor network which utilizes, Clustered Data Diffusion
#called CDD routing protocol. J-Sim was modified to handle the protocol
#
#
#*********************************************************

source "include.tcl"

cd [mkdir -q drcl.comp.Component /CDD]

# TOTAL number of nodes (sensor nodes + target nodes)
set node_num 206

# Number of TARGET nodes ONLY
set target_node_num 5
# Hence, number of SENSORS = node_num - target_node_num

set sink_id 0

# create the sensor channel
mkdir drcl.inet.sensorsim.SensorChannel chan
```

```
# Capacity of the sensor channel is total number of nodes (sensors + targets)
# make simulation for $node_num nodes
! chan setCapacity $node_num

# create the propagation model
mkdir drcl.inet.sensorsim.SeismicProp seismic_Prop
! seismic_Prop setD0 0.2
# create the sensor node position tracker
mkdir drcl.inet.sensorsim.SensorNodePositionTracker nodetracker
# maxX  minX maxY minY
! nodetracker setGrid 200.0 0.0 200.0 0.0

# connect the sensor channel to the sensor node position tracker
connect chan/.tracker@ -and nodetracker/.channel@

# create the wireless channel
mkdir drcl.inet.mac.Channel channel

# Capacity of the wireless channel is number of sensors and sinks ONLY
# which is equal to $node_num - $target_node_num
! channel setCapacity [expr $node_num - $target_node_num]

# create the node position tracker
mkdir drcl.inet.mac.NodePositionTracker tracker
#the dx and dy below represent 'how far' my signal travels
#so in this case any node located in my 100x100m grid will hear
#what a sensor broadcasts
#                    maxX minX maxY minY  dX   dY
! tracker setGrid 200.0 0.0 200.0 0.0 100.0 100.0

connect channel/.tracker@ -and tracker/.channel@

#**************************************************
# FOR THE SINKs ONLY, do the following
```

```
# SINKs have only a network protocol stack
for f {$i < [expr $sink_id + 1]} {incr i} {
	puts "create sink $i"
	set node($i) [mkdir drcl.comp.Component n$i]

	cd n$i
	mkdir drcl.inet.sensorsim.CDD.SinkAppCDD app
	! app setNid $i
	! app setSinkNid $sink_id
	! app setCoherentThreshold 1000.0
	! app setTargetName "Wheeled Vehicle"


	# connect the sensor application to the wireless agent
	# so that sinks can send through the wireless network protocol stack
	# create wireless agent layers
	mkdir drcl.inet.sensorsim.CDD.WirelessLEACHAgent wireless_agent

	# connect the sensor application to the wireless agent
	# so that sensors can send through the wireless network protocol stack
	connect app/down@ -to wireless_agent/up@

	# connect the wireless agent to the sensor application
	# so that sensors can receive thru the wireless network protocol stack
	connect wireless_agent/.toSensorApp@ -to app/.fromWirelessAgent@

	mkdir drcl.inet.mac.LL ll
	mkdir drcl.inet.mac.ARP arp
	mkdir drcl.inet.core.queue.FIFO queue
	mkdir drcl.inet.mac.CSMA.Mac_CSMA mac
	mkdir drcl.inet.mac.WirelessPhy wphy

	#*************
	! wphy setLEACHMode 1
```

```
connect wphy/.channelCheck@ -and mac/.wphyRadioMode@
#************

mkdir drcl.inet.mac.FreeSpaceModel propagation
mkdir drcl.inet.mac.MobilityModel mobility

set PD [mkdir drcl.inet.core.PktDispatcher      pktdispatcher]
set RT [mkdir drcl.inet.core.RT                 rt]
set ID [mkdir drcl.inet.core.Identity           id]

! pktdispatcher setRouteBackEnabled 1

$PD bind $RT
$PD bind $ID

#**************
# Here this contract is needed so we can send directly to neighbors.

connect app/.setRoute@ -to rt/.service_rt@
#*************************

connect wphy/.mobility@    -and mobility/.query@
connect wphy/.propagation@ -and propagation/.query@


#*****************************
connect mac/.sensorApp@ -and app/.macSensor@
#*****************************



connect mac/down@ -and wphy/up@
connect mac/up@   -and queue/output@

connect ll/.mac@ -and mac/.linklayer@
```

```
connect ll/down@ -and queue/up@
connect ll/.arp@ -and arp/.arp@

connect -c pktdispatcher/0@down -and ll/up@

set nid $i

! arp setAddresses  $nid $nid
! ll  setAddresses  $nid $nid
! mac setNode_num_  $nid       ;#set the MAc address

#let it know we are running CDD for SS
! mac setLEACHmode 1
! mac setMacAddress $nid  ;#set MAC
! wphy setNid       $nid
! mobility setNid   $nid
! id setDefaultID   $nid

! queue setMode      "packet"
! queue setCapacity  40

# disable ARP
! arp setBypassARP  [ expr 2>1]

connect mobility/.report@ -and /CDD/tracker/.node@
connect wphy/down@ -to /CDD/channel/.node@

! /CDD/channel attachPort $i [! wphy getPort .channel]

#                                   maxX maxY  maxZ minX  minY minZ  dX  dY   dZ
! mobility setTopologyParameters 200.0 200.0 0.0 0.0 0.0 0.0 100.0 100.0 0.0

connect -c  wireless_agent/down@ -and pktdispatcher/1111@up
```

```
	cd ..
}

#*******************************************************************
*
# FOR THE SENSORS ONLY , do the following

for {set i [expr $sink_id + 1]} {$i < [expr $node_num - $target_node_num]} {incr i}
{

	puts "create sensor $i"
	set node($i) [mkdir drcl.comp.Component n$i]

	cd n$i

	mkdir drcl.inet.sensorsim.CDD.CDDApp app
	! app setNid $i
	! app setSinkNid $sink_id
	! app setCoherentThreshold 1000.0
	! app setTargetName "Wheeled Vehicle"


	#******
	! app setNn_ [expr $node_num - 1]
	! app setNum_clusters 20
	! app setTotal_rounds 5



	# create nodes
	mkdir drcl.inet.sensorsim.SensorAgent agent

	! agent setDebugEnabled 0

	# create sensor physical layers
```

```
mkdir drcl.inet.sensorsim.SensorPhy phy
! phy setRxThresh 0.0
! phy setDebugEnabled 0

# create mobility models
mkdir drcl.inet.sensorsim.SensorMobilityModel mobility

! phy setNid $i
! phy setRadius 40.0

# connect phyiscal layers to sensor agents so that nodes can receive
connect phy/.toAgent@ -to agent/.fromPhy@

# connect sensor agent and sensor application
connect agent/.toSensorApp@ -to app/.fromSensorAgent@

# connect the sensor channel to the nodes so that they can receive
! /CDD/chan attachPort $i [! phy getPort .channel]

# connect the nodes to the propagation model
connect phy/.propagation@ -and /CDD/seismic_Prop/.query@

! mobility setNid $i

# create wireless agent layers
mkdir drcl.inet.sensorsim.CDD.WirelessLEACHAgent wireless_agent

# connect the sensor application to the wireless agent
# so that sensors can send through the wireless network protocol stack
connect app/down@ -to wireless_agent/up@

# connect the wireless agent to the sensor application
# so that sensors can receive thru the wireless network protocol stack
connect wireless_agent/.toSensorApp@ -to app/.fromWirelessAgent@
```

```
mkdir drcl.inet.mac.LL ll
mkdir drcl.inet.mac.ARP arp
mkdir drcl.inet.core.queue.FIFO queue
mkdir drcl.inet.mac.CSMA.Mac_CSMA mac
mkdir drcl.inet.mac.WirelessPhy wphy

#***************
! mac setLEACHmode 1        ;#let it know we are running CDD for SS
! wphy setLEACHMode 1       ;#let it know we are running CDD for SS
! wphy setMIT_uAMPS 1       ;#turn on MH mode settings
connect wphy/.channelCheck@ -and mac/.wphyRadioMode@

mkdir drcl.inet.mac.FreeSpaceModel propagation

set PD [mkdir drcl.inet.core.PktDispatcher     pktdispatcher]
set RT [mkdir drcl.inet.core.RT                rt]
set ID [mkdir drcl.inet.core.Identity          id]

! pktdispatcher setRouteBackEnabled 1

$PD bind $RT
$PD bind $ID

#**************
# create route configuration request for testing
#this is to define the interfaces. So in this case each sensor
#only has 1 interface (hence array size 1) and its eth0.
#another example is (which has 3 interfaces 0, 2, and 4:
#set ifs [java::new drcl.data.BitSet [java::new {int[]} 3 {0 2 4}]]

set ifs [java::new drcl.data.BitSet [java::new {int[]} 1 {0}]]
set base_entry [java::new drcl.inet.data.RTEntry $ifs]
```

```
set key [java::new drcl.inet.data.RTKey $i 0 -1]
set entry_ [!!! [$base_entry clone]]

! rt add $key $entry_

#**************
# Above we inserted a permanent entry in the
# table going to the sink. Here this contract is
# needed so we can send directly to neighbors.

connect app/.setRoute@ -to rt/.service_rt@

#*****************************
connect app/.energy@ -and wphy/.appEnergy@
mkdir drcl.inet.sensorsim.CPUAvr cpu

connect app/.cpu@ -and cpu/.reportCPUMode@
connect cpu/.battery@ -and wphy/.cpuEnergyPort@
#******************************

#******************************
connect mac/.sensorApp@ -and app/.macSensor@
#******************************

connect wphy/.mobility@    -and mobility/.query@
connect wphy/.propagation@ -and propagation/.query@

connect mac/down@ -and wphy/up@
connect mac/up@   -and queue/output@

connect ll/.mac@ -and mac/.linklayer@
connect ll/down@ -and queue/up@
connect ll/.arp@ -and arp/.arp@
```

```
        connect -c pktdispatcher/0@down -and ll/up@

        set nid $i

        ! arp setAddresses  $nid $nid
        ! ll  setAddresses  $nid $nid
        ! mac setNode_num_  $nid              ;#set the MAc address
        ! mac setMacAddress $nid          ;#same as above
        ! wphy setNid       $nid
        ! id setDefaultID   $nid

        ! queue setMode      "packet"
        ! queue setCapacity  40

        # disable ARP
        ! arp setBypassARP  [ expr 2>1]


        connect mobility/.report@ -and /CDD/tracker/.node@
        connect wphy/down@ -to /CDD/channel/.node@

        ! /CDD/channel attachPort $i [! wphy getPort .channel]

        #                           maxX maxY maxZ minX minY minZ dX dY dZ
        ! mobility setTopologyParameters 200.0 200.0 0.0 0.0 0.0 0.0 100.0 100.0 0.0


        connect -c  wireless_agent/down@ -and pktdispatcher/1111@up

        cd ..
}

#*********************************************************
# FOR THE TARGET NODES ONLY , do the following
```

```
if { $target_node_num == 0 } {
        puts "No target agents .... "
} else {
        for {set i [expr $node_num - $target_node_num]} {$i < $node_num} {incr i}
{
                puts "create target $i"

                set node$i [mkdir drcl.comp.Component n$i]

                cd n$i

                # create target agents
                mkdir drcl.inet.sensorsim.TargetAgent agent
                ! agent setBcastRate 1.0
                ! agent setSampleRate 1.0

                # create sensor physical layers
                mkdir drcl.inet.sensorsim.SensorPhy phy
                ! phy setRxThresh 0.0
                ! phy setNid $i
                ! phy setRadius 100.0

                ! phy setDebugEnabled 0

                # create mobility models
                mkdir drcl.inet.sensorsim.SensorMobilityModel mobility

                # connect target agents to phy layers so that nodes can send
                connect agent/down@ -to phy/up@

                # connect phy layers to sensor channel so that nodes can send
                connect phy/down@ -to /CDD/chan/.node@

                # connect the nodes to the propagation model
```

```
                connect phy/.propagation@ -and /CDD/seismic_Prop/.query@

                ! mobility setNid $i

                # set the topology parameters
                ! mobility setTopologyParameters 200.0 200.0 0.0 0.0 0.0 0.0

                cd ..
        }
}

#*******************************************************************
# for SENSORs and TARGETs only. Not SINKs
for {set i [expr $sink_id + 1]} {$i < $node_num} {incr i} {
        # connect the mobility model of each node to the node position tracker
        connect n$i/mobility/.report_sensor@ -and /CDD/nodetracker/.node@
        connect n$i/phy/.mobility@ -and n$i/mobility/.query@
}

#**********************************************
#Positioning
#

# set the position of sink nodes args=> (speed(m/sec), xCoord,yCoord,zCoord
! $node(0)/mobility setPosition 0.0 175.0 175.0 0.0

# for the sensors They will be randomly placed on the grid (2D only)
# set the position of sensor nodes args=> (speed(m/sec), xCoord,yCoord,zCoord
for {set i [expr $sink_id + 1]} {$i < [expr $node_num - $target_node_num]} {incr i}
{
  ! n$i/mobility setPosition 0.0 [expr rand()*200.0] [expr rand() * 200.0] 0.0
}

# for the target we can include random mobility They will be randomly
```

```
# placed on the grid (2D only)
#set the position of target nodes args=> (speed(m/sec), xCoord,yCoord,zCoord
for {set i [expr $node_num - $target_node_num]} {$i < $node_num} {incr i} {
  ! n$i/mobility setPosition 0.0 [expr rand()*80.0] [expr rand() * 80.0] 0.0
}



#*********************************************
#routeInfo() to execute do:
#         script "routeInfo" -at 0.35 -period 4.0 -on $sim
proc routeInfo { } {
  global sim n1
  puts "Current Route Table\n [! n1/rt info]"
}

#*********************************************
#Output remaining energy levels of the sensors to a plotter

set plot_ [mkdir drcl.comp.tool.Plotter .plot]
for {set i [expr $sink_id + 1]} {$i < [expr $node_num - $target_node_num]} {incr i}
{
  connect -c n$i/app/.plotter@ -to $plot_/$i@0
}

#*********************************************
#Plotters for the SINK node

#Graph # 1
#To plot the total number of received packets at the sink
#red line-> actual packets received
#blue line-> what the sink actually would have received if it
#              wasn't for CH aggregation

set sinkPlot1_ [mkdir drcl.comp.tool.Plotter .sinkPlot1]
```

```
connect -c n0/app/.PacketsReceivedPlot@ -to $sinkPlot1_/0@0
connect -c n0/app/.theo_PacketsReceivedPlot@ -to $sinkPlot1_/1@0

#Graph # 2
#Calculate the avg latency when the sink finally receives it
set sinkPlot2_ [mkdir drcl.comp.tool.Plotter .sinkPlot2]
connect -c n0/app/.latencyPlot@ -to $sinkPlot2_/0@0



#************************************************
# disp()
# this method is used to display the consumed
# energy of each sensor
proc disp { } {
global sim node_num sink_id target_node_num
for {set i [expr $sink_id + 1]} {$i < [expr $node_num - $target_node_num]} {incr i}
{
  set val [! n$i/wphy getRemainingEnergy]
  #puts " sensor-> $i remaining energy is $val"
}
}

#wsnLoop()
#
#	This method is called periodically to check
#	if the simulation should continue or not. If
#	the simulation time reaches 500, then stopped.
proc wsnLoop { } {
	global sim node_num node sink_id target_node_num

	#reset variables
	set total_packets 0
	set average_energy 0
```

```
#display statistics if simulation time is reached.
puts " Current simulation time is [! $sim getTime]"
if {[! $sim getTime]>= 500.0 } {
  $sim stop
  puts "--------------------------------------------"
  puts "Simulation Terminated\n"
  puts "Results:"



for {set i [expr $sink_id + 1]} {$i < [expr $node_num - $target_node_num]} {incr i} {
    set curr_packets [! n$i/app geteID]
    puts "Sensor$i Sent $curr_packets Packets to CH"
    set total_packets [expr $total_packets + $curr_packets]
set val [! n$i/wphy getRemainingEnergy]
set val [expr 1000.0-$val]
set average_energy [expr $average_energy + $val]

        #puts "total time Sensor$i has been cluster head is [! n$i/app get_total_times_CH]"

  }
  puts "Base Station Received [! n0/app getTotalINPackets]"
  puts "Total Packets sent from all nodes: $total_packets"
  puts "virtual packet should have been Received by Bsae Station  [! n0/app getTotalVirtualPackets]"
  puts "Success Rate: [expr ([! n0/app getTotalVirtualPackets].0 / $total_packets.0) * 100]"

  #puts "Success Rate: [expr ([! n0/app getTotalINPackets].0 / $total_packets.0) * 100]"
 puts " Average dissipated energy : [expr $average_energy / [expr $node_num - $target_node_num]]"
```

```
        puts " Average latency [! n0/app getAvgLatency]"
        }
}


#*********************************************
#sensorLocPrintOut()
#       Goes through all sensors and prints their
#       (X,Y,Z) Coordinates
proc sensorLocPrintOut { } {
        global sink_id node_num     target_node_num
        for {set i [expr $sink_id + 1]} {$i < [expr $node_num - $target_node_num]}
{incr i} {
                script [! n$i/app printNodeLoc]
        }
}

#*********************************************
puts "Simulation begins...\n"
set sim [attach_simulator .]
$sim stop


#*****************start the sink***********************
script {run n0} -at 0.001 -on $sim

#**************start the sensors**********************
for {set i [expr $sink_id + 1]} {$i < $node_num} {incr i} {
        script puts "run n$i" -at 0.1 -on $sim
}

#*********print out all the node locations*************
script "sensorLocPrintOut" -at 0.002 -on $sim
```

```
#*******Check if Sensor Status**************************
script "wsnLoop" -at 1.0 -period 2.0 -on $sim

script "disp" -at 1.0 -period 2.0 -on $sim

# Sinks subscribing to interests
# longMin longMax latMin latMax duration interval data_interval refreshPeriod)
script {! n0/app subscribe 0.0 80.0 0.0 80.0 922.0 53.0 5.0 10.0} -at 1.5 -on $sim

$sim resumeTo 100000.0
```