DEVELOPMENT OF A MODIFIED ENERGY-EFFICIENT CLUSTERING WITH SPLITTING AND MERGING FOR WIRELESS SENSOR NETWORKS USING CLUSTER-HEAD HANDOVER MECHANISM

BY

Jinadu Agbomoudu BRAIMOH, PGD. (ABU), 2013, HND. (AUP), 2009

P14EGEE8006

jinadu.braimoh@gmail.com

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE (M.Sc.) DEGREE IN TELECOMMUNICATIONS ENGINEERING

DEPARTMENT OF COMMUNICATIONS ENGINEERING

FACULTY OF ENGINEERING

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

MARCH, 2018

DECLARATION

I declare that the work in this dissertation titled **"**Development of a Modified Energy-Efficient Clustering with Splitting and Merging for Wireless Sensor Networks using Cluster-Head Handover Mechanism" has been carried out by me in the Department of Communications Engineering, Ahmadu Bello University Zaria. The information derived from literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree or diploma at this or any other institution.

Jinadu Agbomoudu BRAIMOH	_____	_____
(Student)	Signature	Date

CERTIFICATION

This dissertation titled "DEVELOPMENT OF A MODIFIED ENERGY-EFFICIENT CLUSTERING WITH SPLITTING AND MERGING FOR WIRELESS SENSOR NETWORKS USING CLUSTER-HEAD HANDOVER MECHANISM" by Jinadu Agbomoudu BRAIMOH meets the regulations governing the award of degree of Master of Science (MSc) in Telecommunications Engineering by the Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

Chairman, Supervisory Committee	_____	_____
(Prof. B. G. Bajoga)	Signature	Date
Member, Supervisory Committee	_____	_____
(Dr. A. M. S. Tekanyi)	Signature	Date
Head of Department	_____	_____
(Dr. A. D. Usman)	Signature	Date
Dean. School of Postgraduate Studies	_____	_____
(Prof. S. Z. Abubakar)	Signature	Date

DEDICATION

This research work is dedicated to God Almighty, my entire family, and friends.

ACKNOWLEDGEMENT

I wish to express my profound gratitude to God Almighty, to whom I owe all my life achievements including this academic piece.

I would like to express my deep appreciation to my supervisory committee Prof. B. G. Bajoga (Chiarman) and Dr. A. M. S. Tekanyi (Member), for their tireless efforts, valuable guidance and constant supervision towards the success of this work. The completion of this work could not have been possible without their constant participation and assistance. To all these, I ask that the grace of Allah (SWT) continue to be with them and their families forever (Amin).

I acknowledge with thanks all the lecturers of Electrical and Computer Engineering, Ahmadu Bello University namely: Prof. M. B. Mu'azu, Prof. B. Jimoh, Dr. S. M. Sani, Dr. A. D. Usman, Dr. S. Garba, Dr. Y. Jibril, Dr. T. H. Sikiru, Dr. K. A. Abu-Bilal, Dr. A. E. Adedokun, Dr. G. A. Olarinoye, Engr. M. J. Mu'azu, Engr. F. O. Sadiku, Engr. A. I. Abdullahi, Engr. Bashir Sadiq, Engr. A. Salawudeen and most especially those whose names could not be mentioned, I want to say a big thank to you all.

I am also thankful to my friends, Suleiman Magaji, Shuaib Abdul-Rasheed, Ocholi Haruna, Olayinka Wasiu Ajadi, Malachi Egbugha, Raymond Jerry, Kelechi Okogwu, Ibrahim Abdulwahab, Ajayi Ore-ofe, Oyibo Prosper, Muktar Abubakar and Aminu Abba. Their continuous support and contributions toward the success of this work would never be forgotten, may Allah reward them and strengthen our friendship. I am also very much thankful to all my course mates whose names have not been mentioned.

My appreciation goes to ETISALAT Nigeria for their practical oriented training that was granted to me during the course of my study. It is pertinent to note that the ETISALAT Telecommunications Engineering Programme (ETEP) which is part of its corporate social responsibility, contributed in no small measure to this academic success. May Almighty Allah give you the strength to continue with the program. Also, my thanks goes to the Management of Ahmadu Bello University, Zaria-Nigeria for providing the enabling environment that made the program a successful one.

I am deeply grateful to my parents Mr. & Mrs. M. Z. Braimoh for their endless love, support, and understanding; I will always be appreciative. To my siblings, thank you for being there for me. Finally, my love goes to my wife, Ummulkhair Braimoh for all the pains and for being so supportive throughout my academic pursuit.

Jinadu Agbomoudu BRAIMOH

March, 2018.

ABSTRACT

Energy efficiency is one of the most important challenges for Wireless Sensor Networks (WSNs). This is due to the fact that sensor nodes have limited energy capacity. Therefore, the energy of sensor nodes has to be efficiently managed to provide longer lifetime for the network. To reduce energy consumption in WSNs, a Modified Energy Efficient Clustering with Splitting and Merging ($mEECSM$) for WSNs using Cluster-Head Handover Mechanism was implemented in this research. This model used information of the residual energy of sensor nodes and a suitable Cluster Head (CH) handover threshold to minimize energy consumption in the network. A backup CH was incorporated into the model to take over the responsibilities of the CH once the CH handover threshold is reached. The energy consumed by node's amplifier was varied with its transmission distance. This work was carried out using MATLAB R2013a and the performance of the modified model was validated in terms of network lifetime and residual energy ratio with an existing energy reduction technique in a self-organized clustering of WSNs. Average improvements of 7.5% and 50.7% were achieved for the network lifetime and residual energy ratio respectively which indicate a significant reduction in energy consumption of the network nodes. Also, scalability and robustness test were carried out with respect to network lifetime by randomly adding and removing a number of nodes from the network. Average improvements of 7.8%, and 10.3% were achieved for scalability and robustness test respectively. The results of this work showed that $mEECSM$ has a better residual energy ratio, scalability, robustness and a longer network lifetime when compared with an existing energy reduction technique in a self-organized clustering of WSNs.

TABLE OF CONTENTS

DECLARATION **i**

CERTIFICATION **ii**

DEDICATION **iii**

ACKNOWLEDGEMENT **iv**

ABSTRACT **vi**

TABLE OF CONTENTS **vii**

LIST OF FIGURES **xii**

LIST OF TABLES **xiii**

LIST OF ABREVIATIONS **xiv**

CHAPTER ONE: INTRODUCTION

1.1 Background of the Study **1**

1.2 Significance of the Research **3**

1.3 Statement of Research Problem **3**

1.4 Aim and Objectives **4**

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction **5**

2.2 Review of Fundamental Concept **5**

2.2.1 Wireless Sensor Network (WSN) 5

2.2.1.1 Components of WSN Structure 6

2.2.1.2 Wireless Sensor Network application 7

2.2.2 Wireless Sensor Network Design Requirements 9

2.2.3 Routing in Wireless Sensor Networks 10

2.2.3.1 Flat Scheme Routing 11

2.2.3.2 Hierarchical Scheme Routing 12

2.2.3.3 Location Scheme Routing 14

2.2.4 Quality of Service Requirements in WSN 15

2.2.4.1 Network Lifetime *15*

2.2.4.2 Average Residual Energy *16*

2.2.4.3 Scalability *16*

2.2.4.4 Robustness *17*

2.2.5 Performance Evaluation 17

2.2.6 Clustering in wireless sensor network 18

2.2.6.1 Energy Efficient Self-Organized Clustering with Splitting and Merging (EECSM) 19

2.2.6.2 Definition of Some Terms 21

2.2.7 CH Handover Mechanism 22

2.2.8 Radio Energy Dissipation Model 23

2.2.9 Self-organized Clustering in WSNs 25

2.2.10 Establishing Communication between the CHs and the CMs 26

2.3 Review of Similar Works **26**

CHAPTER THREE: METHODOLOGY

3.1 Introduction **34**

3.2 Replication of the Energy Efficient self-organized Clustering with Splitting and Merging for WSNs (EECSM) 35

3.2.1 Clustering/Self-Organizing Phase 35

3.2.1.1 Broadcasting Step 38

3.2.1.2 Splitting Cluster Step 39

3.2.1.3 CH Selection Step 40

3.2.1.4 Clustering Step 40

3.2.2 Merging Cluster Phase 40

3.2.2.1 Re-Clustering Step 40

3.2.3 Data Transmission Phase 41

3.2.4 CH Backup Mechanism 41

3.3 Modified Energy-Efficient Clustering with Splitting and Merging for WSNs (*mEECSM*) 42

3.3.1 CH Handover Mechanism 43

3.3.1.1 Backup Cluster Head Selection Step 44

3.3.1.2 Determining the CH Handover Threshold 44

3.4 Establishing Communication between the CHs and the CMs 45

3.5 Calculating the Distance between CMs and CHs 46

3.6 Determining the Network Lifetime and Residual Energy Ratios 47

3.7 Performing the Scalability and Robustness Test 47

3.8 Experimental Specifications 48

CHAPTER FOUR: RESULTS AND DISCUSSIONS

4.1 Introduction 50

4.2 Results **50**

4.2.1 CH Handover Threshold 50

4.2.2 Network Lifetime Comparison 52

4.2.3 Comparison of Residual Energy Ratios 53

4.2.4 Comparison of Scalability 54

4.2.5 Comparison of Robustness 56

CHAPTER FIVE: CONCLUSION AND RECOMMENDATION

5.1 Summary **58**

5.2 Conclusion **58**

5.3 Significant Contributions **59**

5.4 Recommendations **59**

REFERENCES **60**

APPENDIX A

Result Tables 64

APPENDIX B

M-file for the Broadcasting Step 70

APPENDIX C

M-file for the Splitting Cluster Step 72

APPENDIX D

M-file for the CH Selection Step 75

APPENDIX E

M-file for the Clustering Step 76

APPENDIX F

M-file for the Merging Cluster Phase 78

APPENDIX G

M-file for the Data Transmission Phase 80

APPENDIX H

M-file for the CH Backup Mechanism 81

APPENDIX I

M-file for the Backup CH Selection Step 83

APPENDIX J

M-file for the CH Handover Threshold 86

APPENDIX K

M-file for Display of NL and RE Results 88

LIST OF FIGURES

Figure 2.1: Architecture of a Wireless Sensor Network 6

Figure 2.2: Structure of a Sensor Node 6

Figure 2.3: Clustering Architecture 19

Figure 2.4: Flowchart for EECSM 21

Figure 2.5: First Order Radio Model 24

Figure 3.1: Random Sensor Node Deployment 37

Figure 3.2: Position of the Sink and CHs at Period Zero 38

Figure 3.3: Flowchart for the Modified EECSM Model 43

Figure 3.4: Flowchart for the Communication Process between the CHs and the CMs 46

Figure 4.1: Average Network Lifetime for Different CH Handover Threshold 51

Figure 4.2: Average Residual Energy for Different CH Handover Threshold 51

Figure 4.3: Comparison of Network Lifetimes of $mEECSM$ and $EECSM$ 53

Figure 4.4: Comparison of the Residual Energy Ratios of $mEECSM$ and $EECSM$ 54

Figure 4.5: Comparison of Scalability of $mEECSM$ and $EECSM$ 56

Figure 4.6: Experiment on Sporadic Failures of Sensor Nodes 57

LIST OF TABLES

Table A1: Average Network Lifetime for Different CH Handover Threshold 64

Table A2: Average Residual Energy Ratio for Different CH Handover Threshold 65

Table A3: Average Minimum and Maximum value of Network Lifetime and Residual Energy Ratio of $mEECSM$ for 10 iterations 66

Table A4: Average Minimum and Maximum value of Network Lifetime and Residual Energy Ratio of EECSM for 10 iterations 66

Table A5: Scalability Test for EECSM 67

Table A6: Scalability Test for $mEECSM$ 68

Table A7: Experiment on Sporadic Failures for EECSM 68

Table A8: Experiment on Sporadic Failures for $mEECSM$ 69

Table A9: Summary of Results 69

LIST OF ABBREVIATIONS

Acronyms	**Definitions**
ACK	Acknowledgement
AvNL	Average Network Lifetime
AvRE	Average Residual Energy
BS	Base Station
CH	Cluster Head
CM	Cluster Member
EECSM	Energy Efficient Clustering with Splitting and Merging
$mEECSM$	modified EECSM
MAC	Media Access Control
NL	Network Lifetime
QoS	Quality of Service
RE	Residual Energy
SNR	Signal to Noise Ratio
WSN	Wireless Sensor Network

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Wireless sensor networks have recently emerged as an important means to study and interact with the physical world. The recent technological advances have made it possible to deploy small, low power, low-bandwidth, and multi-functional wireless sensor nodes to monitor and report conditions and events in their local environments (Liu & Ning, 2007; Sabet & Naji, 2015). A large collection of these sensor nodes can thus form a wireless sensor network in an ad-hoc manner, creating new types of information systems (Liu & Ning, 2007).

A sensor network is a network of large number of tiny sensor nodes and a few powerful control nodes (also called base stations). Sensor nodes are usually deployed randomly in the field and form a sensor network in an ad-hoc manner to fulfill certain tasks. There is usually no infrastructure support for sensor networks (Liu & Ning, 2007; Razaque *et al.*, 2016). Every sensor node in a sensor network has one or a few sensing components to sense conditions (such as temperature, humidity, pressure, etc.) from its immediate surroundings, a processing component to carry out simple computation on the raw data, and a communication component to communicate with its neighbor nodes (Akyildiz *et al.*, 2002; Tahir *et al.*, 2013). Instead of sending raw data to the control nodes responsible for data fusion, sensor nodes are fitted with an on-board processor with processing abilities. This enables sensor nodes to locally carry out simple computations and transmit only the required partially processed data to the control nodes. The control nodes may further process the data collected from the sensor nodes,

disseminate control commands to the sensor nodes, and connect the network to a traditional wired network (Akyildiz *et al.*, 2002; Wajgi & Thakur, 2012).

A critical issue in the design of wireless sensor networks is how to effectively utilize the limited Quality of Service (QoS) parameters. The most basic QoS parameters of sensor networks are bandwidth and energy (Mahfoudh & Minet, 2008). The efficient utilization of these resources has posed numerous challenges. The power supply unit of a sensor node is based on an energy-limited battery. Thus, most solutions elaborated for these networks are aimed at minimizing the energy consumption in order to provide longer network lifetime (Mahfoudh & Minet, 2008; Yick *et al.*, 2008).

For energy-efficient transmission of information obtained by sensor nodes, many routing protocols have been developed. Among these routing protocols, cluster-based routing protocol is proven to be the most energy efficient (Lee & Lee, 2013). In cluster-based routing, a network consists of several clusters, and each cluster is comprised of a Cluster Head (CH) and many Cluster Members (CMs) (Lee & Lee, 2013; Sabet & Naji, 2015).

Self-organizing is the ability of network nodes to detect the presence of other nodes and to organize into a structured, functioning network without human intervention (IEEE). Self-organized clustering has recently been studied intensively. Although wireless sensor nodes have limited capacities, the self-organizing method can perform adequate clustering and routing, by using only localized neighbor information and simple rules. Self-organizing clustering also has advantages, such as scalability and robustness (Lee & Lee, 2013).

Therefore, this research aimed at reducing energy consumption of a self-organized clustering of wireless sensor nodes by taking into consideration the energy level of cluster heads to

maintain load balancing. The developed model reduced the overall energy consumption of the network. The model was implemented on MATLAB R2013a simulator and simulations were carried out to validate its performance in terms of network lifetime, robustness, residual energy and scalability with the work of Lee & Lee, (2013).

1.2 Significance of Research

The significance of this research is the development of a modified energy efficient clustering with splitting and merging for wireless sensor networks using CH handover mechanism to reduce the frequent selection of next CHs at every period. The CH handover mechanism allowed for a suitable energy level threshold to be set for the CH to determine when to handover to the backup CH that is incorporated into the network. The developed model reduced the energy consumption in the network thereby improving its network lifetime and residual energy ratios as compared to the work of Lee and Lee, (2013).

1.3 Statement of Research Problem

Due to the need for long term operation of large scale wireless sensor networks, energy efficiency became a major issue in wireless sensor networks. Numerous schemes proposed in literature have shown that the problem of efficient energy management is as a result of inappropriate cluster sizes and unbalanced network loading. These proposed schemes have attempted to effectively resolve these prevailing problems. To further reduce these problems, this research work is aimed at modifying the existing energy efficient self-organized clustering with splitting and merging for WSNs by incorporating a cluster head handover mechanism to reduce the frequent selection of CHs after every transmitted data. Thus, reducing the energy consumed in transmitting and receiving control packets in the network.

1.4 Aim and Objectives

The aim of this research is to develop a modified energy-efficient clustering with splitting and merging for wireless sensor networks using CH handover mechanism. The objectives of this research are as follows:

a. To replicate and implement an Energy-Efficient self-organized Clustering with Splitting and Merging ($EECSM$) for wireless sensor networks by Lee and Lee (2013).
b. To develop a modified Energy-Efficient Clustering with Splitting and Merging ($mEECSM$) using CH handover mechanism.
c. To validate the performance of $mEECSM$ with results obtained by Lee and Lee, (2013) using Network Lifetime and Residual Energy Ratio as performance metrics.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The literature review comprises the review of fundamental concepts and similar works. The two reviews give an indication of support guiding principles, model equations, techniques and algorithms used in solving this and related problems, as well as the level to which research has reached in this subject. These reviews aided in designing a different approach to the resolution of the same problem efficiently.

2.2 Review of Fundamental Concept

In this section, concepts fundamental to this research work are reviewed. These include networks, models, principles and their mathematical equations as well as techniques and algorithms relevant to this research work.

2.2.1 Wireless Sensor Network (WSN)

A Wireless Sensor Network (WSN) is composed of several sensor nodes that have the capability of sensing and gathering data from an area of interest (Liao & Zhu, 2013). The sensor nodes can sense varying types of parameters and send to a central gateway. Wireless sensor network architecture is shown in Figure 2.1.

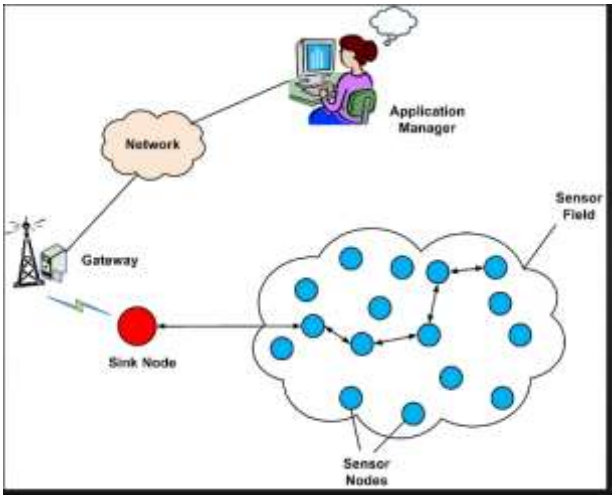


Figure 2.1: Architecture of a Wireless Sensor Network (www.cse.unr.edu, 2016)

2.2.1.1 Components of WSN Structure

The components of wireless sensor network structure are as follows:

a. **Sensor Nodes:** These are the central components that are used in WSN. They perform the basic task of the network: Sensing, computing and communicating (Sruthi & Umamakeswari, 2014). A sensor node typically contains signal-processing circuits, micro-controllers and a wireless transceiver antenna and is characterized by limited resources such as: low memory, limited battery power, and limited processing capabilities (Camilo *et al.*, 2006; Tahir *et al.*, 2013). The basic structure of a sensor node is shown in Figure 2.2.

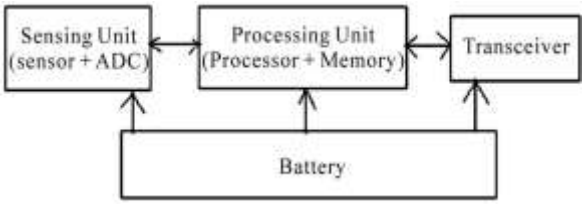


Figure 2.2: Structure of a Sensor Node (Jose & Sadashivappa, 2015)

b. **Clusters:** These are the organizational structures of WSN. Dense nodes that are deployed in the required environment are broken into simple groups called clusters for network management purpose. Power consumption is reduced by aggregating data from all nodes in the cluster and a common head of cluster called Cluster Head communicates to the Base Station. This also organizes the system in a hierarchical structure (Sruthi & Umamakeswari, 2014).

c. **Cluster Heads:** They are the organizational heads of the clusters, responsible for the management, maintenance and routing in a cluster (Sruthi & Umamakeswari, 2014).

d. **Base Station/Sink:** It is the upper most level in the hierarchical structure of WSN. Base station provides a link between the network and end user (Sruthi & Umamakeswari, 2014).

2.2.1.2 Wireless Sensor Network Application

The rapid evolution of sensor technology has led to the design of very small and smart sensors, which are used for various applications. Selection of a given type of sensor depends on the desired application. Each application of WSN has a set of requested requirements such as coverage, location, security, lifetime, etc. (Ari *et al.*, 2015). Typical application of wireless sensor network includes (Liao & Zhu, 2013; Recayte, 2012):

a. **Building automation:** WSNs are key building blocks for smart homes. The central control of lighting, heating, ventilation and air conditioning, local and remote management of home appliances, motorized blinds and curtains are examples of usage possibilities.

b. **Industrial control and monitoring:** WSNs have a potential for replacing traditional cabling in monitoring and control systems in factories. Sensor nodes can be deeply embedded into machines to monitor their health and to ensure safe operation.

c. **Military application:** In military applications, the rapid deployment of WSNs allows instant use of data. Usage scenarios of WSN data are wide ranging in intelligence, include large-scale acoustic ocean surveillance systems, detection of submarines and targeting. Furthermore, WSNs can be a part of the communication infrastructure.

d. **Medical application:** WSNs can be used in a hospital building to track and monitor physiological data of patient and all medical resources. Body sensor networks, a kind of special sensor implanted for healthcare purpose can measure blood pressure, body temperature and electrocardiograph.

e. **Traffic control:** WSNs allow easy vehicles monitoring and control of traffic conditions especially at peak times. Temporary situations such as road works and accidents can be monitored. Also, the integration of monitoring can significantly improve the city traffic management and reduce the emission of carbon dioxide.

f. **Security and surveillance:** Compared to existing wired alarm systems, WSNs allow easier deployment, adaptability, and error robustness. The increasing demand for security and alarm system is evident in homes, offices, and other public environments such as airports and factories.

g. **Environmental control:** Random deployments on large-scale areas make the monitoring of agriculture and wildlife easier. It can be used also for animal tracking, forest surveillance, flood detection, and weather forecasting. Also, WSNs may be used

in catastrophe (e.g. wildfire, earthquake, and tsunami) warning systems and in disaster relief.

2.2.2 Wireless Sensor Network Design Requirements

Standards designed for WSN are expected to satisfy some requirements which are as follows:

a. **Energy Consumption:** This requirement can be considered in two dimensions: low energy consumption and efficient energy consumption. Concerning low energy consumption, the use of low-power signals contributes to increasing the low energy consumption and ensuring a longer lifetime of the network. Concerning efficient energy consumption, the load over the network has to be balanced. Energy-aware management techniques and routing protocols are essential for load balancing and thus increasing the overall lifetime of the network (Ovsthus & Kristensen, 2014; Sabet & Naji, 2015).

b. **Self-Organizing:** WSNs are expected to be self-configuring and self-organizing. The sensor nodes may be required to be placed in strategic locations that may not be easily accessible except by using airplanes. They are thus required to organize themselves to form a network once airdropped (Ovsthus & Kristensen, 2014).

c. **Robustness/Fault-Tolerance**: WSNs are expected to be fault-tolerant and robust against failures. The network should be built such that failure of one or a few sensor nodes does not result in failure of the entire network. Thus, robust routing protocols are required which are responsive to dynamic changes in topology (Hammoudeh & Newman, 2013; Ovsthus & Kristensen, 2014).

d. **Low-delay:** Most systems are extremely delay sensitive and require the WSN communication to have predictable behavior and expect real-time guarantees (Ovsthus & Kristensen, 2014).

e. **Quality of Service (QoS):** This is defined from two perspectives; application specific QoS and network specific QoS. Application specific QoS is a higher level abstraction of QoS requirements at the application level. Minimum coverage area, minimum number of active sensors, and measurement precision could be considered as application specific QoS. Network specific QoS represents a lower level perspective at the more detailed communication part, where the QoS required by the data packets are considered. Reliability, latency, and availability are some of the major network QoS requirements (Mbowe & Oreku, 2014).

f. **Scalability:** WSN installations need to be scalable to adapt to changes in the network and allow for addition or removal of numerous sensor nodes. WSN needs to be scalable enough to accommodate new nodes without degradation in QoS. The self-organizing requirement is also essential for scalability (Hammoudeh & Newman, 2013; Ovsthus & Kristensen, 2014).

g. **Data Aggregation.** Sensors sense data continuously and this consecutively sensed data may be redundant. Also, sensors deployed in a particular area might sense similar or identical data. The importance of this sensed data depends on the application requirements, and for some applications it might be sufficient to get aggregated results from these sensors. Data aggregation can either be of data from the same sensor or from a group of sensors. Data aggregation increases energy efficiency by minimizing the number of packets sent. This is more of an application layer function (Ovsthus & Kristensen, 2014).

2.2.3 Routing in Wireless Sensor Networks

To establish the communication between sensor nodes, routing becomes an essential mechanism. Routing in wired networks is different from that implemented in wireless sensor network as a result of challenges such as limited storage, energy, bandwidth, etc. in wireless sensor network (Vinothini & Umamakeswari, 2014).

Though sensor networks are basically used for sensing and communicating events, they however, depend on the application. So, a single routing algorithm cannot work efficiently for sensor networks for all applications and energy awareness is a major issue in routing protocol (Vinothini & Umamakeswari, 2014). Based on the network structure, routing algorithm in WSN is classified into three types:

a. Flat routing scheme

b. Hierarchical routing scheme

c. Location routing scheme

2.2.3.1 Flat Routing Scheme

In flat routing scheme, each and every node has equal preference. Data centric algorithm comes under the flat based routing. Query message is transferred to the sensor nodes to establish a connection. The base station sends in a query to a region of interest and depending on the data request the nodes reply via data packets. Sensor Protocols for Information via Negotiation (SPIN) was developed as the first data centric protocol. It eliminates redundant data by considering the data negotiation between groups of nodes, thus, saving energy (Lai *et al.*, 2012; Vinothini & Umamakeswari, 2014). Important protocols in this class are Sequential Assignment Routing (SAR) and Directed diffusion. Important characteristic of flat based

routing is that it is mostly contention based MAC (Ovsthus & Kristensen, 2014). However, flat routing protocol can cause sensor nodes to overload with an increase in the number of sensor nodes and does not offer scalability.

2.2.3.2 Hierarchical Routing Scheme

The hierarchical routing protocol considers energy consumption as a major issue in Wireless Sensor Network (WSN). Several clusters can be formed based on this technique. To implement the routing mechanism, the sensor network is grouped to form clusters. A Cluster Head (CH) is assigned for all clusters. CHs play the role as an initiator of the group. The CH has various responsibilities such as data aggregation, collection, and also reduction of the number of communications to sink which help to increase the lifetime of the network (Lai *et al.*, 2012; Vinothini & Umamakeswari, 2014). Some of the hierarchical protocols are:

a. **Low-Energy Adaptive Clustering Hierarchy (LEACH):** LEACH is an important hierarchal routing protocol whose basic idea is to form clusters. To protect the battery life of a single node, LEACH uses a normalization technique that allows cluster head to be randomly selected. Here, CH also does aggregation of data to reduce the number of communications to Base Station (BS) (Vinothini & Umamakeswari, 2014). The CHs are changed periodically in order to balance the energy of sensor nodes. However, the protocol forces every node to become a CH once regardless of the energy level of the node. Since probability is also used here, sometimes there is a chance that node is left out of becoming a CH (Razaque *et al.*, 2016). In LEACH protocol, nodes use high power for transmitting data to CHs or to a BS. Transmitting at high-power level may lead to interference and energy wastage (Dahnil *et al.*, 2012).

b. **Power-Efficient Gathering in Sensor Information Systems (PEGASIS**): PEGASIS is an enhanced version of the LEACH algorithm. As an alternative for multiple clustering, this algorithm form chain of nodes, so that every node communicates to a neighbor and only one node is responsible for communication with the BS (sink). Communicated data gets aggregated at the CH and gets communicated to BS. This type of chain of nodes construction is considered in a greedy way (Vinothini & Umamakeswari, 2014). PEGASIS is a greedy chain protocol that is near optimal for a data-gathering problem in sensor networks. The PEGASIS greedy approach considers the physical distance of CH from the BS only, ignoring the capability of a prospective node on the chain. Thus, a node with a shorter distance and with less residual energy may be selected in the chain and may die quickly (Ganesh & Amutha, 2013).

c. **Threshold Sensitive Energy Efficient sensor Network (TEEN):** TEEN is a hierarchical protocol proposed for time-critical applications. It is an event-driven protocol aimed at event-based applications. Once the clusters are created and a CH has been selected, the CH broadcasts two thresholds to its cluster member nodes. The two thresholds are related to the value of the data and are referred to as soft threshold and hard threshold. (Ovsthus & Kristensen, 2014). Soft threshold is the lowest set level which the value of a sensed data must exceed before the data can be forwarded to the CH, while the highest level which the value of a sensed data must exceed before the sensed data can be forwarded to the BS, is the hard threshold (Vinothini & Umamakeswari, 2014). When the sensed data of a node exceeds the soft threshold, the data is sent to the CH which relays it to the sink. When the hard threshold is exceeded the sensor node directly relays the data to the sink. The CHs are changed periodically

for load balancing (Ovsthus & Kristensen, 2014). In TEEN, periodic data collection is not designed in the protocol. Thus, a hybrid model which is suitable for both periodic data-collections and time-critical events was proposed, the protocol is named Adaptive TEEN (APTEEN) (Ovsthus & Kristensen, 2014).

d. **Hybrid Energy-Efficient Distributed Clustering (HEED):** HEED is a multiple-hop clustering algorithm for WSN and aims at efficient clustering and efficient CH selection (Vinothini & Umamakeswari, 2014) by two parameters: The first parameter is the residual energy of the sensor nodes. The second parameter is the intra-cluster communication cost to solve break-ties. The intra cluster communication costs are minimum degree cost, maximum degree cost, and Average Minimum Reachability Power (AMRP). The HEED prolongs the network lifetime longer than that of the generalized LEACH (Lee & Lee, 2013). HEED has a basic assumption that sensor nodes have multiple transmission power levels, which is not always true. It has a hybrid approach in terms of CH selection, which considers both residual energy level and communication cost (Ovsthus & Kristensen, 2014). However, HEED is still heuristic in nature and suffers a high network delay due to the complexity of the CH selection algorithm (Hammoudeh & Newman, 2013).

2.2.3.3 Location Routing Scheme

Location information is the basic requirement for this kind of routing protocol. This calculates the node's distance by means of which energy consumption is predicted. To sense location, queries are diffused to the particular region instead of transmitting all over the network. It is mainly designed for Ad-hoc networks. Location scheme routing are not energy aware (Vinothini & Umamakeswari, 2014). Examples of the location aware protocols are:

a. Geographic Adaptive Fidelity (GAF).
b. Minimum Energy Communication Network (MECN).
c. Geographic Energy Aware Routing (GEAR).

All these routing protocols reviewed randomly select a few sensor nodes as CHs and rotate this role to evenly distribute the energy load among the sensors in the network. However, none of these protocols is able to effectively perform a self-organized clustering with splitting and merging.

2.2.4 Quality of Service Requirements in WSN

There is no formal definition of Quality of Service (QoS) term. Conceptually, it can be regarded as the capability to provide assurance that the service requirements of applications can be satisfied. Depending on the type of target application, QoS of WSNs can be characterized by reliability, timeliness, robustness, scalability, network lifetime, availability, and security, among others. Some QoS parameters are used to measure the degree of satisfaction of these services, such as throughput, delay, jitter, packet loss rate and others (Mbowe and Oreku, (2014).

2.2.4.1 Network Lifetime

Network lifetime strongly depends on the lifetimes of the single nodes that constitute the network. The lifetime of a sensor node depends basically on two factors: how much energy it consumes over time, and how much energy is available for its use (Dietrich & Dressler, 2009). Network lifetime has been discussed from different points of view, which led to the development of various lifetime metrics depending on the energy consumers regarded in each metric and the specific application requirements considered (Dietrich & Dressler, 2009).

Network lifetime is the period until a certain number of sensor nodes are all discharged of their energy (Lee & Lee, 2013). A period until 30 sensor nodes are discharged of their energy as defined by Lee & Lee, (2013), is adopted in this research work.

2.2.4.2 Average Residual Energy

The residual energy level is defined as the average of the remaining energy level of the sensor nodes at the end of each simulation experiment (Torkestani, 2015). Residual energy of node i in time t is defined by omitting consumed energy from the initial battery power (Kamyabpour & Hoang, 2011):

$$E_{residual,i}\,(t) = E_{initial,i}\,(t) - E_{consumed,i}(t) \tag{2.1}$$

Also, the percentage ratio of the residual energy of sensor node is calculated as follows:

$$\%\, E_{residual}\, Ratio(t) = \frac{sum\ of\ E_{current}\ of\ nodes}{sum\ of\ E_{initial}\ of\ nodes}\ x\ 100\% \tag{2.2}$$

where $E_{current}$ is the current energy of the nodes.

The residual energy varies based on parameters such as sensing, transmission radius, and routing methods, since these parameters have direct effect on the overall energy consumption (Kamyabpour & Hoang, 2011).

2.2.4.3 Scalability

WSN installations need to be scalable to adapt to changes in the network and allow for addition or removal of numerous sensor nodes. The self-organizing requirement is essential for scalability (Ovsthus & Kristensen, 2014). When new nodes are added to the network, the lifetime of the network increases and the amount of increment determine the level of scalability

of the network. Scalability measurement in this regard, is done based on the network lifetime as implemented by Lee & Lee, (2013), by adding a certain number of nodes to the network and comparing results. Assuming uniform energy levels at initial deployment, sensing, data rate and radius, a longer network lifetime indicates lesser energy consumption in the network.

2.2.4.4 Robustness

This is the ability of the network to continue its operation after a sudden breakdown of several nodes with little or no significant effect on the network lifetime. The robustness of the network also comes from the presence of backup CHs which ensure that communication remains possible in spite of damage to CHs. Robustness in this regard in measured relative to the network lifetime by systematically removing some nodes from the network and comparing results. That is, by assuming a sudden breakdown of some nodes in the network.

2.2.5 Performance Evaluation

The performance of the modified model ($mEECSM$) is validated with the work of Lee and Lee, (2013) using scalability, robustness, network lifetime and residual energies of sensor nodes as performance metrics. These performance metrics are explained in section (2.2.4). Simulations were carried out using MATLAB R2013a. The following equations were used for performance evaluation of $mEECSM$:

a. The percentage network lifetime improvement is given as:

$$\%\ NL\ improvement = \frac{mEECSM\ NL - EECSM\ NL}{EECSM\ NL}\ x\ 100\% \qquad (2.3)$$

where:

NL is Network Lifetime

$mEECSM\ NL$ is the network lifetime of $mEECSM$ until a node is discharged

$EECSM\ NL$ is the network lifetime of $EECSM$ until a node is discharged of its energy.

b. The percentage ratio improvement is given as:

$$RE = \frac{AvRE\ of\ mEECSM}{AvRE\ of\ EECSM}\ x\ 100\% \quad (2.4)$$

where:

RE is residual energy ratio

$AvRE$ is average ratio of residual energy

c. The percentage scalability improvement is given as:

$$\%\ scalability\ improvement = \frac{Scalable_NewNL - NL}{NL}\ x\ 100\% \quad (2.5)$$

where:

$Scalable_NewNL'$ is the network lifetime when new nodes are added to the network

NL is the network lifetime when 100 nodes are initially deployed in the field.

d. The percentage robustness improvement is given as:

$$\%\ robustness\ improvement = \frac{Robust_NewNL - NL}{NL}\ x\ 100\% \quad (2.6)$$

where:

$Robust_NewNL$ is the network lifetime when 50% of the sensor nodes have malfunctioned with or without sudden breakdown.

2.2.6 Clustering in Wireless Sensor Network

Clustering involves dividing node deployment area into smaller geographical regions called clusters. Cluster size depends upon communication capability of the radios of a WSN nodes. Note that, as cluster size increases, the nodes' transmission range reduces and thus, the overall energy consumption of the network decreases. But packet delivery ratio can become better or worse depending on the number of hops that the packet needs to transverse before getting to the Base Station (BS) (Thakkar & Kotecha, 2015). The clustering concept offers tremendous benefits for wireless sensor networks because it reduces the size of the routing table, conserves communication bandwidth, prolongs network lifetime, decreases the redundancy of data packets, reduces the rate of energy consumption, etc. (Wajgi & Thakur, 2012).

In a cluster network, each cluster has a Cluster Head (CH) and many Cluster Members (CMs). The CH nodes aggregate received data from the CMs and send it to the sink (Ganesh & Amutha, 2013). A typical example of a cluster architecture is depicted in Figure 2.3.

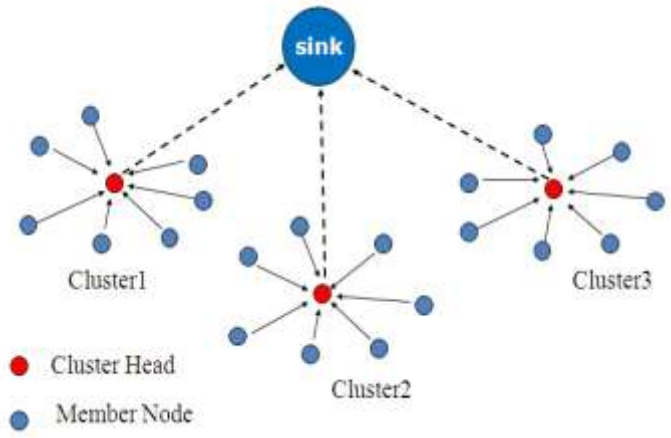


Figure 2.3: Clustering Architecture (www.ejaet.com, 2014)

2.2.6.1 Energy Efficient Self-Organized Clustering with Splitting and Merging (EECSM)

The usual cluster-based routing protocols do not guarantee proper clustering size, thus forcing the CMs to send data packets to a distant CH when the size of a cluster is too large. This gives rise to high energy consumption, transmission delays, and bottlenecks. In contrast, when the size of clusters is too small, the number of clusters is increased, creating an increase in the number of CHs, which consume much more energy when compared with CMs sending data directly to BS (Lee & Lee, 2013). The size of clusters should be properly adjusted to maximize energy efficiency.

To properly adjust the size of clusters and the number of clusters, Energy Efficient Self-organized Clustering with Splitting and Merging (EECSM) for wireless sensor networks was developed by Lee and Lee, (2013), which has been modified and adopted in this research. EECSM merges small clusters into other clusters and splits each large cluster into two or more clusters depending on the splitting and merging threshold.

However, the CH backup mechanism employed by Lee and Lee (2013) requires that whenever the CH fails, the CMs participate in selecting a new CH by exchanging information about their energy levels after which re-clustering takes place. This process consumed extra energy. Also, CHs that were selected after every data transmission resulted to frequent exchange of control packets between the CMs and the CHs which also consumed extra energy. The EECSM flowchart shown in Figure 2.4 is modified for the approach used in this research work.

a. **The Splitting and Merging Threshold:** Experiments were carried out by Lee and Lee (2013) to find the optimal values of the splitting threshold value and the merging threshold value. This was done in order to avoid having a very large cluster and a very small cluster that would consume much more energy. It was concluded from these

experiments that the optimal value of the splitting threshold value was 46, and the optimal value of the merging threshold value was 20. Therefore, EECSM splits a cluster that has more than 46 CMs into two clusters and merged a cluster that has less than 20 CMs into other clusters (Lee & Lee, 2013).

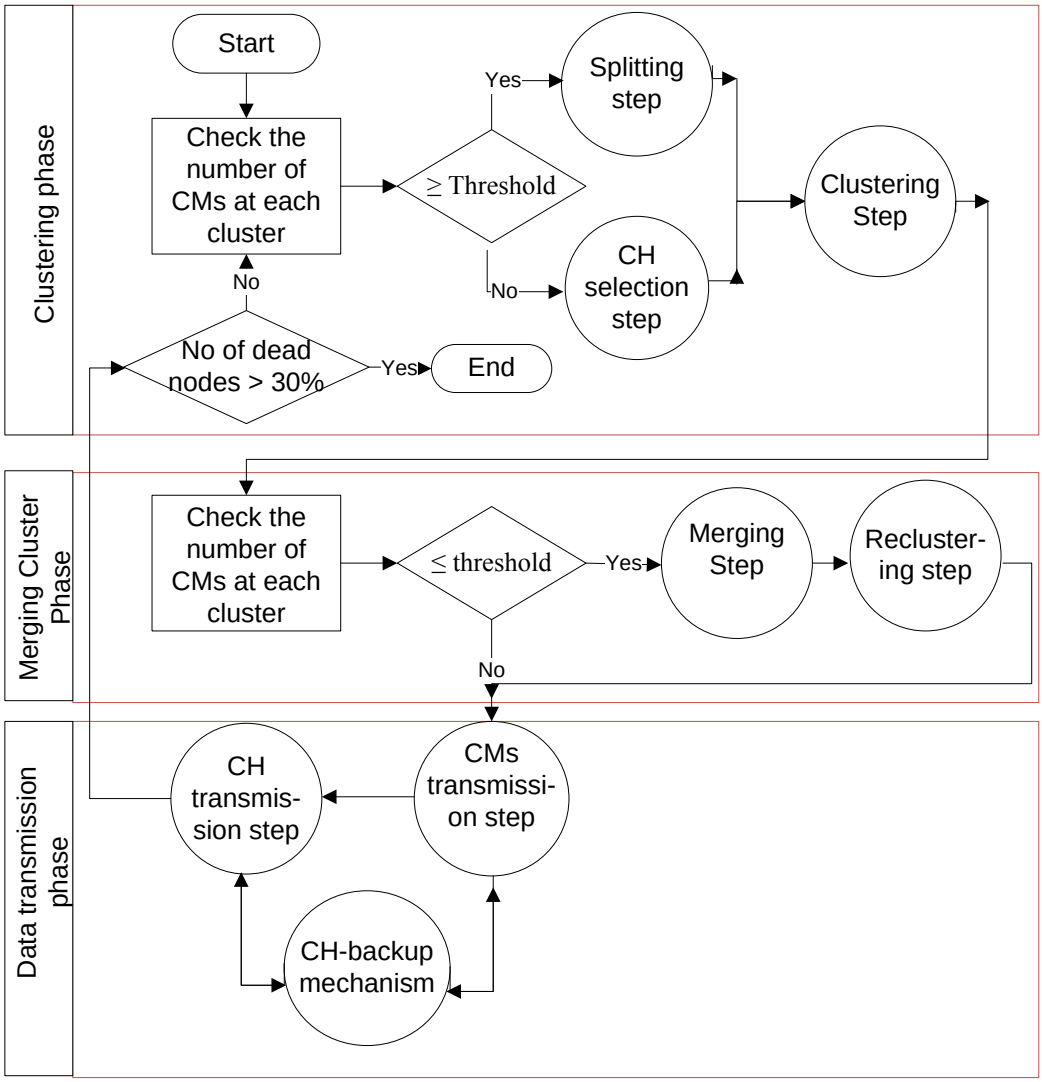


Figure 2.4: Flowchart for EECSM (Lee & Lee, 2013).

2.2.6.2 Definition of Some Terms

Some of the terms used in the course of this research are as defined:

a. **CH-signal:** This is the signal broadcasted by a CH to announce itself to nodes in the network.
b. **CH-failure Signal:** This is the signal broadcasted by either a CH or a CM to announce the failure of a CH. This is the signal that helps to activate the backup CH.
c. **Request to join Signal:** This is the signal broadcasted by the CMs that want to join a cluster or connect to a CH.
d. **Undecided State:** This is the state of the nodes when deciding on whether to be CH or CM. This usually happens during initial deployment. The undecided state signal is the signal broadcasted by those nodes.
e. **CH Handover Threshold:** This is the defined energy threshold suitable for the CH to perform handover to the backup CH. That is, a CH can only handover to the backup CH once it has used this defined percentage of its energy level.
f. **Merging Cluster Signal:** This is the signal broadcasted by the CH that has a number of CMs less than the merging threshold. When a node receives this signal from its CH, it starts looking for another CH to connect to by broadcasting the '*request to join'* signal.
g. **Period:** In the data transmission phase, data packets are transmitted from all CMs to the BS. A cycle of this process is defined as a "period." (Lee & Lee, 2013).
h. **Clustering Round:** This is the number of repeated periods during a data transmission phase (Lee & Lee, 2013).

2.2.7 CH Handover Mechanism

This is a technique through which a CH hands over its responsibilities to the backup CH when its energy level reaches a predefined threshold. All the CMs in the cluster send information about their residual energy levels to the CH when requested. This request is sent only when the

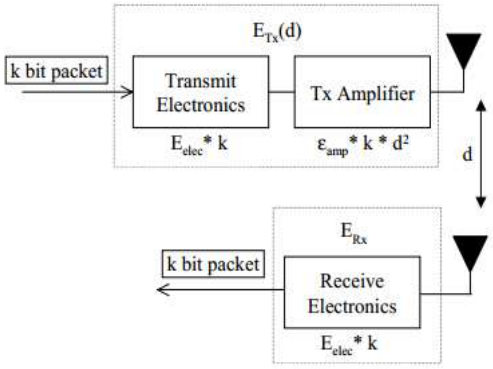
CH has been selected. Once replies are received by the selected CH, it decides on the backup CH which is the node with the highest energy level and put it to sleep. When the CH handover threshold is reached, the CH transmits '*CH-failure signal'* to the backup CH which then acts as the CH, broadcast '*CH-signal'* and selects its own backup CH. The CH that reached its set energy threshold then becomes a CM. A '*CH-handover threshold*' is set for any node that became a CH. The CH handover mechanism is implemented in Figure 3.3.

This technique is adopted to limits the frequent selection of CHs after every transmitted data and thus, a reduction in the number of control messages in the network which in turn, improve network lifetime. The remaining energy of nodes is used by the CH in selecting the backup CH. The selected backup CH is usually the node with the highest residual energy level in the cluster. The backup CH is used when the CH reaches its energy threshold level or fails unexpectedly due to factors such as human activities or mechanical defects. All sensor nodes in the network only participate in CH selection at initial deployment after which the CH decides its backup CH.

2.2.8 Radio Energy Dissipation Model

It is assumed that constant amount of energy is consumed in internal processing of a packet, whereas the energy consumed in amplifying the signal to achieve acceptable signal to noise ratio at a receiver is proportional to the square of the distance between transmitter and the intended receiver (Liao & Zhu, 2013; Patel *et al.*, 2004). Therefore, the sensor node transmission energy consumption is modeled as the sum of constant electronic components energy consumption and an amplifier energy proportional to the receiver distance (Meghji & Habibi, 2011).

From Figure 2.5, it is assumed that the radio channel is symmetrical such that the energy required to transmit a message from node A to node B is the same as the energy required to transmit a message from node B to node A for a given Signal to Noise Ratio (SNR).



.

Figure 2.5: First Order Radio Model (Heinzelman et al., 2000)

Thus, the energy required for a node to transmit a packet of length k bits over a distance d is (Liao & Zhu, 2013):

$$E_{Tx}(k, d) = (E_{elec} + \varepsilon_{amp} d^2)k \qquad : d < d_{xover} \tag{2.7}$$

While the energy consumed at the receiving node is (Liao & Zhu, 2013):

$$E_{Rx}(k) = E_{elec} \times k \tag{2.8}$$

where:

$E_{Tx}(k, d)$ is the energy consumption in transmitting k bits data to a node with a distance of d.

$E_{Rx}(k)$ is the energy consumption in receiving k bits data.

E_{elec} is the per bit energy consumption for transmitter and receiver circuitry.

ε_{amp} is the per bit energy consumption by the node's amplifier.

2.2.9 Self-Organizing Clustering in WSNs

In self-organizing clustering, the sensor nodes are to organize themselves to form clusters of bounded sizes on their own after deployment. A self-organizing clustering network maintains the integrity of the network when any topological changes happen due to new nodes joining or existing nodes leaving the network. In a self-organizing network, new nodes are deployed to increase the coverage of the network. Moreover, existing nodes may leave the network due to hardware malfunctioning, environment condition, and low battery power. The network is able to withstand these changes and still maintain its integrity (Zeb *et al.*, 2015). The advantages of using a self-organizing wireless networks include: scalability, complete distributed control, capability to adapt to changing environmental conditions, problems prevention due to damage, or failure of individual elements, etc. (Lee & Lee, 2012).

This research work addressed the three (3) major categories of a self-organizing networks. These are: self-configuration, self-optimization and self-healing (Østerbø & Grøndalen, 2012).

a. **Self-Configuration:** is the ability of the network to generate adequate formations depending on the current situation in terms of environmental circumstances, e.g. connectivity, quality of service parameters (Dressler, 2006). To address the self-configuration aspect of the network, the sensor field is automatically formed into clusters at initial deployment with each cluster having a CH and many CMs. New nodes joining the network connect to any CH of their choice based on set parameters.
b. **Self-Optimization:** is the ability of a network to maintain high quality performance and to automatically adjust network parameters with minimum manual intervention from the operator to keep the network optimized at all times (Østerbø & Grøndalen,

2012). Self-optimization focuses on the optimal choice of methods and their parameters based on the system behavior (Dressler, 2006). To address the self-optimization aspect of this research work, nodes with highest energy level are selected as CHs and every node in the network is allowed to join the CH closer to them in order to minimize energy consumption.

c. **Self-Healing:** is the ability of the network to detect, and recover from faults appearing in either network nodes or communication links, without human intervention (Dressler, 2006). To address the self-healing aspect of this work, a backup CH is dynamically activated such that the breakdown of a CH does not significantly disrupt the network.

2.2.10 Establishing Communication between the CHs and the CMs

The communicating nodes of a sensor network are linked by a wireless medium. These links can be formed by radio frequency, Bluetooth, infrared, or optical media. In this research, nodes were allowed to choose the CH closer to them to reduce energy consumption in the network. Nodes closer to CHs transmitted at a reduced range than the nodes farther away from the CHs. Therefore, the energy consumed by a node during data transmission is varied with the distance between the node and the receiver. The higher the transmission distance, the higher the energy consumed for amplification of the transmitted signal (Dahnil *et al.*, 2012). This is shown in equation (2.7).

2.3 Review of Similar Works

Some literature relevant to the subject area of clustering in wireless sensor network are discussed in this section. The section explains the different approaches that were proposed and

implemented by other researchers to deal with energy constraint in wireless sensor networks. Some employed self-organized clustering techniques in the resolution of sensor nodes power consumption problem.

Wajgi and Thakur, (2012) proposed a clustering technique that balanced load among CHs using backup nodes. The backup nodes had high energy and processing power and were meant to replace the CHs when their energy level reach a limit. The performance of the algorithm was compared with LEACH algorithm and was implemented using MATLAB. Simulation results established that the proposed approach was effective in prolonging the network lifetime with respect to the number of dead nodes. The proposed clustering technique emphasized that nodes with high initial energy level and processing power be selected and deployed close to each other as cluster heads and backup nodes, respectively. This ensures a close distance between them so that when the CH fails, nodes will be able to communicate with the backup nodes with a slight difference in the transmission radius. However, systematically placing nodes with high energy level and processing power in a large and remote network is time consuming and a challenging task.

Dahnil *et al.*, (2012) employed a topology-controlled adaptive clustering protocol to increase the lifetime of WSNs and maintain a required network connectivity. The proposed scheme allowed CHs to adjust their power level to achieve optimal degree of connectivity and maintain this value throughout the network on every periodic update. The average energy of all the nodes was aggregated by their respective CHs and this aggregated information was sent to other CHs in the network on every round so that an updated information about the number of nodes in each cluster can be stored. Simulation results showed that the proposed clustering algorithm maintained the required degree for inter-cluster connectivity on many rounds compared to

Hybrid Energy-Efficient Distributed clustering (HEED), Energy-Efficient Clustering Scheme (EECS), and Low-Energy Adaptive Clustering Hierarchy (LEACH). However, the exchange of information about energy levels of nodes in each cluster by CHs led to increased overhead in the network. This consumed energy as much control information had to be exchanged.

Lai *et al.*, (2012) implemented a cluster-based routing protocol called ‘‘Arranging Cluster sizes and Transmission ranges for wireless sensor networks (ACT)’’ to reduce the number of nodes in clusters close to the BS since CHs closer to the BS relayed more data from other CHs than CHs far away from the BS. The proposed protocol allowed every CH to consume approximately the same amount of energy so that the CHs closer to the BS did not exhaust their power faster than others. ACT determined the size of a cluster for each CH according to the distance between the cluster and the BS. Simulations were carried out using Network Simulator 2 (NS-2). The results obtained showed that the proposed routing protocol effectively improved the network lifetime and reduced the energy consumption of CHs around the BS as compared to Low Energy Adaptive Clustering Hierarchy (LEACH), Base Station Controlled Dynamic Clustering Protocol (BCDCP), and Multi-hop Routing with LEACH (MR-LEACH). ACT tried to evenly distribute the transmission capacity (data) of CHs far from the BS to the BS through the CHs closer to the BS. The transmission capacity was divided until it was equally shared among CHs close to the BS. This sharing of data among CHs caused increased control overhead in the network. Much energy was expelled for data traversing multiple CHs before getting to the BS in order to balance the energy of the network, even when this data can be delivered directly by CHs that are not too far away.

Liao and Zhu, (2013) proposed an improved clustering algorithm based on LEACH, that considered node’s residual energy and location information in selecting CHs. Normal nodes

selected the optimal CH based on some cost function which included taking into account their distances from the base station, energy level and location information. Simulation results showed that the algorithm outperformed LEACH in balancing nodes' energy consumption, improving the efficiency of data transmission and prolonging the network lifetime. However, getting location information from CHs consumed energy.

Lee and Lee, (2013) proposed an Energy Efficient Self-organized Clustering model with Splitting and Merging (EECSM) model for wireless sensor networks. EECSM performed splitting and merging of clusters using information of the energy state of sensor nodes to select a CH and to maintain load balancing. The performance of EECSM in terms of energy consumption, scalability, robustness, residual energy and network lifetime was compared with Hybrid Energy Efficient Distributed Clustering (HEED) using Visual C++ of Visual Studio 2008. Simulation results showed that EECSM significantly outperformed HEED. However, CHs that were selected after every data transmission resulted to frequent exchange of control packets between the CMs and the CHs. This consumed extra energy. Also, the CH backup mechanism employed requires that whenever the CH fails, the CMs participate in selecting a new CH by exchanging information about their energy levels after which re-clustering takes place. This process also consumed extra energy.

Tahir *et al.*, (2013) proposed an Energy-Efficient Adaptive Scheme for data Transmission (EAST) in Wireless Sensor Networks (WSNs). The scheme was modelled to achieve high Signal to Noise Ratio (SNR) at the base station irrespective of the varying effect of temperature on the transmission power of sensor nodes. The relationship between RSSI and temperature was analyzed for the scheme (EAST). Two techniques were used to regulate the transmission power of nodes that varied with temperature; the open-loop feedback process was used for link

quality estimation and compensation, and the closed-loop feedback process was used to minimize overhead of control packets by dividing the network into three logical regions. Simulation results obtained showed that the proposed scheme effectively adapted transmission power level to changing link quality, with less control overhead and energy consumption as compared to classical approach with single region where maximum transmitter level was used. However, if the network is relatively unstable, link quality changes continuously and the initial phase of sending beacon message periodically to all nodes will still be repeated thus, generating much control overhead in the network which consumed energy.

Ganesh and Amutha, (2013) worked on an Efficient and Secure Routing Protocol for wireless sensor networks using SNR-based Dynamic Clustering (ESRPSDC) mechanisms. Ad-hoc On Demand distance Vector (AODV) routing was modified in their work by incorporating Signal-to-Noise Ratio (SNR) based dynamic clustering which partitioned the network into clusters and selected Cluster Heads (CHs) among nodes based on their energy level. The proposed scheme allowed non-CH nodes to connect to a specific CH based on the received SNR values. Error recovery was implemented on the network during inter-cluster routing. Simulations were carried out using the Global Mobile Simulator (GloMoSim-2.03) and the results showed that ESRPSDC (which is a hybrid ESRP) significantly improved the energy efficiency and packet reception rate as compared with the SNR unaware routing algorithms such as the Low Energy Adaptive Clustering Hierarchy (LEACH) and Power Efficient Gathering in Sensor Information Systems (PEGASIS). In this approach, nodes were divided equally into clusters. However, the concept of dividing nodes equally into clusters indicates that a node can connect to a farther CH in order to balance the number of nodes in a cluster when a particular CH is occupied. This approach consumed energy.

Hammoudeh and Newman, (2013) implemented a cluster-based Route Optimization and Load-balancing protocol (ROL) aimed at prolonging network lifetime, providing timely message delivery and improving network robustness. Nutrient-flow-based Distributed Clustering (NDC), an algorithm for achieving load balancing by equalizing the diameter and the membership of clusters was also proposed. ROL improved on the robustness of LEACH by ensuring that each node learned multiple paths to its Cluster Head (CH). Energy expenditure was reduced by shortening the distance between nodes and CHs which was achieved, using a hop count metric in addition to transmission back-off delay. Simulation results showed that ROL/NDC maintained a maximum of 7% variation from the optimal cluster population, reduced the total number of set-up messages by up to 60%, reduced the end-to-end delay by up to 56%, and enhanced the data delivery ratio by up to 0.98% compared to LEACH. However, the hop count metrics introduced, allowed a message to transverse a smaller hop count even when there is high traffic on such links, thus, making these hops to discharge rapidly. In the case of a damage to a CH, the nodes were made to transmit directly to the sink until a new cluster round is due. These nodes dissipated much energy in order to reach the faraway sink.

Mir *et al.*, (2014) worked on a Two-Tiered Topology Control (TTTC) protocol that combined clustering and power control approaches towards topology control of wireless networks. TTTC operation was divided into two phases. In the first phase, a clustering algorithm was executed to obtain clusters of varying sizes. In the second phase, each CH ran a local Minimum Spanning Tree (MST) based power control algorithm that helped to control the transmission power of cluster members. Performance evaluation of the proposed TTTC framework with Minimum Connected Domination Set (MCDS) algorithm and Minimal Virtual Connected Dominating Set (MVCDS) algorithm was conducted with Network Simulator-2 (NS-2) using various values of

virtual range. The results showed that TTTC offered versatile performance in terms of energy cost, hop count, and nodal degree while maintaining network-wide connectivity. However, the minimum and maximum number of CMs in a cluster was not defined which could lead to a CH having too many CMs. This causes the CH to discharge faster.

Castagnetti *et al.*, (2014) implemented a global power management approach for energy harvesting sensor nodes that utilized a joint duty-cycle optimization and transmission power control. Duty-cycle management, dealt with the control of task activation rate where a node followed a sleep-wake up cycle in order to balance the energy that is harvested and the energy that is consumed. Transmission power control dealt with the RF transmission power adjustment of the node for quality packet reception at the base station. A waveform-level RF simulator was developed to model the wireless channel and the node communication hardware. Simulation results showed a 15% improvement in energy efficiency and Packet Reception Ratio (PPR) with respect to a fixed transmission power configuration. However, a situation where the sleep and wake-up periods of the nodes are controlled by the state of charge of the battery, rendered the nodes incapable of taking measurements at times when there is need to take measurements.

Sabet and Naji, (2015) proposed a Decentralized energy-efficient Hierarchical Cluster-based Routing (DHCR) algorithm for wireless sensor networks to reduce the energy consumption caused by extra control message transmissions. A routing tree was constructed such that a Base Station (BS) was located at the root, a Cluster Head (CH) at each edge and cluster member nodes at the leaves. CHs were selected based on the distance from BS and the remaining energy of sensor nodes. DHCR associated cluster members to each CH by considering energy, distance, and density metrics. Simulation results showed that the proposed algorithm was more energy efficient and provided longer lifetime compared to Low Energy Adaptive Clustering Hierarchy

(LEACH) and Hybrid Energy-Efficient Distributed (HEED) clustering algorithms. However, shorter routes which were used by CHs to reach the BS led to congestion in the intermediate CHs. In their work, a node can only be CH once till it is discharged of its energy before another node is selected in the cluster to be CH.

Razaque *et al.*, (2016) employed a Hybrid Low Energy Adaptive Clustering (H-LEACH) protocol that combined Hybrid Energy Efficient Distributed clustering (HEED) and LEACH for CH election. H-LEACH considered residual and maximum energy of nodes for every round when selecting a CH so as to balance the energy consumption of nodes. H-LEACH was compared with LEACH and HEED protocol using MATLAB as the simulation tool. Simulation results proved H-LEACH to be more energy efficient than both LEACH and HEED. However, the authors only considered a heterogeneous network, where sensor nodes are assumed to have different energy levels at initial deployment. Also, the total number of alive nodes which was calculated for every network round, consumed the network processing time as more control messages had to be processed.

It is evident from the literature reviewed that one of the ways of improving the Quality of Service (QoS) of wireless sensor networks is by improving the network availability through reducing energy consumption. This has been given significant research attention leading to the development of improved algorithms and routing protocols in this area. This research work is an extension of Lee and Lee, (2013)'s work, that proposed an energy efficient self-organized clustering with splitting and merging model for WSNs. Developing an improvement of this model to efficiently manage energy consumption by taking into consideration the residual energy level of CHs to perform handover, is the basis for the motivation of this research.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

This section describes the detailed procedure carried out in performing and modeling: node deployment, sink node deployment, clustering, cluster merging, backup CH selection, CH handover mechanism, data transmission, and finally, performing the simulation procedure for network lifetime, residual energy, scalability and robustness test. The materials used are MATLAB R2013a, WSN, and $mEECSM$. The steps of the methodology adopted for this research, towards developing a modified energy-efficient clustering with splitting and merging for WSNs are itemized as follows:

a. To replicate and implement the existing $EECSM$ technique developed by Lee and Lee (2013), the following steps were carried out:
    i. Node deployment in a $50m^2$ field
    ii. Broadcast an undecided state-signal and select a Head node.
    iii. Split the network into clusters using the defined splitting threshold.
    iv. Nodes select the closest CH and join in the cluster.
    v. Merge clusters if necessary by using the defined merging threshold.
    vi. Initiate data transmission and count the number of completed transmissions.
    vii. Implement a CH backup mechanism.
b. To develop a modified $EECSM$ for wireless sensor networks using CH handover mechanism, the following steps were carried out:

i. Step (i-v) of methodology (a) are repeated.

ii. Implement a backup CH nodes to take over from CHs when required.

iii. Determine a suitable CH handover threshold to determine when to handover to the backup CH.

iv. Initiate data transmission and count the number of completed transmissions.

c. Simulation of items a and b on MATLAB R2013a and validation of the performance of $mEECSM$ with $EECSM$ using Network Lifetime and Residual Energy Ratio of sensor nodes as performance metrics. Also, carrying out scalability, robustness test relative to Network Lifetime.

3.2 Replication of the Energy Efficient Self-organized Clustering with Splitting and Merging for WSNs (EECSM)

Figure 2.4 shows the flowchart for implementation of the existing energy efficient self-organized clustering with splitting and merging for WSNs. The processes involved in the replication of the existing EECSM are discussed in details in the following sub-sections:

3.2.1 Clustering/Self-Organizing Phase

Sensor nodes are deployed in a geographical area. After deployment, these nodes are to organize themselves to form a network on their own. This typically involves the decomposition of the network into clusters of bounded sizes with each cluster having a CH and many CMs. In this approach, the clustering phase commences when the sensor nodes are first scattered in the sensor field or after the completion of the "data transmission phase." To achieve energy efficiency, the criterion in the selection of CH is the remaining energy of CMs. The clustering

phase is comprised of four steps: broadcasting step, splitting step, CH selection step, and clustering step.

The process of nodes deployment, sink node deployment, and the First and Second CH selection are as illustrated:

a. **Node Deployment:** This involves deploying the sensor nodes in the field randomly. At first, 100 nodes were randomly distributed in the sensor field of $50m^2$ as shown in Figure 3.1. Considering the X and Y axis of the Cartesian plane, the sensor field was modeled using the developed MATLAB function code (algorithm) as follows:

```
function deployment();n=100;xcor=50;ycor=50;Figure(1);
for i=1:1:n
S(i).xd=rand(1,1)*xcor; XR(i)=S(i).xd;
S(i).yd=rand(1,1)*ycor; YR(i)=S(i).yd;
plot(S(i).xd,S(i).yd,'o-r');
xlabel('X-Axis')
ylabel('Y-Axis')
hold on;
end, end
```

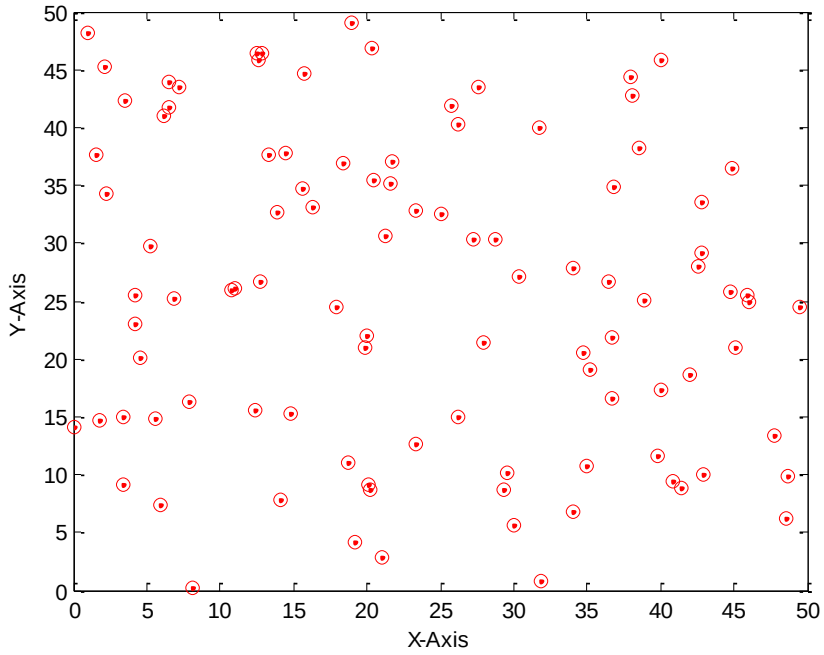


Figure 3.1: Random Sensor Node Deployment

b. Sink Node Deployment

The deployed sink node is shown in Figure 3.1, represented with the big blue colored node. The developed MATLAB function code for positioning the sink is as follows:

```
function sink() % sink position
sink.x=0.5*xcor;
sink.y=3*ycor;
Figure(1); hold on;
S(n+1).xd=sink.x;
S(n+1).yd=sink.y;
plot(S(n+1).xd,S(n+1).yd,'o', 'MarkerSize', 12,
'MarkerFaceColor', 'b'); ylim([0, 160]); grid on;
```

From Figure 3.2, the initial CH node is represented with the green colored node. This initial CH selects the First and Second CH based on their residual energy ratio and number of neighbors. The process of selecting the First and Second CH is explained in the splitting cluster step of section 3.2.1.2. The First and Second CH are represented with the blue colored nodes. The red nodes represent the undecided state nodes (that is, nodes waiting to become CH or connect to a CH). Since CHs have been determined, they broadcast '*CH-signal'* packet to the entire sensor field. The undecided state nodes decides on which CH to connect to by selecting the closest CH and then sending '*request to join*' signal packet to the CH. The CH sends ACK to the nodes and confirms them as CMs.

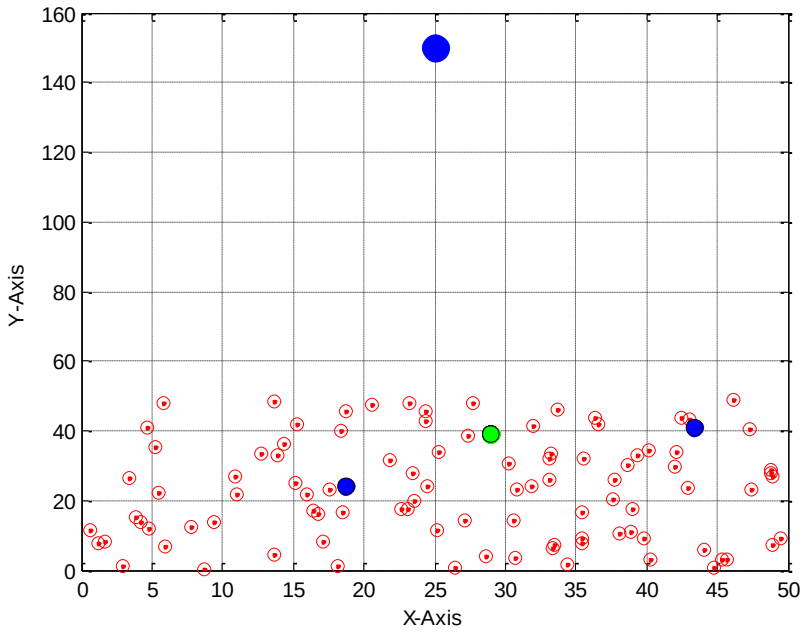


Figure 3.2: Position of the Sink and CHs at Period Zero

3.2.1.1 Broadcasting Step

At initial deployment, there are no CHs. Every node in the network competes for the position of a CH by broadcasting information about its "sensor ID" and "energy level" to neighbor nodes

within its broadcasting range (10 meters). This broadcasted information is called the *"undecided state signal"*. Any node that received broadcasted energy level higher that its own automatically stops competing for the position of a CH and tries to connect with the node that reported higher energy level which then becomes the initial CH. In case of a tie in energy level of nodes, the number of neighbors within the broadcasting range of a node is considered. That is, the node having the most neighbors is selected as the initial CH. Thus, a cluster is formed after which splitting now takes place according to set parameters. This splitting and merging technique is part of the self-organizing ability of the network. The developed MATLAB function code for the broadcasting step is shown in Appendix B.

3.2.1.2 Splitting Cluster Step

The node with the highest number of members within its broadcasting range is selected as the initial CH to partition the network into clusters by choosing the First CH based on residual energy level. The initial CH transmits a '*First CH-signal'* to the First CH. The First CH broadcasts CH-signal packet to its neighbors and receives ACK from them. The First CH aggregates the number of ACKs received from neighbors and transmits to the CH. The initial CH decides the Second CH that is, the node having maximum residual energy of the cluster, except for the First CH and the nodes included in its list of neighbors. The initial CH transmits a *'Second CH-signal'* to the Second CH and the initial CH then becomes a node with an undecided state. The developed MATLAB code for splitting the network into clusters is shown in Appendix C.

3.2.1.3. CH Selection Step

After selecting the First and Second CH, the initial CH that participated in the selection process become nodes with an undecided state. Hence. EECSM executes the next CH selection step. This is where the CHs decide the CH for the next clustering round which is the node with the maximum residual energy level. The CH transmits a *'CH-signal'* to the next CH. The developed MATLAB function code to implement the CH selection step is shown in Appendix D.

3.2.1.4 Clustering Step

The selected CHs broadcast *'CH-signal'* packet and all the nodes in the network compare the distance from the CHs and connect to the one closer to them. The sensor nodes store and list the CH-signal packets due to the fact that information can be used at the re-clustering step of the merging cluster phase so as to reduce overhead of re-clustering. The developed MATLAB function code for clustering step is shown in Appendix E.

3.2.2 Merging Cluster Phase

When the number of nodes in a particular cluster is less than or equal to the merging threshold value which may be as a result of nodes breakdown after a long operation period of the network, the CH broadcast a '*merging-cluster signal'* to the nodes. That CH then becomes an ordinary node.

3.2.2.1 Re-clustering Step

All the nodes initially connected to that CH, realizing they no longer have a CH, connect to a CH closer to them by sending '*request to join*' signal packet. The CH that receives this message sends ACK and add them to its cluster. Re-clustering is performed using the information stored

in the clustering step of the clustering phase. The developed MATLAB function code to implement the merging cluster phase is shown in Appendix F.

3.2.3 Data Transmission Phase

Once the merging cluster phase is finished, clustering is complete. The EECSM enters into the data transmission phase. In the data transmission phase, information gathered from the sensor field is sent to the CHs which in turn aggregate this information and relay it to the Base Station (BS). These procedures are repeated during each clustering round. The developed MATLAB function code to implement this is shown in Appendix G.

3.2.4 CH Backup Mechanism

Once the CH breaks or fails, the CMs that are close to their CH can recognize the breakdown by knowing whether the data packet is transmitted from their CH to the BS. When the CM recognizes a breakdown of its CH during the data transmission phase, CH re-election step is carried out immediately. The CM recognizing a breakdown of its CH is usually the closest CM from the CH. It broadcast a "CH failure signal". Upon receiving this signal, all the CMs broadcast their energy state signal within the broadcasting range twice, in order to elect a new CH. The CM having maximum residual energy becomes the new CH within that range. The new CH then broadcasts the CH-signal to the entire area of the sensor field, since the new CH does not know the area of the cluster exactly. The CMs can then decide their CH not only for the new CH of the cluster, but also for the CHs of other clusters, according to the distance. The developed MATLAB function code to implement the CH backup mechanism is shown in Appendix H.

3.3 Modified Energy-Efficient Clustering with Splitting and Merging ($mEECSM$)

For the development of the modified energy efficient clustering with splitting and merging using CH handover threshold, steps (i - iv) of methodology in section (1.5) were repeated. Figure 3.3 shows the flow chart for the modified EECSM. However, in this modified, a new CH is activated only when the current CH handover threshold is reached or during a sudden breakdown of the CH. Also, for the modified splitting cluster step, the First and Second CHs are allowed to further split the network into twos by selecting another First and Second CH in their respective clusters until the splitting threshold is maintained before the commencement of the merging cluster phase. In addition to these, a backup CH and a CH handover mechanism was incorporated to prevent frequent control message transfer and frequent CH selection at every data transmission respectively. The CH handover mechanism is explained as follows:

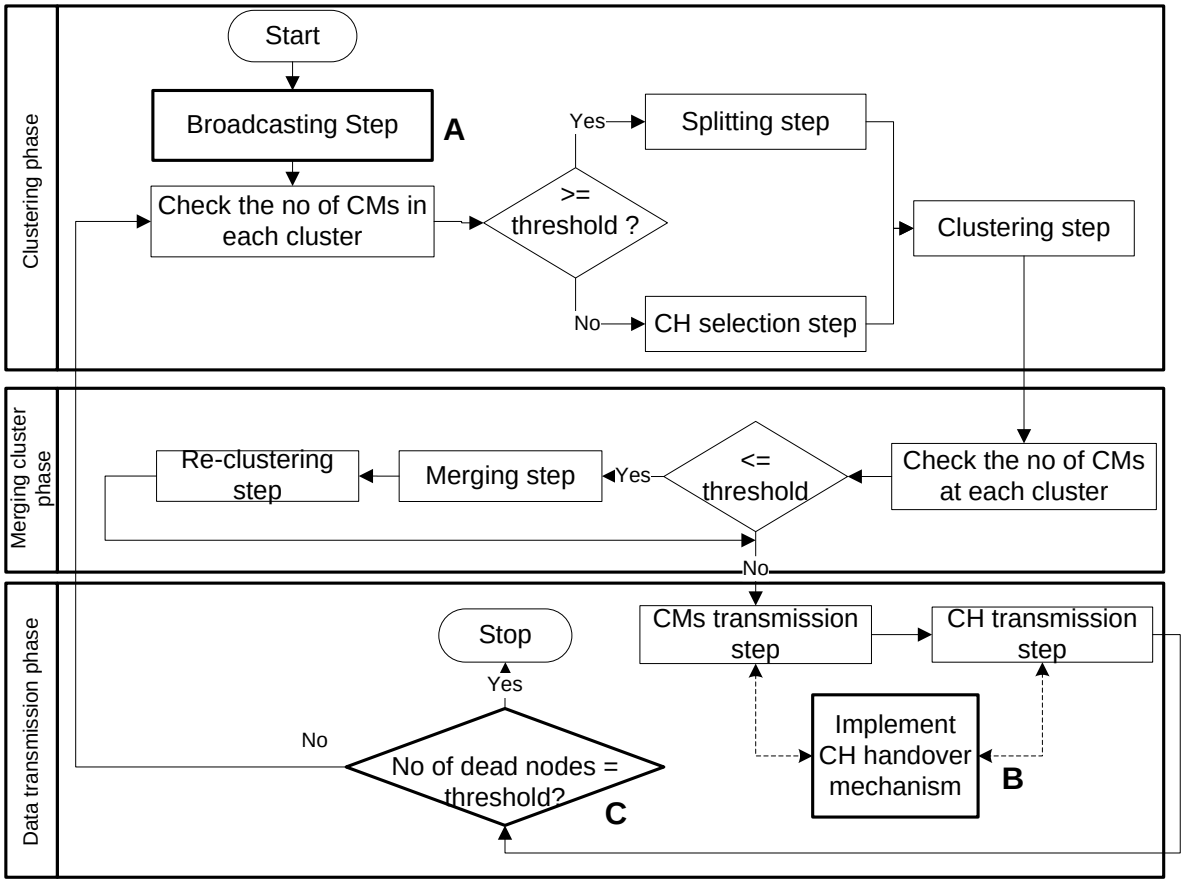


Figure 3.3: Flowchart for the Modified EECSM Model

The dotted bidirectional arrows in Figure 3.3 indicate the flow of control messages between the CMs and CHs while the solid arrow between the CMs and the CHs, indicates a direct communication link through which the sensed information is transmitted from the CMs to the CHs.

3.3.1 CH Handover Mechanism

All the CMs in the cluster send information about their residual energy levels to the CH when requested. This request is sent only when the CH has been selected. Once replies are received by the selected CH, it decides on the backup CH which is the node with the highest energy level and put it to sleep. When the CH handover threshold is reached, the CH transmits '*CH-failure signal'* to the backup CH which then acts as the CH, broadcast '*CH-signal'* and selects its own

backup CH. The CH that reached its set energy threshold then becomes a CM. In this approach re-clustering takes place only when the: CH reaches its handover threshold, CH breaks down, and during the merging process of the cluster.

3.3.1.1 Backup CH Selection Step

Here, a backup cluster head is incorporated into $mEECSM$ to ensure that once the energy level of the CH reaches a certain threshold, it hands over to the backup CH. The remaining energy of nodes is used by the CH in selecting its backup CH. The selected backup CH is usually the node with the highest residual energy level in the cluster and it is put to sleep until its need arises.

Also, when a CH breaks or fails, the CM broadcast a '*CH-failure signal'*. The backup CH upon receiving this signal, quickly wakes from sleep mode and assume the position of the CH, broadcast '*CH-signal'* packet to the entire network, and selects a new node as its backup CH. The CMs then choose the cluster to join by connecting to the closest CH based on their stored information of the distance from other CHs in the network. After which, the spitting and merging step is again, executed. The CH decides its backup which later becomes the CH when the energy level of the current CH reaches a set threshold. In this approach, the CMs do not need to exchange energy state signal since they do not participate in the backup CH selection process. The developed MATLAB function code for selecting the backup cluster head is shown in Appendix I.

3.3.1.2 Determining the CH handover threshold

Simulations were conducted to determine the most energy efficient CH handover threshold by allowing a CH to use 10%, 20%, 30%, 50%, 60%, and 70% of its energy level respectively,

before handing over to the backup CH. The threshold at which the network lifetime was increased the most, was chosen and used throughout the simulation. The desired CH Handover Threshold ($E_{threshold}$) is set using equation (3.1). The developed MATLAB function code is shown in Appendix J.

$$E_{threshold} = E_{current}\ x\ Selected_{threshold\ in\ \%} \quad (3.1)$$

where:

$E_{current}$ is the current energy of the node

$Selected_{threshold\ in\ \%}$ is the selected threshold of 10% to 70% respectively

3.4 Establishing Communication between the CHs and the CMs

To establish communication between the CHs and the CMs, the CHs were designed to broadcast a "CH-signal" packet. Upon receiving this signal, the nodes in the network send a "request to join" message to the corresponding CH which is the closest, and then wait for acknowledgement. The CH receives all the request messages sent by the nodes that would like to join in the cluster, sends acknowledgement to the nodes, confirms them as members of the cluster, and then adds them to its register. The flow chart for the communication process between the CMs and the CHs is shown in Figure 3.4. This flow chart is embedded in item B of Figure 3.3.

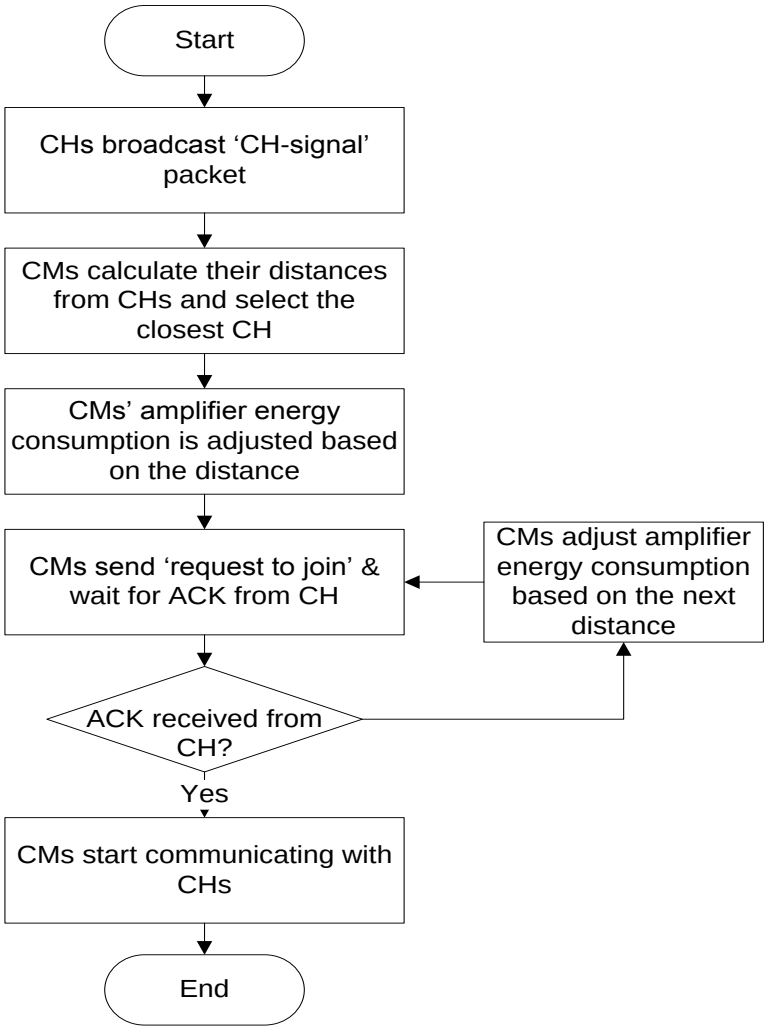


Figure 3.4: Flowchart for the Communication Process between the CHs and the CMs

3.5 Calculating the Distance between CMs and CHs

The distance d, of a node from a CH is calculated from any position by defining its $I\ and\ J$ coordinate relative to the position of the CH given as:

$$d = \sqrt{(I_1 - I_2)^2 + (J_1 - J_2)^2} \tag{3.2}$$

Where $I_2\ and\ J_2$ represent the position of node 2, $I_1\ and\ J_1$ represent the position of node 1 in x and y axis respectively. The developed MATLAB function code to calculate the distance is as shown:

```
function dij = dist(nodeI, nodeJ)
     dij = sqrt(((nodeI(1)-nodeJ(1))^2) + ((nodeI(2)-
     nodeJ(2))^2));
end
```

3.6 Determining the Network Lifetime and Residual Energy Ratios

If the initial energy of a sensor node is 0.5 Joules as stated in section 3.9, for every data transmitted or received, energy is consumed. The energy consumed during transmission is given by equation (2.7) while the energy consumed for data reception is given by equation (2.8). This consumed energy is subtracted from the initial energy of the node according to the number of data transmissions and receptions made. This process continues until the residual energy of the node becomes zero. The number of completed transmissions until the residual energy level of certain number of nodes in the network become zero is taken as the network lifetime which is recorded in Table A1 in Appendix A.

As the network lifetime increases, it gets to a point where a node is drained of its energy. At every node discharge, the residual energy ratio of the network is calculated using equation (2.2). The residual energy ratio of the network is calculated and recorded for every drained node until after the discharge of the 30th node when the network is considered dead. These recorded ratios at the discharge of the 30th node is seen in Table A2 in Appendix A.

3.7 Performing the Scalability and Robustness Test

For the scalability test, the number of sensor nodes was increased from 100 to 200, 300, 400, and 500. The network lifetime in this regard, is defined as the time when 30% of the sensor nodes are discharged of their energy.

To prove the robustness of the network, experiment was conducted to compare the network lifetimes of normal states and abnormal states. The normal state means that a WSN is operating normally, without sudden breakdowns of sensor nodes. The abnormal state means that 0, 10, 20, 30, or 40 sensor nodes suddenly broke sporadically in the WSNs. In this experiment, the network lifetime is defined as the time when 50% of the sensor nodes have malfunctioned. For example, when the number of sensor nodes are 100, 50% of sensor nodes which malfunctioned consist of sensor nodes discharged of their energy (A) and sensor nodes which suddenly broke down (B). Where (A, B) will be taken as (50, 0), (40, 10), (30, 20), (20, 30), or (10, 40) (Lee & Lee, 2013).

3.8 Experimental Specifications

All assumptions and parameters considered during simulation are as shown (Lee and Lee, 2013):

a. The locations of all sensor nodes and the BS are fixed.

b. The deployment of sensor nodes uses random distribution.

c. The location of the BS ($x-axis$: $25m$, $y-axis$: $150m$) is known in advance in the $50m\ by\ 50m$ sensor field. For this field, the broadcasting range of $mEECSM$ is set to $10m$.

d. The data packet size is 1,000 bits, and the signal packet size is 50 bits.

e. All sensor nodes have an initial energy of 0.5 J.

f. The comparison test utilized the average performance of 10 different deployments of sensor nodes in the sensor fields.

g. It is assumed that a WSN cannot operate when more than 30% of the sensor nodes are discharged of energy.

h. In these experiments, the number of sensor nodes is 100, except for scalability experiment.

i. Cluster heads directly transmit the data packets received from their CMs to the BS.

j. The per bit energy consumption for transmitter and receiver circuitry (E_{elec}) and the per bit energy consumption by the node's amplifier (ε_{amp}) is assumed as 50 nJ/bit and 100 $pJ/bit/m^2$ respectively.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents simulation results and discussions. Simulations were carried out according to the topology of Figure 3.3. Simulations were carried out for different CH handover threshold and for the energy consumed by nodes. Results of scalability, robustness, network lifetime and residual energies of sensor nodes were obtained and used to evaluate the performance of the modified model. Finally, comparison was done between the results obtained using EECSM and those obtained using $mEECSM$.

4.2 Results

The results obtained through simulations for CH handover threshold, network lifetime, residual energy ratios, scalability, and robustness are discussed under the following headings:

4.2.1 CH Handover Threshold

Figure 4.1 shows the network lifetime for different CH handover threshold. Simulations were carried out to determine the CH handover threshold from 10% to 70% suitable for the CH to initiate handover to its backup CH. From the Figure, it is seen that the highest increase in network lifetime was achieved when the CH handover threshold of 40% was used which is about 3042 periods. That is, when the energy level of a CH reaches 40% of its initial energy level, handover takes place. This is when the CH hands over its responsibilities to the backup CH and become a CM. Figure 4.2 shows the average residual energy ratio for different CH handover threshold. From the Figure, the average residual energy until the 30th node is drained of its energy was lowest when the CH handover threshold of 40% was used which was about

0.4611%. This indicates that the load is more evenly distributed in the network when a CH handover threshold of 40% is used. Therefore, throughout the simulations, CH handover threshold of 40% was used. CH handover to the backup CH occurred when the energy level of the CH reached a threshold of 40% of its initial energy. The results for the charts of Figures 4.1 and 4.2 are obtained from Tables A1 and A2 in Appendix A.

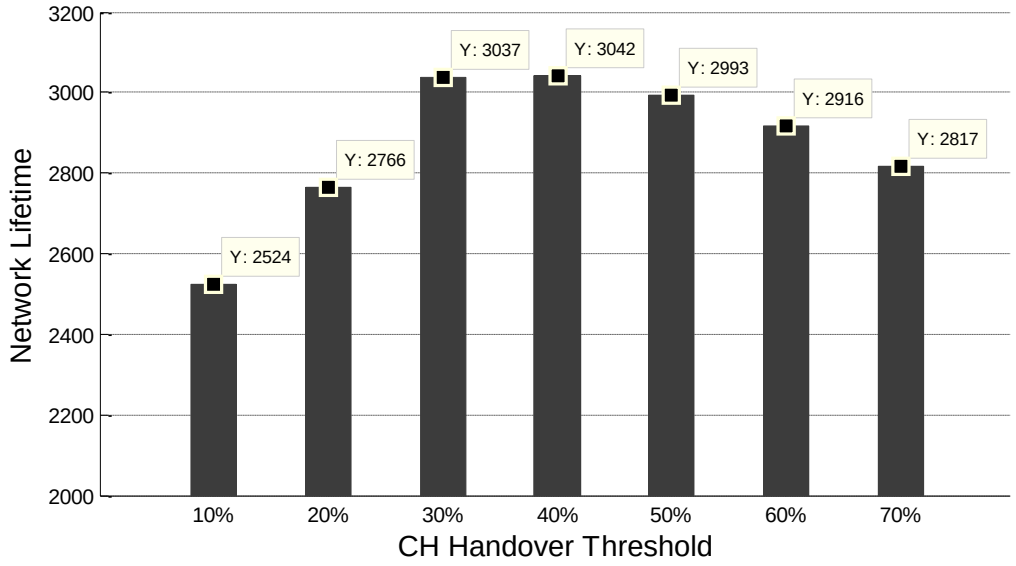


Figure 4.1: Average Network Lifetime for Different CH Handover Threshold

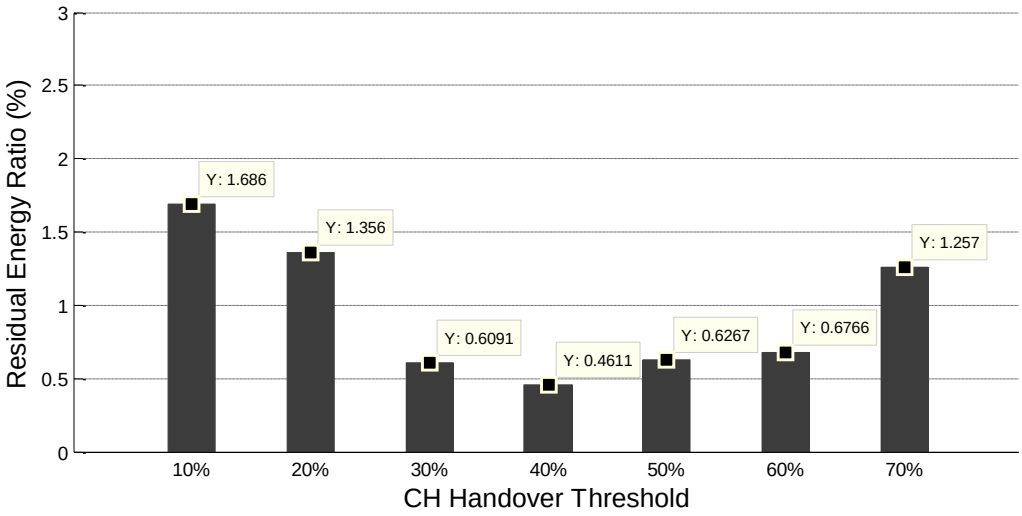


Figure 4.2: Average Residual Energy for Different CH Handover Threshold

From Figures 4.1 and 4.2, it is seen that at a threshold of 10%, 20%, and 30%, handover takes place too often which give room to frequent exchange of control packets between the CMs and the CHs. This process consumes energy thereby reducing the network lifetime. Also, at a threshold of 50%, 60%, and 70%, handover takes place when the CHs have almost exhausted or used a substantial amount of their energies. This causes a reduction in network lifetime since their energy levels become insufficient to handle the responsibilities of a CH by the time they are to become CHs again.

4.2.2 Network Lifetime Comparison

In Figure 4.3, the network lifetime against number of dead sensor nodes was plotted for 100 nodes in a field of $50\ m^2$. The number of periods it took 30 nodes to be completely discharged of their energy was taken as the lifetime of the network. From the Figure, it is seen that using the model, $EECSM$, developed by Lee and Lee (2013), it took an average of 2632 periods for the first node to be completely discharged of its energy as against the 2933 periods it took using $mEECSM$. Also, it took an average of about 2831 periods for the 30th node to be completely discharged of its energy as against the 3042 periods it took using $mEECSM$. This indicates that the backup CH coupled with a suitable CH handover threshold aided in prolonging the network lifetime.

Thus, it can be stated that the network lifetime of $mEECSM$ is 11.4362% (when the 1st sensor node is discharged) and 7.4532% (when the 30th sensor node is discharged) longer than the lifetime of $EECSM$, respectively. This improvement is as a result of the presence of backup CH and the use of a suitable CH handover threshold of 40%.

Tables A3 and A4 in Appendix A show the average value of Network Lifetime and Residual Energy Ratio when the 1st and the 30th node are discharged of their energies using $mEECSM\ and\ EECSM$ respectively. This result is also shown in Figure 4.3.

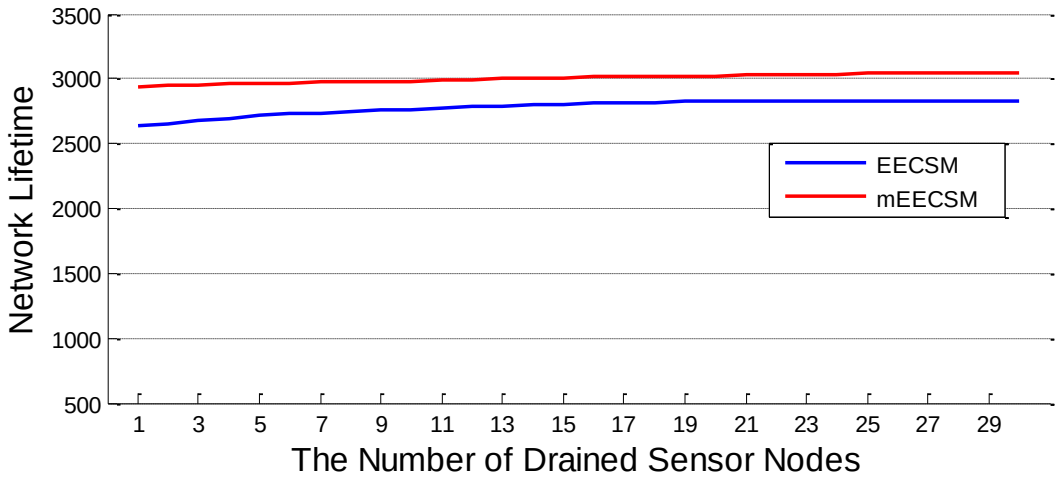


Figure 4.3: Comparison of Network Lifetimes of $mEECSM$ and $EECSM$

4.2.3 Comparison of Residual Energy Ratios

From Figure 4.4, the average residual energy ratio of $mEECSM$ when the 1st and 30th nodes are discharged of their energies are 4.4310% and 0.4611% respectively. Also, the average residual energy ratio of $EECSM$ when the 1st and 30th nodes are discharged of their energies are 8.7336% and 1.0482% respectively. Therefore, from equation (3.4), the average residual energy ratio of $mEECSM$ is at most 50.7351% and at least 43.9897% of that of $EECSM$ which was developed by Lee and Lee, (2013).

The difference between the residual energy ratios of $mEECSM$ from the discharge of the 1st to the 30th node (4.4310% – 0.4611%) is smaller relative to $EECSM$ (8.7336% - 1.0482%). This indicates that the sensor nodes of $mEECSM$ consumed energy much more uniformly than those

of EECSM. The minimum and maximum values of the Residual Energy Ratio are shown in Figure 4.3, and this is obtained from Table A3 and A4 in Appendix A.

With EECSM, a node was already drained of its energy even when other nodes in the network still had about 9% of their energy left. While for $mEECSM$, a node was drained when other nodes had only about 4% of their energy left. This indicates a more balanced network load with $mEECSM$.

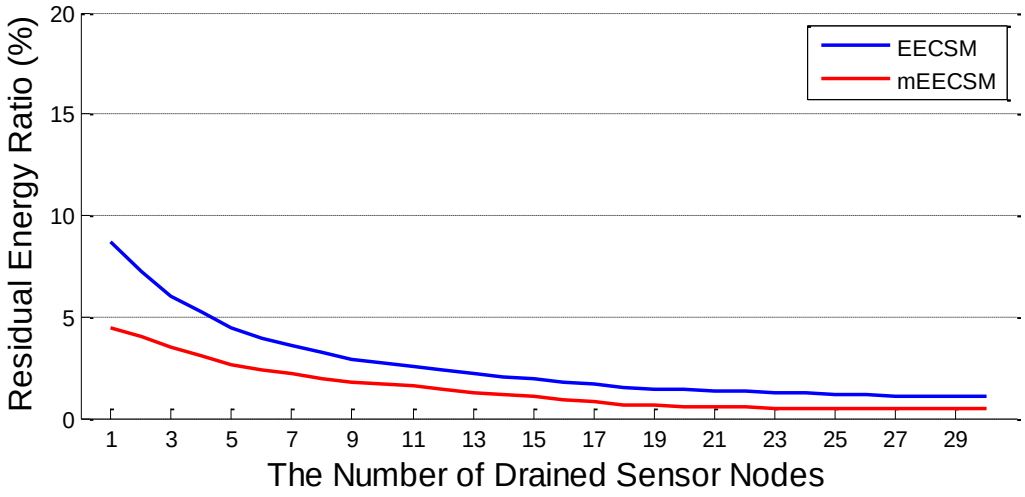


Figure 4.4: Comparison of the Residual Energy Ratios of $mEECSM$ and $EECSM$

4.2.4 Comparison of Scalability

In this experiment, the number of sensor nodes was increased from 100 to 500. The network lifetime of the WSN is defined as the time when 30% of the sensor nodes are discharged. From Figure 4.5, when the number of sensor nodes increases, both $mEECSM$ and $EECSM$ tend to increase in network lifetime. From the Figure, an improvement of 7.4532%, 7.4662%, 7.8618%, 8.0467%, and 8.0968% in the network lifetime of $mEECSM$ over $EECSM$ when the number of sensor nodes is 100, 200, 300, 400, and 500, respectively, was recorded. Thus, it can be concluded that $mEECSM$ has better scalability since its network lifetime is higher than $EECSM$

with 7.4532% and 8.0968% when 100 and 500 nodes are deployed respectively. This improvement was achieved because the frequent selection of CHs at every clustering round was prevented by the use of backup CH. Also, nodes were able to self-organize and join the network by broadcasting the *'request to join'* signal. The average percentage improvement recorded was 7.7922%. This indicates that $mEECSM$ is more scalable compared with $EECSM$ with an average improvement of 7.7922%. The result for the plot of Figure 4.5 is derived from Tables A5 and A6 in Appendix A.

However, it can also be observed from Figure 4.5 that the increase in the network lifetime of $mEECSM\ and\ EECSM$ did not increase proportionally to the increase in the number of nodes from 100 to 200, 300, 400, and 500. This is due to the fact that the nodes participates in gathering and transmitting information at the same time. Also, the nodes became closely packed in the same square meters ($50\ m^2$) and these closely packed nodes resulted in:

a. Nodes calculating their distance between many CHs in order to select the closest since many CHs became available in the network. This process consumed extra energy.
b. CH receiving data and signal packets from many nodes in the network even when they will not be confirmed as CMs. This also consumed extra energy.

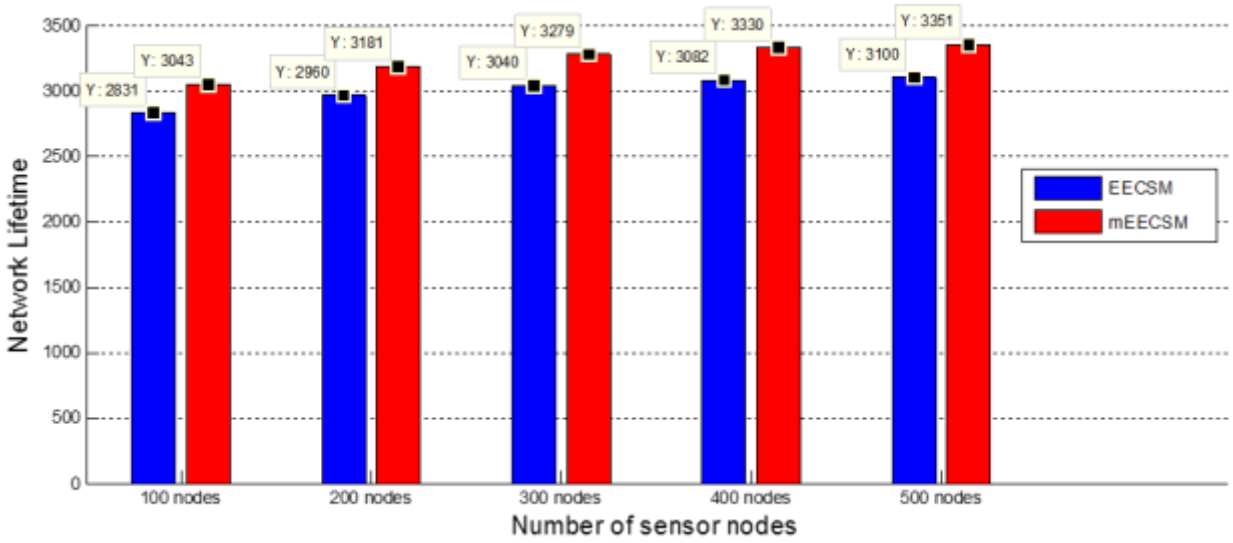


Figure 4.5: Comparison of Scalability of $mEECSM$ and $EECSM$

4.2.5 Comparison of Robustness

Here, the network lifetime is defined as the period until 50% of the nodes are either discharged of energy or have malfunctioned (Lee & Lee, 2013). From Figure 4.6, it is observed that the sudden failure of nodes did not cause much degradation in the network lifetime. From the first deployment when no failure was observed to the deployment when 10, 20, 30, and 40 nodes failed drastically, the network lifetime of $mEECSM$ showed an improvement of 8.8928%, 9.5255%, 10.3499%, 10.8892%, and 11.7602% respectively over that of $EECSM$. From Figure 4.6, it is seen that, although unexpected breakdowns occurred, the effect was not too detrimental to the network.

The average percentage improvement is 10.2725%. This indicates that $mEECSM$ is more robust compared with $EECSM$ with an improvement of about 10.2725%. This improved robustness was achieved as a result of the backup CH which is activated immediately a breakdown occur and also, as a result of the CH handover threshold incorporated into $mEECSM$ which helps to prevent the frequent selection of CHs at every clustering round. Tables A7 and A8 in Appendix

A show the result of network lifetime obtained for sporadic failure of 0, 10, 20, 30, and 40 sensor nodes using $EECSM\ and\ mEECSM$ respectively, noting that the network lifetime is defined as the period until 50% of the deployed nodes are discharged of energy or malfunctioned. The result for the plot of Figure 4.6 is obtained from Table A7 and A8 in Appendix A.

The backup CH which suddenly becomes the CH during a breakdown of the CH, can restore the network to a stable state, without any additional overhead. Hence, the backup CH of $mEECSM$ is a robust technique that is capable of coping with abnormal external environmental conditions of the network.

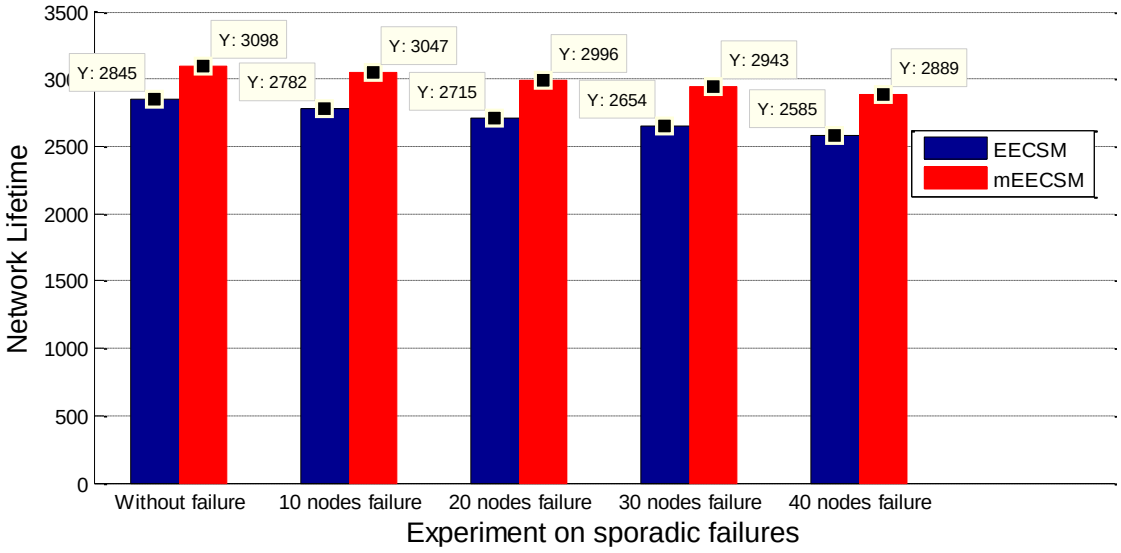


Figure 4.6: Experiment on Sporadic Failures of Sensor Nodes.

The summary of the results of Network Lifetime, Residual Energy Ratio, Scalability, and Robustness is shown in Table A9 in Appendix A.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Summary

This research work presents the development and implementation of $mEECSM$ for WSNs. MATLAB R2013a was used to illustrate its performance. From the results obtained it can be summarized that $mEECSM$ performs better in terms of higher Network Lifetime, Residual Energy, Scalability and Robustness as compared to EECSM. Results obtained from simulations showed that $mEECSM$ was able to improve Network Lifetime, Residual Energy Ratio, Scalability, and Robustness by 7.4532%, 50.7351%, 7.7922%, and 10.2725% over EECSM respectively.

5.2 Conclusions

In this dissertation, an energy-efficient self-organized clustering with splitting and merging for wireless sensor networks was replicated and modified using CH handover mechanism. The modified model integrated a suitable CH handover threshold to maintain fair balance in energy consumptions of sensor nodes and prolong network lifetime. This was achieved by allowing the CH to hand over its responsibility of the Backup-CH once the CH handover threshold was reached. CHs were selected based on their residual energy level. Also, nodes in the network were allowed to connect to the CH closer to them in order to minimize energy consumption. Simulations were carried out and from the presented results, $mEECSM$ improved the network lifetime, scalability, robustness, and residual energy of WSN better than $EECSM$.

5.3 Significant Contributions

Several studies have been carried out to reduce energy consumption in WSN. $EECSM$ is one of the ways to reduce energy consumption in sensor network, but comes with its own challenges. One of the major challenges of EECSM is selecting the next CH at every network round which required each CM sending their residual energy status to their respective CHs at every round. This approach limits the potential of EECSM improving the total network energy consumption.

a. This research was able to achieve a network lifetime of 3042 periods as against 2831 periods recorded using EECSM. Also, a residual energy ratio of 4.4310% as against 8.7336% recorded using EECSM. This brought about 7.4532% and 50.7351% average improvements in terms of network lifetime and residual energy ratio respectively.
b. For the scalability test, a network lifetime of 3237 periods was achieved as against 3003 periods recorded using EECSM which gave an improvement of 7.7922%. Also, an improvement of 10.2725% was recorded in terms of Robustness with a network lifetime of 2995 periods as against EECSM with a network lifetime of 2716 periods.

5.4 Recommendations

a. This network can be deployed to monitor the temperature of birds in a poultry farm.

b. The energy consumed by the CHs communicating with the BS can be taken into consideration since most of the network energy is used up in that aspect.

c. The work can be extended to consider mitigating interference among neighboring nodes by using modulation techniques such as the Time Division Multiple Access (TDMA) technique on the Cluster Member (CM) nodes to increase Network Lifetime.

REFERENCES

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks, 38*(4), 393-422.

Ari, A., Abba, A., Gueroui, A., Labraoui, N., & Yenke, B. O. (2015). Concepts and evolution of research in the field of wireless sensor networks. *International Journal of Computer Networks & Communications (IJCNC), 7*(1), 81 - 98. doi: 10.5121/ijcnc.2015.7106

Camilo, T., Carreto, C., Silva, J. S., & Boavida, F. (2006). An energy-efficient ant-based routing algorithm for wireless sensor networks. Paper presented at the *5th International Workshop on Ant Colony Optimization and Swarm Intelligence,*. 49-59

Castagnetti, A., Pegatoquet, A., Le, T. N., & Auguin, M. (2014). A joint duty-cycle and transmission power management for energy harvesting WSN. *Ieee Transactions On Industrial Informatics, 10*(2), 928-936.

Dahnil, D. P., Singh, Y. P., & Ho, C. K. (2012). Topology-controlled adaptive clustering for uniformity and increased lifetime in wireless sensor networks. *IET Wireless Sensor Systems, 2*(4), 318-327.

Dietrich, I., & Dressler, F. (2009). On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN), 5*(1), 1-38.

Dressler, F. (2006). Self-organization in ad hoc networks: Overview and classification. *University of Erlangen, Dept. of Computer Science, 7*, 1-12.

Ganesh, S., & Amutha, R. (2013). Efficient and secure routing protocol for wireless sensor networks through SNR based dynamic clustering mechanisms. *Journal of Communications and Networks, 15*(4), 422-429.

Hammoudeh, M., & Newman, R. (2013). Adaptive routing in wireless sensor networks: QoS optimisation for enhanced application performance. *Information Fusion, 22*, 3-15.

Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. Paper presented at the *Proceedings of the 33rd annual Hawaii International conference on System sciences,*. 1 - 10

IEEE. 802.15.4, (2003): Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Computer Society, LAN/MAN standards committee, October 2003., 6

Jose, D. V., & Sadashivappa, G. (2015). Mobile sink assisted energy efficient routing algorithm for wireless sensor networks. *The World of Computer Science and Information Technology Journal (WCSIT), 5*(2), 16-22.

Kamyabpour, N., & Hoang, D. B. (2011). Modeling overall energy consumption in Wireless Sensor Networks. *arXiv preprint arXiv:1112.5800*, 1-8.

Lai, W. K., Fan, C. S., & Lin, L. Y. (2012). Arranging cluster sizes and transmission ranges for wireless sensor networks. *Information Sciences, 183*(1), 117-131.

Lee, K., & Lee, H. (2012). A self-organized and smart-adaptive clustering and routing approach for wireless sensor networks. *International Journal of Distributed Sensor Networks, 8*(1), 1-14.

Lee, K., & Lee, H. (2013). Energy-efficient self-organized clustering with splitting and merging for wireless sensor networks. *International Journal of Distributed Sensor Networks, 9*(3), 1 - 12. doi: 10.1155

Liao, Q., & Zhu, H. (2013). An energy balanced clustering algorithm based on LEACH protocol. Paper presented at the *International Conference on Applied Mechanics and Materials*. 1138-1143

Liu, D., & Ning, P. (2007). *Security for wireless sensor networks* (S. Jajodia Ed. Vol. 28). 233, Spring Street, New York, NY 10013, USA: Springer Science & Business Media, 1 & 2.

Mahfoudh, S., & Minet, P. (2008). Survey of energy efficient strategies in wireless ad hoc and sensor networks. Paper presented at the *Seventh International Conference on Networking, ICN*. 1-7

Mbowe, J. E., & Oreku, G. S. (2014). Quality of service in wireless sensor networks. *Wireless Sensor Network, 6*(2), 19 - 26.

Meghji, M., & Habibi, D. (2011). Transmission power control in multihop wireless sensor networks. Paper presented at the *Third International Conference on Ubiquitous and Future Networks (ICUFN)*. 25-30

Mir, Z. H., Lim, K.-W., & , & Ko, Y.-B. (2014). *On the interplay between clustering and power control in multihop wireless networks*. Paper presented at the *Paper presented at the*

11th International Conference on Computer Systems and Applications (AICCSA). 310-317

Østerbø, O., & Grøndalen, O. (2012). Benefits of Self-Organizing Networks (SON) for mobile operators. *Journal of Computer Networks and Communications, 2012*, 1 - 16.

Ovsthus, K., & Kristensen, L. M. (2014). An industrial perspective on wireless sensor networks—a survey of requirements, protocols, and challenges. *IEEE communications surveys & tutorials, 16*(3), 1391-1412.

Patel, M., Chandrasekaran, R., & Venkatesan, S. (2004). Efficient Minimum-Cost Bandwidth-Constrained Routing in Wireless Sensor Networks. Paper presented at the *International Conference on Wireless Networks*. 447-453

Razaque, A., Mudigulam, S., Gavini, K., Amsaad, F., Abdulgader, M., & Krishna, G. S. (2016). H-LEACH: Hybrid-low energy adaptive clustering hierarchy for wireless sensor networks. Paper presented at the *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT).* 1-4

Recayte, E. (2012). *A Java application to display temperature, humidity and luminosity in a Wireless Sensor Network.* (Corso di Laurea in Ingegneria dell'Informazione), Universita degli Studi di Padova ´ 9-20.

Sabet, M., & Naji, H. R. (2015). A decentralized energy efficient hierarchical cluster-based routing algorithm for wireless sensor networks. *International Journal of Electronics and Communications, AEU, 69*(5), 790-799.

Sruthi, K., & Umamakeswari, A. (2014). A study on metrics based clustering algorithms in wireless sensor networks. *International Journal of Advances in Engineering Research (IJAER), No 7*(IV), 10 - 22.

Tahir, M., Javaid, N., Iqbal, A., Khan, Z. A., & Alrajeh, N. (2013). On adaptive energy-efficient transmission in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks, 9*(5), 1 - 10. doi: 10.1155/2013/923714

Thakkar, A., & Kotecha, K. (2015). A new Bollinger Band based energy efficient routing for clustered wireless sensor network. *Applied Soft Computing, 32*, 144-153.

Torkestani, J. A. (2015). An Energy-Efficient Topology Control Mechanism for Wireless Sensor Networks Based on Transmit Power Adjustment. *Wireless Personal Communications, 82*(4), 2537-2556.

Vinothini, M., & Umamakeswari, A. (2014). A study on routing techniques in wireless sensor network. *International Journal of Advances in Engineering Research, No. 7*(IV), 1 - 9.

Wajgi, D., & Thakur, N. V. (2012). Load balancing based approach to improve lifetime of wireless sensor network. *International Journal of Wireless & Mobile Networks, 4*(4), 155.

www.cse.unr.edu, (2016). Wireless Sensor Network Architecture, Retrieved from, http://www.cse.unr.edu/arslan/pmwiki/uploads/WSNArchitecture.jpg. 30th August, 2016.

www.ejaet.com, (2014). Clustering Architecture in WSNs, Retrieved from, http://pubs.sciepub.com/iteces/2/5/1/image/fig1.png, Accessed Jun. 25, 2016.

Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks, 52*(12), 2292-2330.

Zeb, A., Islam, A. M., Baharun, S., Mansoor, N., & Katayama, Y. (2015). A SURVEY ON SELF-ORGANIZED CLUSTER-BASED WIRELESS SENSOR NETWORK. *Jurnal Teknologi, 76*(1), 3-11.

Appendix A

Result Tables

With an initial energy level of 0.5 Joules, the network was able to complete its data transmissions using a CH handover threshold of 40% for an average of 3042 number of times before 30 nodes in the network got drained of energy. The energy consumed for each transmission was calculated using equation (2.7) which was subtracted from the initial energy of 0.5 Joules until the residual energy of 30% of the deployed 100 nodes became zero. The number of data transmissions that occurred is recorded in Table A1 as the network lifetime. This also applies to a CH handover threshold of 10%, 20%, 30%, 50%, 60%, and 70% respectively.

Table A1: Average Network Lifetime for Different CH Handover Threshold

CH handover threshold	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**
1st iteration	2472	2793	3029	3042	3006	2921	2820
2nd iteration	2621	2680	3034	3052	2986	2929	2820
3rd iteration	2431	2773	3047	3030	2987	2916	2816
4th iteration	2524	2853	3037	3052	3005	2924	2815
5th iteration	2513	2773	3042	3035	2989	2906	2810
6th iteration	2478	2771	3029	3046	3006	2921	2820
7th iteration	2601	2810	3034	3031	2986	2915	2820
8th iteration	2544	2683	3037	3057	3005	2924	2815
9th iteration	2563	2853	3042	3035	2989	2906	2810
10th iteration	2489	2675	3043	3043	2975	2901	2822
Average NL	2524	2766	3037	3042	2993	2916	2817

Also, as the network lifetime increases, it gets to a point where a node is drained of its energy. At every node discharge, the residual energy ratio of the network was calculated using equation (2.2). The residual energy ratio was calculated and recorded for every drained node until after the discharge of the 30th node when the network was considered dead. These recorded ratios

after the discharge of the 30th node can be seen in Table A2 for a CH handover threshold of 10%, 20%, 30%, 40%, 50%, 60%, and 70% respectively with 10 iterations each.

Table A2: Average Residual Energy Ratio for Different CH Handover Threshold

CH handover threshold	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**
1st iteration	1.3527	1.3527	0.4053	0.4333	0.2899	0.6559	1.3345
2nd iteration	1.9546	1.2546	0.2535	0.4388	0.6599	0.4654	0.8476
3rd iteration	1.8981	1.2981	0.8938	0.5909	0.8617	0.7261	0.5137
4th iteration	1.9103	1.5103	0.3660	0.5273	0.4693	0.7648	1.5125
5th iteration	1.5324	1.2324	1.4149	0.3579	0.6522	0.8985	1.5079
6th iteration	1.2127	1.2127	0.4053	0.5733	0.2951	0.9559	1.3345
7th iteration	1.9546	1.4546	0.2535	0.4388	0.9592	0.4654	0.8476
8th iteration	1.9103	1.3103	0.3660	0.3529	0.4693	0.7648	1.6355
9th iteration	1.5324	1.5324	1.4149	0.3579	0.6522	0.8185	1.6079
10th iteration	1.6061	1.4061	0.3174	0.5399	0.9582	0.2503	1.4311
Average RE (%)	1.6864	1.3564	0.6090	0.4611	0.6267	0.6765	1.2572

For Table A3 and A4, A CH handover threshold of 40% was used throughout the simulation because it gave the highest increase in network lifetime. The average Minimum values of network lifetime and residual energy ratios using $mEECSM$ and $EECSM$ as shown in Table A3 and A4 is for the discharge of the 1st node respectively. While also, the average Maximum values of network lifetime and residual energy ratios as shown in the Tables is for the discharge of the 30th node respectively. ,

These values were calculated using equation (2.7) and (2.2) for network lifetime and residual energy ratio, respectively.

Table A3: Average Minimum and Maximum value of Network Lifetime and Residual Energy Ratio of mEECSM for 10 Iterations.

mEECSM			
MIN-NL	**MAX-NL**	**MAX-RE**	**MIN-RE**
3012	3042	3.9452	0.4333
2897	3052	5.2478	0.4388
2921	3030	4.9254	0.5909
2938	3052	3.3651	0.5273
3012	3035	5.2452	0.3579
3009	3046	5.2901	0.5733
2844	3031	4.4521	0.4388
2884	3057	3.4782	0.3529
2931	3035	4.2578	0.3579
2878	3043	4.1031	0.5399
2933	**3042**	**4.4310**	**0.4611**

Table A4: Average Minimum and Maximum value of Network Lifetime and Residual Energy of $EECSM$ for 10 Iterations.

EECSM			
MIN-NL	**MAX-NL**	**MAX-RE**	**MIN-RE**
2732	2822	8.9784	0.7411
2558	2836	9.1245	0.8953
2625	2828	9.3278	1.2033
2669	2829	7.9852	0.9871
2587	2839	7.9521	0.9772
2701	2840	8.6733	1.4876
2617	2834	9.4789	0.9772
2538	2834	8.2412	0.8561
2581	2827	8.2412	1.3058
2715	2822	9.3338	1.0512
2632	**2831**	**8.7336**	**1.0482**

For Table A5 and A6, the scalability test was measured against network lifetime by adding more nodes to the network and checking the network response. When more nodes are deployed in the same area, it is expected that an increase in the network lifetime be achieved since the communication distance between CMs and CHs becomes relatively small. This is because, the distance, d in equation (2.7) becomes smaller leading to a reduction in energy consumption which in turns improve the network lifetime.

Hence, the energy consumed during communication which is subtracted from the initial energy of the nodes become smaller leading to an increase in number of completed transmissions. This increase in number of transmissions is seen in Table A5 and A6 for $mEECSM$ and EECSM respectively which translates to increase in network lifetime.

Table A5: Scalability Test for EECSM

EECSM	**100 nodes**	**200 nodes**	**300 nodes**	**400 nodes**	**500 nodes**
1st iteration	2822	2984	3039	3049	3044
2nd iteration	2836	3017	3037	3072	3062
3rd iteration	2828	2987	3051	3090	3120
4th iteration	2829	2891	3035	3045	3055
5th iteration	2839	2980	3046	3093	3113
6th iteration	2840	3019	3048	3104	3106
7th iteration	2834	2871	3037	3102	3101
8th iteration	2834	2957	3021	3075	3057
9th iteration	2827	2898	3053	3085	3235
10th iteration	2822	2995	3035	3106	3105
Average NL	2831	2960	3040	3082	3100

Table A6: Scalability Test for $mEECSM$

$mEECSM$	100 nodes	200 nodes	300 nodes	400 nodes	500 nodes
1st iteration	3042	3212	3239	3375	3306
2nd iteration	3052	3131	3259	3269	3379
3rd iteration	3030	3223	3274	3405	3284
4th iteration	3052	3119	3241	3337	3419
5th iteration	3035	3120	3357	3249	3339
6th iteration	3046	3229	3245	3235	3351
7th iteration	3031	3217	3241	3407	3457
8th iteration	3057	3221	3352	3251	3275
9th iteration	3035	3118	3341	3399	3299
10th iteration	3043	3215	3245	3371	3401
Average NL	3042	3181	3279	3330	3351

For Table A7 and A8, the robustness test was done with respect to the network lifetime by systematically removing a number of nodes from the network. When some nodes fail drastically in a network, it is expected that a decrease in the network lifetime be experienced since the communication distance between CMs and CHs becomes larger. This leads to a decrease in number of completed transmissions as seen in Table A7 and A8 for $mEECSM$ and EECSM respectively.

Table A7: Experiment on Sporadic Failures for $EECSM$

$EECSM$	(50,0)	(40,10)	(30,20)	(20,30)	(10,40)
1st iteration	2846	2788	2709	2665	2585
2nd iteration	2827	2792	2709	2678	2585
3rd iteration	2846	2789	2722	2631	2598
4th iteration	2859	2772	2719	2635	2581
5th iteration	2845	2765	2715	2639	2582
6th iteration	2860	2789	2715	2650	2578
7th iteration	2837	2779	2727	2645	2579
8th iteration	2850	2787	2701	2689	2581
9th iteration	2852	2765	2708	2645	2595
10th iteration	2829	2790	2721	2665	2583
Average NL	2845	2782	2715	2654	2585

Table A8: Experiment on Sporadic Failures for $mEECSM$

mEECSM	**(50,0)**	**(40,10)**	**(30,20)**	**(20,30)**	**(10,40)**
1st iteration	3105	3048	3018	2931	2897
2nd iteration	3092	3051	2982	2941	2891
3rd iteration	3108	3047	2985	2949	2895
4th iteration	3087	3044	3009	2955	2888
5th iteration	3097	3057	2987	2939	2891
6th iteration	3092	3037	2989	2941	2878
7th iteration	3111	3043	3003	2949	2893
8th iteration	3087	3044	2991	2953	2895
9th iteration	3097	3037	2999	2938	2878
10th iteration	3101	3060	2998	2935	2879
Average NL	3098	3047	2996	2943	2889

Table A9: Summary of Results

	EECSM	***mEECSM***	**% Improvement**
$Average\ NL$	2831	3042	7.4532%
$Average\ RE$ Ratio	8.7336%	4.4310%	50.7351%
Average Scalability Value	3003	3237	7.7922%
Average Robustness Value	2716	2995	10.2725%

Appendix B

M-file for the Broadcasting Step

```
function assumptions ()
 clear all
 format short
 BSLocation = [25, 150]; sensorField = [50, 50]; bRange = 10;
 initE = 0.5; n = 100; eElect = 50 * 10^-9;
eAmp = 100 * 10^-12; threshold.spliting = 46;
 threshold.merging = 20; packetSize.Data = 1000;
 packetSize.Signal = 50; deadCount=30; Ethreshold = 0.4;
 bsx=25; bsy=150;  xcor=50; ycor=50;

     % determine the coordinate of the nodes;
CHs = [];
     % if period==0
for i = 1:n
     sensors(i).energy=0.5;
     sensors(i).x=rand(1)*xcor;
     sensors(i).y=rand(1)*ycor;
     sensors(i).rsigpack=0;
end

     % Broadcast an undecided state-signal packet within
     % the broadcasting range
     % calculate transmission energy
for i = 1:n
tE = eElect * packetSize.Signal + eAmp * bRange *
packetSize.Signal;


     % calculate sensor energy
     % Receive the undecided state-signal packets and
     % count the number of received signal packets.
     % broadcast message to all nodes
sensors(i).energy = sensors(i).energy - tE;
for j = 1:n
if i ~= j

     % calculate the distance between sender and receiver
dij =sqrt((sensors(i).x-sensors(j).x)^2+(sensors(i).y-
sensors(j).y)^2);
if dij <= bRange
```

```
     % calculate the received energy, calculate the sensor
     energy & count the number of received signal packets

rE = eElect * packetSize.Signal;
sensors(j).energy = sensors(j).energy - rE;
sensors(j).rsigpack = sensors(j).rsigpack + 1;
end
end
end
end


     % Broadcast the number of received signal packets
     within 2 * (broadcasting range).
for i = 1:n
tE = eElect * packetSize.Signal + eAmp * 2*bRange *
packetSize.Signal;
sensors(i).energy = sensors(i).energy - tE;
othersRSigCount = [];


for j = 1:n
     % Receive the signal packets(from1.3), and compare
     % the number of received signal packets from others
     % with the number of received signal packets of its own.
     % Within 2 * (broadcasting range), if one's value is Max,
     % the sensor node becomes a CH.
 if i ~= j
dij =sqrt((sensors(i).x-sensors(j).x)^2+(sensors(i).y-
sensors(j).y)^2);

if dij <= 2 * bRange
rE = eElect * packetSize.Signal;
sensors(i).energy = sensors(i).energy - rE;
othersRSigCount = [othersRSigCount sensors(j).rsigpack];
end
end
end

if sensors(i).rsigpack >= max(othersRSigCount)
hi=i;
end; end
```

Appendix C

M-file for the Splitting Cluster Step

```
%The CH decides the First CH that is the CM having maximum
residual energy of the cluster.
nc = numel(clusters);
for i=1:nc
     if numel(clusters(i).CM) >= threshold.spliting
for j=clusters(i).CM
CH = clusters(i).CH;
tL = sensors(j).location;
rL = sensors(CH).location;
K = packetSize.Signal;
[tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
sensors(j).energy = sensors(j).energy - tE;
sensors(CH).energy = sensors(CH).energy - rE;
end
     % The CH makes the bCH to be the First CH
firstCH = clusters(i).bCH;
clusters(i).bCH = [];
     % The CH transmits a First CH-signal to the First CH.
CH = clusters(i).CH;
tL = sensors(CH).location;
rL = sensors(firstCH).location;
K = packetSize.Signal;
[tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
sensors(CH).energy = sensors(CH).energy - tE;
sensors(firstCH).energy = sensors(firstCH).energy - rE;

  % The First CH broadcasts a neighborhood-signal
  % to neighbor sensor nodes.
```

```
tE = eElect * packetSize.Signal + eAmp * bRange *
packetSize.Signal;
sensors(firstCH).energy = sensors(firstCH).energy - tE;
listOfNeighbors = [];
eMax = 0;
for m = clusters(i).CM
dij = norm(sensors(m).location - sensors(firstCH).location);
  if dij <= bRange
rE = eElect * packetSize.Signal;
sensors(m).energy = sensors(m).energy - rE;
   % The CMs that receive a neighborhood-signal
   % transmit ACK to the First CH.
tL = sensors(m).location;
rL = sensors(firstCH).location;
K = packetSize.Signal;
[tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
sensors(m).energy = sensors(m).energy - tE;
sensors(firstCH).energy = sensors(firstCH).energy - rE;
listOfNeighbors = [listOfNeighbors, m];
else
     % The CH decides the Second CH that is the CM having
     % maximum residual energy of the cluster, except for the
     % First CH and the CMs included in the list of neighbors.
if sensors(m).energy > eMax
eMax = sensors(m).energy;
secondCH = m;
end;
end;
end
     % The First CH aggregates information of ACKs,
     % and transmits to the CH. (Listofneighbors)
 tL = sensors(firstCH).location;
```

```
rL = sensors(CH).location;
K = packetSize.Signal * numel(listOfNeighbors);
[tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
sensors(firstCH).energy = sensors(firstCH).energy - tE;
sensors(CH).energy = sensors(CH).energy - rE;

% The CH transmits a Second CH-signal to the SecondCH.
tL = sensors(CH).location;
rL = sensors(secondCH).location;
K = packetSize.Signal;
[tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
sensors(CH).energy = sensors(CH).energy - tE;
sensors(firstCH).energy = sensors(firstCH).energy - rE;

% The CH becomes a node with a CM state.
```

Appendix D

M-file for the CH Selection Step

```
 for j=clusters(i).CM
  % The CH decides the next CH that is the CM having
  % maximum residual energy of the cluster
    if sensors(j).energy > eMax
         eMax = sensors(j).energy;
         newCH = j;
    end
 end
  % The CH transmits a CH-signal to the next CH.
      oldCH = clusters(i).CH;
      tL = sensors(oldCH).location;
      rL = sensors(newCH).location;
      K = packetSize.Signal;
      tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
      sensors(oldCH).energy = sensors(oldCH).energy - tE;
      sensors(newCH).energy = sensors(newCH).energy - rE;

CHs = [CHs, newCH];
```

Appendix E

M-file for the Clustering Step

```
CHs = unique(CHs);
    clusters = struct([]);
    clustNum = 0;
    for i=1:n
        if sensors(i).energy <= 0
           sensors(i).state = 'Discharged';
        else
            if ismember(i, CHs)
                clustNum = clustNum+1;
                sensors(i).state = 'CH';
                clusters(clustNum).CH = i;
                clusters(clustNum).CM = [];
     % CH broadcasts a CH-signal packet to the entire area of
     the sensor field.
tE = eElect * packetSize.Signal + eAmp * 50 *
packetSize.Signal;
sensors(i).energy = sensors(i).energy - tE;
else
sensors(i).state = 'Undecided';
end
end
end
     % Others decide their own CH, by finding the closest CH.
for i = 1:n
  if strcmp(sensors(i).state, 'Undecided')
dijs = [];
for j = 1:numel(clusters)
rE = eElect * packetSize.Signal;
sensors(i).energy = sensors(i).energy - rE;
```

```
dij = norm(sensors(i).location -
sensors(clusters(j).CH).location);
dijs = [dijs dij];
end

[dijMin, ind] = min(dijs);
clusters(ind).CM = [clusters(ind).CM, i];
sensors(i).state = 'CM';
end; end
[sensors, clusters] = selectbCH(sensors, clusters, packetSize,
eElect, eAmp);
end
```

Appendix F

M-file for the Merging Cluster Phase

```
dc=[];
for i=1:numel(clusters)
 if numel(clusters(i).CM) <= threshold.merging
% broadcast a merging cluster signal packet.
tE = eElect * packetSize.Signal + eAmp * bRange *
packetSize.Signal;
sensors(clusters(i).CH).energy =
sensors(clusters(i).CH).energy - tE;

% All CMs and CH of the cluster become nodes
% with an undecided state.
sensors(clusters(i).CH).state = 'Undecided';
sensors(clusters(i).bCH).state = 'Undecided';
for j=clusters(i).CM
rE = eElect * packetSize.Signal;
sensors(j).energy = sensors(j).energy - rE;
sensors(j).state = 'Undecided';
end
dc=[dc i];
end
end

dcount = 0;
for i=dc
clusters(i - dcount)=[];
dcount = dcount + 1;
end
 % All nodes with an undecided state decide on their own CH,
```

```
    % by finding the closest CH
for i=1:n
 if strcmp(sensors(i).state, 'Undecided')
 if sensors(i).energy > 0
dijs = [];
for j=1:numel(clusters)
dij = norm(sensors(i).location -
sensors(clusters(j).CH).location);
dijs = [dijs, dij];
end
[dijMin, ind] = min(dijs);
clusters(ind).CM = [clusters(ind).CM, i];
sensors(i).state = 'CM';
else
sensors(i).state = 'Discharged';
end
end
end
```

Appendix G

M-file for the Data Transmission Phase

```
i=1;
while i <= numel(clusters)
for j=clusters(i).CM
% Each CM Sense environment, Make a data packet and
% Transmit a data packet to its CH.
tL = sensors(j).location;
rL = sensors(clusters(i).CH).location;
K = packetSize.Data;
[tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
sensors(j).energy = sensors(j).energy - tE;
sensors(clusters(i).CH).energy =
sensors(clusters(i).CH).energy - rE;
end

 if sensors(clusters(i).CH).energy > clusters(i).CHThreshold
% Each CH Aggregate received data packets from its CMs
% Transmit a data packet to the BS.
tL = sensors(clusters(i).CH).location;
rL = BSLocation;
K = packetSize.Data;
[tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
sensors(clusters(i).CH).energy =
sensors(clusters(i).CH).energy - tE;

function [tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp)
%ditance between nodes
dij = norm(tL - rL);
tE = eElect * K + eAmp * dij^2 * K;
rE = eElect * K;    end;
```

Appendix H

M-file for the CH Backup Mechanism

```
CH = clusters(c).CH;
dijs = [];
for j=clusters(c).CM
    dij = norm(sensors(CH).location - sensors(j).location);
    dijs = [dijs, dij];
end
[dijMin, ind] = min(dijs);
% A CM that detects a failure of its CH broadcasts a
% CH-failure-signal
tE = eElect * packetSize.Signal + eAmp * 2 * bRange *
packetSize.Signal;

sensors(clusters(c).CM(ind)).energy =
sensors(clusters(c).CM(ind)).energy - tE;

% (Foreach CM received the signal)
e = [];
for j=clusters(c).CM
    dij = norm(sensors(clusters(c).CM(ind)).location -
sensors(j).location);
    if dij <= 2 * bRange
        rE = numel(clusters(c).CM) * eElect *
packetSize.Signal;
        sensors(j).energy = sensors(j).energy - rE;
        % The CMs broadcast an energy state-signal within
        % 2?broadcastingranges
        tE = eElect * packetSize.Signal + eAmp * 2 * bRange *
packetSize.Signal;
        sensors(j).energy = sensors(j).energy - tE;
    end
    % The CMs compare the energy states
    % The CM having maximum energy becomes the new CH
    e = [e sensors(j).energy];
end

[eMax, ind] = max(e);
CH = clusters(c).CM(ind);
clusters(c).CH = CH;
sensors(CH).state = 'CH';

% Cluster recovery step
% The CH broadcasts a CH-signal packet to
% the entire area of the sensor field
```

```
CMs = setdiff(clusters(c).CM, CH);
tE = eElect * packetSize.Signal + eAmp * bRange *
packetSize.Signal;
sensors(CH).energy = sensors(CH).energy - tE;
clusters(c).CM =[];

for j = CMs
    % Receive the CH-signal packet
    rE =  eElect * packetSize.Signal;
    sensors(j).energy = sensors(j).energy - rE;

    % The CMs compare the distances
    djks = [];
    for k = 1:numel(clusters)
        djk = norm(sensors(clusters(k).CH).location -
sensors(j).location);
        djks = [djks, djk];
    end
    [djkMin, ind] = min(djks);

    % If (the distance between the CH of the cluster and the
CM
    % ? The distance between the closest CH of other cluster
and the CM)
    % if The CM becomes a CM of the cluster
    % else The CM becomes a CM of the other closest cluster
    clusters(ind).CM = [clusters(ind).CM, j];
end
```

Appendix I

M-file for the Backup CH Selection Step

```
function [sensors, clusters] = CHHandover(sensors, clusters,
c, packetSize, eElect,  eAmp, bRange)

clusters(c).CM = [clusters(c).CM, clusters(c).CH];
sensors(clusters(c).CH).state = 'CM';
Ethreshold = 0.4;

% the bCH becomes the cH
clusters(c).CH = clusters(c).bCH;
clusters(c).CHThreshold = Ethreshold *
sensors(clusters(c).CH).energy;
clusters(c).bCH = [];
sensors(clusters(c).CH).state = 'CH';

% CM selects a new bCH
for i=1:numel(clusters)
% sensors(clusters(i).CH).state = 'CH';
if i ~= c
clusters(i).CM = [clusters(i).CM, clusters(i).bCH];
sensors(clusters(i).bCH).state = 'CM';
clusters(i).bCH = [];
end
end
[sensors, clusters] = selectbCH(sensors, clusters, packetSize,
eElect, eAmp);
end
```

```
function [sensors, clusters] = selectbCH(sensors, clusters,
packetSize, eElect, eAmp)
for i=1:numel(clusters)
    % each cluster member sends energy
    % information to cluster head
eL = [];
for j=clusters(i).CM
tL = sensors(j).location;
rL = sensors(clusters(i).CH).location;
K = packetSize.Signal;
[tE, rE] = transmitPacket(tL, rL, K, eElect, eAmp);
sensors(j).energy = sensors(j).energy - tE;
sensors(clusters(i).CH).energy =
sensors(clusters(i).CH).energy - rE;
eL = [eL, sensors(j).energy];
end
% make the CM with the maximum energy level the bCH
numel(eL)
[eLmax, ind] = max(eL);
ind=ind(1);
bCH = clusters(i).CM(ind);
clusters(i).CM(ind) = [];
clusters(i).bCH = bCH;
sensors(bCH).state = 'bCH';
end
end
% calculate the residual energy of the new BCH
function [count, resE]= dSensors(sensors, clusters)
count = 0;
resE = 0;
for i = 1:numel(sensors)
 if sensors(i).energy <= 0
```

```
count = count+1;
else
resE = resE + sensors(i).energy;
end; end; end
```

Appendix J

M-file for the CH Handover Threshold

```
[E1max, E1Ind] = max(E1);
numel(clusters(i).CM)
sensors(firstCH).state = 'CH';
clusters(i).CH = firstCH;
clusters(i).CHThreshold = Ethreshold *sensors(firstCH).energy;
clusters(i).bCH = CM1(E1Ind);
sensors(CM1(E1Ind(1))).state = 'bCH';
CM1(E1Ind) = [];
clusters(i).CM = CM1;

[E2max, E2Ind] = max(E2);
nn = numel(clusters) + 1;
clusters(nn).CH = secondCH;
sensors(secondCH).state = 'CH';
clusters(nn).CHThreshold = Ethreshold *
sensors(secondCH).energy;
clusters(nn).bCH = CM2(E2Ind);
sensors(CM2(E2Ind(1))).state = 'bCH';
CM2(E2Ind) = [];
clusters(nn).CM = CM2;
end
end
% check energy threshold of CH and calc. residual energy
if sensors(clusters(i).CH).energy > clusters(i).CHThreshold
i=i+1;
else
[count, resE]= dSensors(sensors, clusters)
pre = (resE/(0.5*n))*100;
     if count > 0
```

```
result = [result; count, period, pre];
end
     if count >= deadCount
fCount = count;
break
end
% CH hands over to bCH
display('handover 1')
[sensors, clusters] = CHHandover(sensors, clusters, i,
packetSize, eElect,  eAmp, bRange);
i=1;
end
else
  % check the number of dead sensor nodes
[count, resE]= dSensors(sensors, clusters)
pre = (resE/(0.5*n))*100;
if count > 0
result = [result; count, period, pre];
end
if count >= deadCount
fCount = count;
break
end
 % CH hands over to bCH
display('handover 2')
[sensors, clusters] = CHHandover(sensors, clusters, i,
packetSize, eElect,  eAmp, bRange);
i = 1;
end
end
```

Appendix K

M-file for the Display of NL and RE Results

```
    if fCount >= deadCount
       drainedSensors = result(:,1);
        netLifeTime = result(:,2);
        reRatio = result(:,3);
               [drainedSensors, netLifeTime]
         Figure;
         plot(drainedSensors, netLifeTime)
         ylim([0, 3500])
               [drainedSensors, reRatio]
         Figure;
         plot(drainedSensors, reRatio)
         ylim([0, 10])
        return
    end
    period = period + 1;
end
end
```