ENERGY AWARE QOS ROUTING PROTOCOL FOR CLUSTER BASED MOBILE WIRELESS SENSOR NETWORK

Dissertation Submitted in partial fulfillment of the requirements
for the degree of
Master of Sciences
By

Abbirah Ahmed
2010-NUST-MS PhD-ELEC (Comm-n) 07

Under the supervision of
Dr. Sameer Qazi

DEPARTMENT OF ELECTRONICS AND POWER ENGINEERING, PAKISTAN NAVY ENGINEERING COLLEGE, NUST KARACHI

NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF ELECTRONICS AND POWER ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Master Studies for acceptance a thesis entitled

“Energy Aware QoS Routing Protocol for Cluster Based Mobile Wireless Sensor Network”

by

Abbirah Ahmed

in partial Fullfillment of the requirements for the degree of Master of Science

Dated: _______________

Supervisor:

Dr. Sameer Hashmat Qazi _________________________________

Guidance and Examination Committee:

Cdr.Dr. Attaullah Memon PN _________________________________

Cdr. Dr. Syed Sajjad Haider Zaidi PN _________________________________

Dr. Khawaja Bilal _________________________________

NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, PAKISTAN

PAKISTAN NAVY ENGINEERING COLLEGE, KARACHI

Date: Sept 2013

Author:	**Abbirah Ahmed**
Title:	**Energy Aware QoS Routing Protocol for Cluster Based Mobile Wireless Sensor Network**
Department:	**Power and Electronics**
Degree:	**M.Sc**

Permission is hereby granted to National University of Science and Technology to circulate and to have copied for non commercial purposes, at its discretion, the above title upon the request of individuals or institutions.

Signature of Author

ABSTRACT

Generally, Wireless Sensor Network consists of small sized, low-cost and low-weighted, sensor nodes with the limited processing power of sensing, communications and computation among them and to the base station (BS). Energy minimization and routing are two major constraints of mobile WSNs due to the moving sensor nodes. As the node move with certain velocity it uses more energy than static sensor node hence suffers from severe energy dissipation. Cluster based and hierarchical routing has been accounted as one of the best routing techniques for WSNs by improving the network performance, energy minimization and efficient power usage among the sensor nodes. We have proposed LEACH-MAE a modification in LEACH-M, our method supports residual energy based CH election with an appropriate mobility model for all the mobile nodes in the network.

The main contribution of our thesis is to propose a routing algorithm which is more energy efficient, reliable and increases the network lifetime of a mobile WSN. In proposed scheme, residual energy based Cluster Head (CH) election has been used to avoid energy dissipation and data loss. Our Proposed LEACH-MAE has been implemented on NS2.

Software simulations are performed to analyze the energy consumption when the nodes have different speeds and moving randomly. We have considered mobility of sensor nodes along with the certain pause time and certain mobility model. Results are found for different scenarios. These results show that energy based CH election technique performs well than random election.

ACKNOWLEDGMENT

First and foremost, I would like to thank Almighty Allah for giving me courage and strength to complete this thesis successfully. I would also like to take this opportunity to express my gratitude to my supervisor, Dr Sameer Qazi, for his trust in me, his continuous support and encouragement throughout this research journey. Without his invaluable guidance, this work would never have been a successful one. Also his true criticism towards technical issues provided us to concentrate on transparency of our work. What I learnt from him will provide me with lifetime benefits.

I would also like to thank my GEC members Cdr. Dr Ataullah Memon, Cdr. Dr Syed Sajjad Haidar Zaidi and Dr. Bilal Khawaja for their suggestions, support and helpful discussions.

I would like to thank Mr. Samer Awwad, Department of Computer and Communication Systems Engineering, Faculty of Engineering, University Putra Malaysia, for his valuable suggestions.

I would like to thank my friends Arsalan Sheikh and Naveera Saeed for their companionship during my work. I would also like to thank all those people who helped me directly or indirectly.

Finally, I would like to extend my profound gratitude to my parents (Mr. and Mrs. Ahmed) and my family for their hard work and encouragement. They all supported me and inspired me at every stage. I would have not reached this far without their love and prayers.

ABBIRAH AHMED (2013)

TABLE OF CONTENTS

LIST OF FIGURES

List of Tables

LIST OF ABBREVIATIONS

WSN	Wireless Sensor Network
MWSN	Mobile Wireless Sensor Network
LEACH	Low Energy Adaptive Clustering Hierarchy
LEACH-M	LEACH-Mobile
LEACH-MAE	LEACH Mobile with Average Energy
BS	Base Station
CH	Cluster Head
NS-2	Network Simulator 2
MEMS	Microelectronic Mechanical System
MANETs	Mobile Ad Hoc Networks
RPGM	Reference Point Group Mobility
RWP	Random Waypoint Mobility
GM	Group Mobility
Tcl	Tool Command Language
OTcl	Object Tool Command Language
NAM	Network Animator
CN_{tot}	Total number of Cluster Nodes
E_{tot}	Total Energy
CI Graphs	Confidence Interval Graphs
RSS	Received Signal Strength

CHAPTER 1
INTRODUCTION

1.1 Introduction

In this chapter, we discuss scope of the study and the goal of our thesis work. Wireless Sensor Network (WSN) is most recent technology in the field of wireless communications network. Mobility in sensor nodes is recently adapted by WSNs for increasing its applications of modern age. This section highlights the technological need of mobile-WSN. One of the major constrain in static WSN is the limited battery power of each sensor node but in mobile WSN energy consumption and the route management due to the mobility of sensor nodes makes it even more complicated. Cluster based routing protocols are best suitable to improve the energy dissipation in WSNs. Conventional energy efficient routing protocols are typically designed for the static WSN which results in poor network performance for mobile WSNs. Therefore we propose extension in one of the hierarchical routing protocol i.e. LEACH to improve its energy efficiency capabilities for mobiles sensor nodes and to prolong the network lifetime.

1.2 Motivation

Wireless sensor networks [1] are one of the most popular fields of research due to the wide applications in physical world [2]. WSN involves sensing, processing, and communication ability which helps to monitor, information gathering and react to events in the environment. These normally consist of hundreds and thousands of small sized, low weight, low cost and low power sensor nodes which are deployed into a particular field for sensing, collecting data, tracking and transferring the information to the base station .These sensor nodes further perform local signal processing functions on the gathered data and forward to the Base Station (BS) either directly or through neighbor nodes [6]. WSNs have many unique features which distinguish them from conventional networks such as small life time, redundant data gathering, energy constraints, heterogeneity and mobility in sensor nodes. WSN support variety of applications includes military, health and med-care, land and underwater disaster management, weather forecast and industry disaster management, pollution exposure.

The key feature of WSNs is they are self-organizing, self configuring and independent networks in which all nodes completely rely on battery power for communication, data gathering and processing.

Regardless of the rapid adaptation of WSNs there exist many challenges which limit the WSNs performance. As sensor nodes are battery operated the main issue in WSN is efficient utilization of energy resource to prolong the life time of the network for static and mobility centric environment. It has been observed that most of the energy consumption comes from data reception and transmission while in Mobile Wireless Sensor Networks one more factor for energy consumption is mobility in sensor nodes.

Other than energy minimization MWSN face many other challenges which include node deployment, self organization, node location, topology control and link connectivity and breakage.

Hierarchical routing protocols [4] have proved to be the best for WSNs to maximize the network lifetime as well as for scalability and communication efficiency. These routing algorithms optimize energy consumption by forming multiple groups of sensor nodes. Data aggregation is performed within the clusters in order to minimize the energy dissipation .

Low Energy Adaptive clustering Hierarchy (LEACH) [3] is one of the first hierarchical routing protocols for wireless sensor networks. LEACH does not support *even* distribution of clusters and equal energy dissipation. However it assumes single hop communication between Cluster Head (CH) and Base Station (BS).

Main challenges in LEACH are how to maintain the mobility in sensor nodes, CH selection and placement among mobile nodes to minimize the energy dissipation.

1.3 Problem Statement

Mobile WSNs are seriously constrained by the energy resource due to the limited battery life of mobile sensor nodes. All the mobile nodes perform the same tasks as static nodes. However, the challenge of less energy dissipation and link breakage due to the mobility causes to degrade the network performance. Heinzelman et al [3] proposed cluster based routing protocol LEACH to improve the power efficiency and reliable data transmission in WSN. LEACH is based on the randomized rotation of CHs principle where only CH is responsible for transmitting the data to base station.

In this thesis new algorithm for CH selection is proposed to support and maintain the mobility factor by overcoming link breakage problem. Proposed CH selection algorithm performs cluster formation in a way that equalizes the energy dissipation and link maintenance by making some changes.

1.4 Objective of Thesis

The main objectives of this thesis are:

- To make proposed improvements on LEACH for mobile WSN.
- Improved CHs Selection in the network.
- To minimize the power dissipation in mobile WSN.
- To increase the network lifetime.
- To test and validate the proposed improvements.

1.5 Thesis Contribution

First hierarchical routing protocol LEACH [1] is proposed by Heinzelman et al. for static WSN. To make it mobility centric protocol, D. S. kim and Y.J.Chung [42] proposed a modification in basic LEACH. In our research work LEACH protocol is modified for MWSN which leads to the development of LEACH-MAE protocol. The major contributions of this work can be summarized as follows:

Improved Network Lifetime: to achieve this goal, LEACH-MAE considers the amount of residual energy of mobile nodes as an important factor during CH election procedure.

Efficient communication: LEACH-MAE considers all the factors in LEACH and LEACH-M which caused the communication failure and provides a better mechanism for efficient communication.

Mobility model for better network performance: we also used a mobility model to avoid the random movement of nodes over the field. This model also helps to decrease the amount of dissipated energy of each individual mobile node. Furthermore, This mobility model can be used with other mobility models and routing algorithms which are dealing with mobile sensor nodes.

Although in LEACH-MAE, the network lifetime has been prolonged in compare to LEACH-M, for applications in which minimum mobility is used LEACH-M is more efficient. In contrast, LEACH-MAE is more efficient where high mobility is observed in all sensor nodes.

1.6 Thesis Organization

The rest of the thesis is arranged as follows:

In Chapter 2 we briefly discuss the background of research, introduction of WSNs, routing protocols and clustering algorithms. In Chapter 3, we discuss the design of our proposed algorithm. Chapter 4 discusses the methodology used to carry out the software simulation for our work. In Chapter 5 we discuss the implementation and software simulation in NS2. Chapter 6 gives the detailed analysis of results achieved and finally in Chapter 7 we present the conclusion of our work and future recommendations.

CHAPTER 2

LITERATURE SURVEY

2.1 Chapter Overview

Clustering algorithms for efficient energy resource management has been most focused area in previous [3, 4, 6, 10] research studies. In this chapter, brief background and introduction for WSNs, conventional routing protocols and clustering algorithms is provided. Mobility model is also discussed with the aim of using it among other energy efficient algorithms in designing the proposed clustering algorithm.

2.2 WSNs (Wireless Sensor Networks)

Wireless sensor networking is most recent evolution in the field of microelectronic mechanical systems (MEMS) technology [8,9]. In these networks tiny, low power, low weight and low cost sensor nodes are responsible for communication, processing and computational tasks. A large number of sensor nodes with limited battery power are deployed randomly and densely over the sensing field. These sensor nodes are self organized and routes the sensed data to the base station (BS) or Sink which can be fixed or mobile.

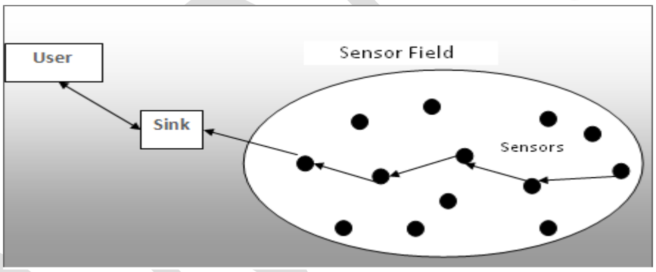


Figure2.1 WSN Architecture

In Figure 2.2 basic architecture of a sensor node is illustrated which consists of: a) Sensor Unit, b) Processing Unit, c) Power Unit and d) Communication Unit. Mobile unit is found in mobile sensor devices for calculating mobility parameters e.g. speed, direction while GPS is found in some location aware devices. Sensor unit comprised of sensor and an ADC converter where sensor collects the specific data from the environment. ADC then translates the sensed data to digital form and sends it to processing unit. CPU or the processing unit is associated with the small storage unit; it interprets the queries and commands to the ADC, responsible for the monitoring of power unit and processes the received data.

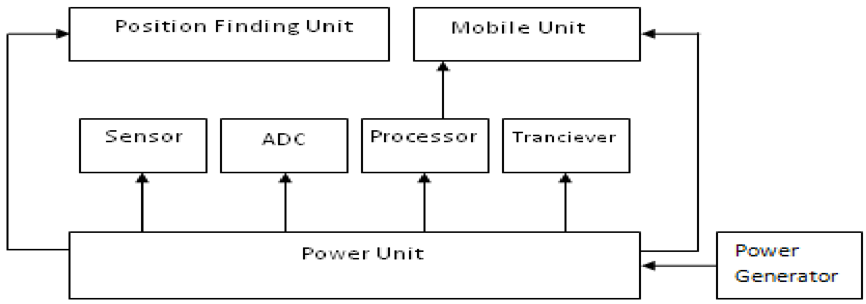


Figure2.2 Sensor Node Architecture [8]

Transceiver unit is responsible for the communication among the nodes as well as with the network. Power unit considered to be the backbone of the sensor which may be supported by solar cells or chemical batteries. These sensor nodes communicate with each other and with the base station through multi hop and single hop techniques. There are certain features which makes the WSNs more applicable includes cost effectiveness, short range broadcast communications, frequently change topology and easy installation.WSN generally works on license-free bands between 913MHz-2.4GHz [42].

Beside these advantages WSNs faces many challenges such as the nodes are power operated and energy resource is one of the major constrain, nodes failure due to fading and rapid topological change, memory limitation, computational complexity, route management and network life time.

To overcome from these challenges many routing protocols and algorithms have been proposed by the researchers [2, 6, 11, 16,28].

2.3 Applications

Wireless Sensor networks can be used for a wide range of applications across various fields [6, 11], some of them are:

- Industrial fields for equipment and machinery monitoring.
- Health and Medicare for tracking the patients, doctors and their activities.
- Land and under water disaster and management such as flood detection.
- Weather forecast.
- Habitat monitoring.
- Environmental applications such as climate and temperature monitoring.
- Home automation, vehicle tracking and home appliances monitoring.
- Military fields' application for target tracking, intelligence, control and commands of the systems as well as nuclear and biological attacks detection.

2.4 Mobility in Wireless Sensor Networks

WSNs were originally assumed to have only static sensor nodes. MWSN is a new area of current research in which all the mobile sensor nodes [11] act same as the static sensor nodes with self-organization and self-configuration capabilities. It has been suggested that mobility feature of sensor nodes helps to improve the sensing coverage[27] and data capacity as well as improves the network lifetime. Another major advantage of MWSN is data reliability. It has been already observed, increased number of hops that a data packet has to travel increases the probability of errors. So if the number of hops is reduced this error probability can be optimized. Further energy spent in data transmission also reduced.

Currently MWSNs [29] have been researched to be improved in the field of node deployment, position, self-organization, topological control, link breakage and connectivity, power saving and route management. Due to the mobility in the sensor nodes conventional routing protocols results in the performance degradation of the network [7].

Hence for the better network performance conventional routing protocols for WSNs are modified for MWSNs.

2.5 Routing protocols for WSNs

Generally routing protocols are categorized on the basis of network structure and protocol operation based [10]. They can further be subdivided in:

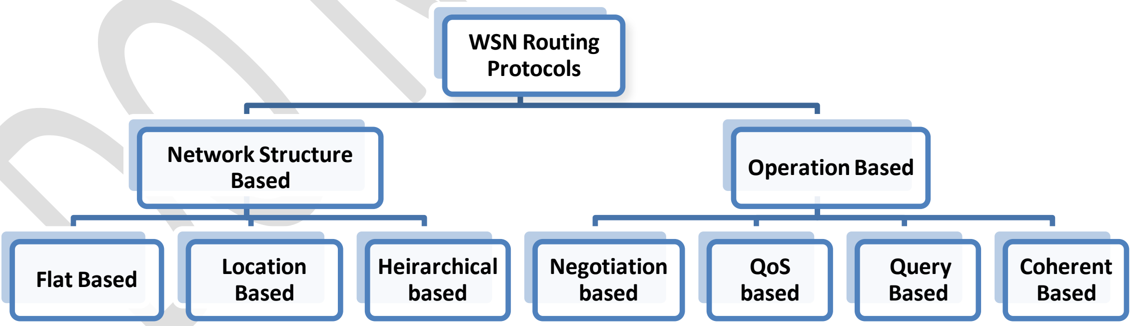


Figure2.3 WSN Routing Protocols Classification

2.5.1 Network Structure Based

This category describes the network feature based on sensor node and base station characteristics.

Flat-Based: All nodes are homogenous and perform equal and same tasks. This is non hierarchical based architecture [14]where data transmission is done by discovering best route to the destination using hop by hop technique.

Hierarchical-Based: This category is more prone to conserve energy [17]by using cluster formation. Sensor nodes can be homogenous or heterogeneous for performing the tasks. In this technique groups of sensor nodes are formed and a group leader is selected which is responsible for further transmission after local data gathering.

Location-based: In this type of routing all data transmission and routing is based on the nodes position. To consume less energy node distance is calculated by using node location information and the node closer to the respective node is more favorable for communication.

2.5.2 Protocol Operation Based

This is based on the characteristics such as communication technique, data delivery method, computational techniques and hop count.

Negotiation Based: Before transferring the date nodes transmits a fixed number of messages to avoid the redundancy. Energy resource can be maintained in these protocols.

Query-based: Node which requires some information broadcast the query throughout the network[13]. Only the node having requested information answers the query and hence saves energy dissipation.

QoS (Quality of Service) Based: These type of protocols are not favorable for energy minimization as the main objective is to provide QoS. A high energy link may be adopted to maintain QoS.

Coherent based: In these protocols some nodes are data aggregated [15]nodes which are responsible for the processing of data after getting it from other nodes however at every single node some possible data processing is done to serve better for energy reservation. Data aggregated nodes must have more residual energy to avoid rapid energy depletion.

Table2.1 Characteristics of WSN protocols [41]

Protocol	Classification	Data Aggregation	Query based	Data Delivery Model
SPIN	Flat Topology Based	Yes	Yes	Event Driven
Direct Diffusion	Flat Topology Based	Yes	Yes	Demand Driven
CADR	Flat Topology Based	No	Yes	Continuous
COUGAR	Flat Topology Based	Yes	Yes	Leader Query
AQUIRE	Flat Topology Based	Yes	Yes	Complex Query
LEACH	***Hierarchical***	***Yes***	***No***	***Cluster Head***
TEEN & APTEEN	Hierarchical	Yes	No	Active Threshold
PEGASIS	Hierarchical	No	No	Chains
SPAN	Hierarchical/Location	Yes	No	Continuous
SAR	QoS based	Yes	Yes	Table Driven
SPEED	QoS based	No	Yes	Geographic

2.6 Hierarchical routing protocols for WSN

Clustering analysis is defined by Backer and Jain [4] as: "A group of objects is divided into many homogenous subgroups on the basis of an often individually chosen measure of similarity such that the similarity among objects within a subgroup is larger than the similarity among the objects belonging to different subgroups".

In recent years cluster based routing also known as Hierarchical routing in WSNs gained a great interest by the researchers to enhance energy efficiency. These are best suited protocols for energy conservation through cluster formation technique. The basic idea of WSNs clustering has been introduced by Heinzelman et al [3].All sensor nodes in the network are grouped together to form several subgroups. In general hierarchical routing is considered for heterogeneous nodes where some nodes are more powerful in terms of energy resource and responsible for direct transmission of aggregated data to the Base Station (BS). Cluster membership may be fixed or variable according to the network architecture. LEACH is one of the fundamental routing protocols in hierarchical routing family.

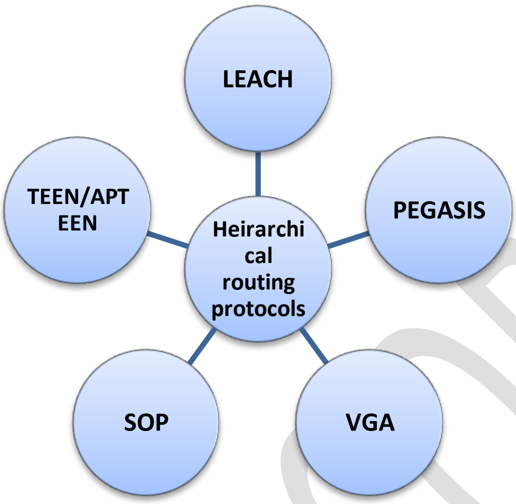


Figure2.4 Hierarchical routing Protocols for WSN [4]

There are many constraints while proposing a new clustering algorithm or modifying existing clustering algorithms. One of major constraint frequently observed is that algorithm should be energy efficient ,having the capability of handling maximum nodes in their respective clusters with the best possible performance.

2.7 Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH is first cluster based routing protocol proposed by Heinzelman et al [3]. It is Self-organized and adaptive hierarchical routing protocol that uses randomized rotation technique for CH selection. In LEACH all the nodes are divided into small clusters where each node has equal probability of being a CH. one node per cluster is chosen to be the CH for every round. In conventional routing protocols CHs are fixed throughout the network lifetime hence results in quick failure of the CHs.

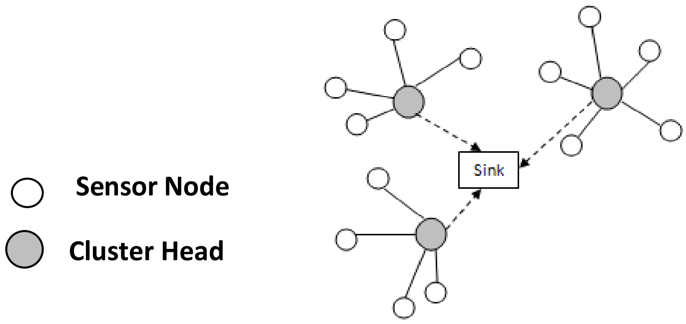


Figure2.5 Clustering Hierarchy in LEACH

All the cluster member nodes transmit data to their respective CH. CH performs local data compression and transfers this aggregated data directly to the base station. LEACH provides equal energy dissipation by rotating CH position among the nodes which helps to improve load balancing within the network.

LEACH operation is based on rounds which are defined as "The time duration a CH takes to get the data from all its cluster nodes, aggregate them and transmits it to the BS".

Each round in LEACH is divided in two phases:

Set-up Phase: Each round starts with the set-up phase. During this phase each node decides to be or not to be the CH based on the probability function. Individual node in the cluster chooses a random number between 0 and 1. If the number is less than the threshold probability T (n) eq (2.1), node is elected to be the CH for current round.

This threshold probability function is given by the following equation:

$$T(n) = \begin{cases} \dfrac{p}{1-p*\left(r \bmod \frac{1}{p}\right)} & , n \in G \\ 0 & \text{otherwise} \end{cases} \qquad (2.1)$$

Where ***p*** is the percentage of CHs, r is current round number and G is the group of nodes that have not been CH for past ***1/p*** rounds.

T (n) increases with the increase in rounds which helps to elect optimal number of CHs when some nodes die out in the network.

Once CHs are elected, a unique code is assigned to each CH by the Base Station through CDMA technique. During CH advertisement all elected CHs inform their neighbor nodes by sending advertisement packet containing the nodes status and the unique code. After receiving advertisements from different CHs, non-CH nodes pick one having strongest received signal strength. Now the cluster formation begins and all the non-CH nodes reply back to their CH with join packet that contains nodes ids. After getting all the join packets from different nodes, each CH sets up TDMA schedule and broadcast to all nodes within the cluster. Setup phase is comparatively shorter than the steady-state phase.

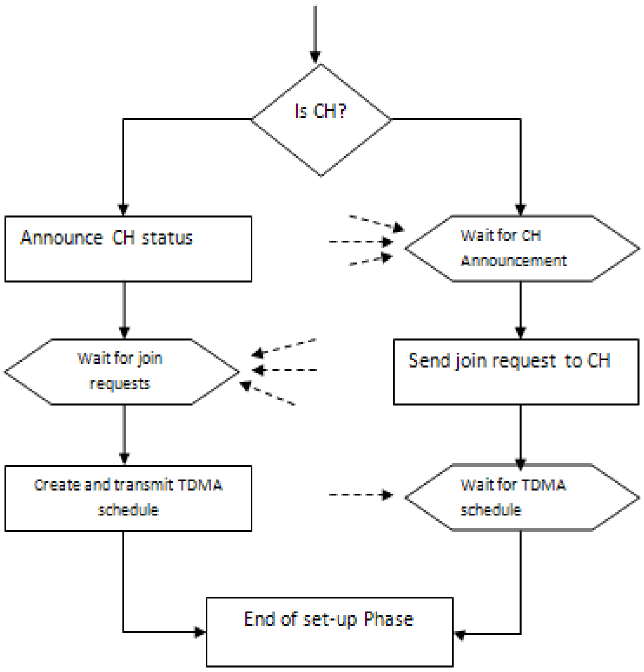


Figure2.6 LEACH Set-up phase

Steady-state Phase: Once clusters are formed steady-state phase is started. During this phase cluster nodes transmits all data to the CH in their respective time frame. Sensor node goes to sleep mode after data transmission and only wakes up according to its TDMA schedule to save the energy resource. However CH must keep their radio on throughout the round to receive data. After getting all the data from cluster nodes, CH performs some local processing and aggregation on it and transmits it to the BS.

Main characteristic of LEACH which makes it dominant over other hierarchical routing protocols is randomized rotation mechanism of CHs which helps it to conserve more energy.

Despite the energy conservation there are certain issues LEACH is not suitable for. In general, LEACH is designed for the static WSN hence mobility handling in LEACH can lead up to poor performance. Furthermore LEACH doesn't provide the placement information of the sensor nodes for each round.

There are many flavors of Leach have been proposed so far for improved network performance [39]:

2.7.1 Two-Level LEACH [40] (TL-LEACH)

In LEACH protocol all CHs are responsible for data aggregation and then for the direct transmission to the base station. Over the sensor field CHs might be located at a larger distance from the BS hence needs more energy for long distance transmission and also due to always on radios they die faster than other sensor nodes.

To overcome with this problem a newer version of LEACH i.e. Two-level LEACH has been proposed where data gathering is same as of basic LEACH, but for the data transmission to the BS it uses a CH that is located between the CH and BS as a relay station.[7]

2.7.2 Multi-Hop LEACH [8]

In this version of LEACH, communication between CHs and BS is multi hop. It picks the best possible path between the CH and BS through other CHs and transmits their data through them.

2.7.3 LEACH-C [13]

Leach protocol provides randomized cluster formation but doesn't provide any information about the placement of the sensor nodes during each round which lead the research to introduce one more flavor of LEACH protocol i.e. LEACH-C which is based on the centralized algorithm for the clustering. Unlike LEACH, during set-up phase each sensor node sends its current position information along with the residual energy level to the BS. Where BS ensures that energy load is evenly distributed among all nodes. After the steady-state phase is same as of leach.

2.7.4 V-LEACH[39]

In LEACH CHs die quickly than other cluster nodes because of the always on radios, data gathering, processing and transmitting to the BS which might be located far away from the CH due to which a large amount of energy is dissipated. Once CH is dead it results in the complete failure of the whole cluster. To prevent from the above failure vice-CH technique has been introduced that takes over the place of CH when it dies out.

2.7.5 F-LEACH[3]

It is based on the fixed clusters which are formed once throughout the network life time. Same like LEACH it uses randomized rotation of CHs within the cluster. This scheme doesn't allow any new node to enter into the system and is unsupportive for the nodes' mobility.

2.7.6 LEACH-M[26]

The evolution in WSN technology introduced mobility in all nodes i.e. sensor nodes and sink. For mobility handling, improvements have been made to the basic LEACH scheme. It is the enhanced LEACH scheme proposed for mobility centric environments. In this scheme all nodes including CH is considered to be mobile. Nodes having less mobility with respect to the other cluster members become CH. Basic purpose of this enhancement is to ensure whether mobile sensor node is able to communicate with specific CH or not. It uses same Set-up Phase as of basic LEACH but with the calculation of node's mobility.

Leach-M is not mobility and traffic adaptive protocol [12, 28] due to which timeslots allocated to the sensor nodes remain wastes if they have no data to send or if they moved out of the cluster. Other major issue is energy depletion after losing the connection to the cluster node needs to keep its receiver on for hearing the CH announcement.

2.8 Mobility Models For WSN

In MWSN, a mobility model describes the pattern of mobile entities along with the location, velocity and acceleration change with time. These mobility models have a great impact on the performance of routing protocols and on mobile networks therefore it is important to choose an underlying mobility model to evaluate any protocol.

To meet the requirements of real world researchers proposed different mobility models:

- Random Way Point Model
- Random Walk Mobility Model
- Reference Point Group Model (RPGM)
- Gauss Markov Mobility model

2.9 The Random Waypoint Model

It is simple and widely used mobility model in wireless networks. Johnson and Maltz [16] proposed it for Ad hoc Networks.

In this model, mobile nodes or entities move randomly and independently without restrictions. A mobile node chooses a random destination and moves towards the point with constant speed. This mobile node waits for a certain pause time, chooses new destination point and speed, then moves towards this new point and continues its movement following the same movement pattern. This movement of the node from one point to another is known as one 'Movement Period' or 'transition' [18]. These destination points are all uniformly distributed over the area.

This model correlates direction and speed [25]. This time of two direction changes is an input parameter of the mobility model but depends upon the node's speed and size of the area. Higher the speed of the node, higher the frequency of the direction changes.

Two important factors of RWP are: a) Transition length, b) Transition period.

Transition Length: It is the Euclidean distance between two points that a node travels during single movement period and the sequence of these distances can be defined as stochastic process.

For a transition length:

$$L_i = \|P_i - P_{i-1}\| \tag{2.2}$$

For a two-dimensional area distribution of waypoints is given by uniform distribution

$$f_{p_x p_y}(x, y) = \begin{cases} 1/(ab) & for\ 0 \le x \le a\ and\ 0 \le y \le b \\ 0 & else \end{cases} \tag{2.3}$$

For two points P1 = (Px1 , Py1) and P2 = (Px2 , Py2), The distance is:

$$L = \|P_2 - P_1\| = \sqrt{|P_{x1} - P_{x2}|^2 + \left|P_{y1} - P_{y2}\right|^2} = \sqrt{L_x^2 + L_y^2} \tag{2.4}$$

Transition period: It is the time taken by a node to reach from one point to other node.

The period is denoted by random variable ***T*** and outcome is denoted by $\boldsymbol{\tau}$. If node moves with constant speed during entire movement,

$$\mathrm{V_i} = \mathrm{v} = \mathrm{const}\ \forall i \tag{2.5}$$

and **v > 0**, then this transition period is:

$$T = \frac{1}{v} L \tag{2.6}$$

To generate trace of RWP a tool named ***setdest*** included in NS-2 package.

2.9.1 Mobility Metric for RWP

Usually RWP is based on average speed of nodes which is relative speed of two nodes that determines link break or link formation rather than their individual speeds and pause time is also there. Average speed hence is not appropriate as it doesn't take account of individual speed of nodes [20].

To overcome this problem Johansson, Larsson and Hedman et al [21] proposed the mobility metric.

If relative speed of two nodes ***a*** and ***b*** at sometime ***t*** is

$$S_r(a,b,t) = \left|\bar{V}_a(t) - \frac{\bar{V}_b(t)}{M}\right| \tag{2.7}$$

Then the mobility metric ***M*** is calculated as the measure of relative average speed for all nodes' pairs at all times.

$$\bar{M} = \frac{1}{|a,b|} \sum_{a=1}^{N} \sum_{b=j+1}^{N} \frac{1}{T} \int_0^T S_r\,(a,b,t)dt \tag{2.8}$$

Where **|a,b|** is number of individual node pair **(a,b)**, ***N*** is total number of network nodes and ***T*** is simulation time. Based on this mobility metric we can measure the level of nodes' speed.

The use of this mobility metric allows measuring the level node speed and helps to differentiate multiple mobility scenarios.

2.10 Random Walk Model

It is the basic mobility model which a variant of Random Waypoint Model introduces random movement patterns. It is used in unpredictable movement situations of a mobile entity just like most of the natural entities which moves in an extremely unpredictable ways [19]. A mobile node moves from one location to another in random direction chosen between [0,2π] at random speed ranging between [$Speed_{min}$, $Speed_{max}$][20]. Before choosing a new direction and speed, nodes' movement occurs in either a constant time interval ***t*** or a constant distance travelled ***d***. It is memory less mobility model which doesn't depends upon the past behaviors and locations of nodes [22, 17].

The RW Mobility Model produces Brownian motion with a small input parameter (distance or time) and, therefore, basically evaluates a static network [19].

2.11 Gauss Markov Mobility model

The Gauss-Markov Mobility Model was introduced by Liang and Haas [22].In this model future movement behavior of nodes depends upon the past speed and direction. It is a temporal dependent model where the degree of dependency is determined by a tuning parameter (memory level) $\boldsymbol{\alpha}$. This parameter can be tuned for various scenarios: (i) **α= 0** the model is memory less, (ii) **α= 1** the model has strong memory and (iii) **0 < α<1** the model has some memory [14]. Initially each node is assigned a current speed and direction. At fixed time intervals n movements occur by all mobile nodes. Particularly, the values of speed and direction calculated at nth instance is based upon the **(n-1)th** instance using the equations below:

$$s_n = \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)s_{x_{n-1}}} \tag{2.9}$$

$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)d_{x_{n-1}}} \tag{2.10}$$

Where s_n and d_n are new speed and direction at n time interval; **α** is a tuning parameter; $\bar{s}$ *and* $\bar{d}$ are constants representing mean value of speed and direction as $\boldsymbol{n} \longrightarrow \infty$; $\boldsymbol{s}_{x_{n-1}}$ *and* $\boldsymbol{d}_{x_{n-1}}$ are random variables for Gaussian distribution. Complete random values (or Brownian motion) are obtained by setting **α = 0** and linear motion can be found by setting **α = 1** [22]. The Gauss- markov model avoids sudden stops and sharp turns by allowing past values to influences future values.

2.12 Reference Point Group Mobility (RPGM) Model

RPGM [17] is a mobility model which represents the movement of mobile entities in a group. Each group of nodes has a group leader which determines the group motion behavior i.e. location, speed acceleration etc. Group movements are based upon the path traveled by a logical center for the group. It is used to calculate group motion via a group motion vector, GM. Individual mobile nodes move randomly about their own reference points whose movement depends upon the group movement [24,39]. When individual reference points move from ***t*** to ***(t+1)*** their location are updated according to the group leader [19].Movement of each group leader and random motion of each individual node within a group follows the random waypoint mobility model by avoiding pause time during group movement[20]. This pause time can only be used after a reference point reaching the desired destination and all group nodes pause for the same time.

2.13 Summary

The performance of any ad hoc network and WSN protocol considerably vary with different mobility models. Performance of any WSN fluctuates due to the movement of nodes, routing protocol and the mobility model. In mobile-WSN random movement of mobile entities leads to the link maintenance issues between nodes which directs to serious data loss. The use of appropriate mobility model with routing protocol helps to overcome this problem. The research study [19] proves that Random waypoint mobility model is a flexible and simple model used in WSN and Ad hoc networks. It appears to create real-world mobility patterns. Further, in random waypoint model mobile entities are more likely to cluster in the center of the sensing area hence increased throughput rate as well as decrease the end to end delay due to minimum link breakage.

From the above study it is concluded that as In random waypoint mobility model, nodes stops at random locations and make sudden turns, during these turns, mostly the direction is toward cluster-head .For this reason packet loss of random waypoint mobility model is less. Authors further concluded in [23] that Random Way Point Mobility Model has highest packet delivery ratio and lowest end to end delay compared to other selected mobility models.

CHAPTER 3
PROPOSED ALGORITHM DESIGN

3.1 Chapter Overview

In this chapter, basic clustering algorithm (section 2.7) is modified with proposed clustering algorithm and implemented on NS-2. Some modifications have been done for cluster formation and for the method of choosing CHs on the basis of residual energy using different network scenarios.

3.2 Proposed Algorithm

In our thesis work we modified Low Energy Aware Clustering Hierarchy (LEACH) [1] for mobility centric environment using energy consumption metric.

In MWSN sensor nodes move in random directions with different speeds shown in figure 3.1:

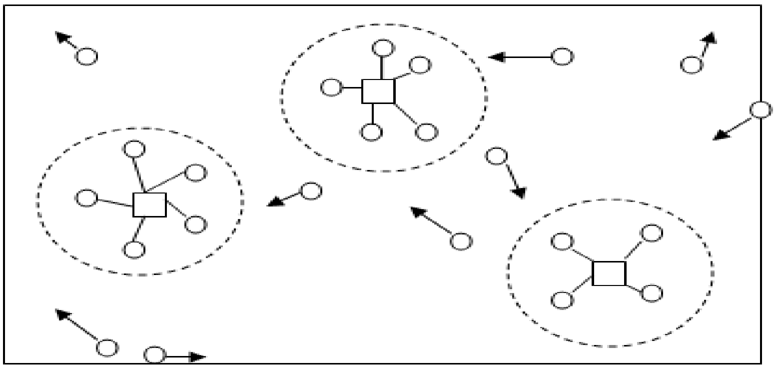


Figure3.1 Random Movement of Mobile Sensor Nodes

This random movement affects the network performance in terms of data loss, higher energy consumption, delays and communication failure. To prevent from these problems an efficient cluster head election algorithm is needed.

In our proposed LEACH-MAE, we consider residual energy of each individual sensor node for a particular round. Like in basic LEACH, our algorithm is based on multiple rounds. Initially random nodes are eligible to be cluster heads with some probability ***P*** [1]. After successful completion of first round cluster head election process is purely energy based throughout the network lifetime.

In our proposed algorithm, we considered mobility of individual sensor for a particular round. Software simulation results prove that our proposed technique helps to prolong the network lifetime.

To get the desired results we made some assumptions in our technique following the basic LEACH algorithm.

Assumptions:

1. All the nodes in the network are homogeneous i.e. have equal amount of energy initially.
2. Error free communication
3. All nodes can communicate BS directly.
4. Sensor nodes are densely and randomly deployed over the sensor field.
5. All the sensor nodes are mobile and in continuous motion.
6. All sensor nodes performing same tasks.
7. BS position is fixed.
8. Sensor nodes moving with some mobility model.
9. Number of CHs is not fixed

3.3 LEACH-MAE Algorithm design

In our proposed algorithm, each round consists of two phases *Setup Phase* and *Steady-state Phase* same as in basic LEACH protocol.
At first, during the setup phase each sensor node is eligible to be a CH with some probability P (eq 3.1), after first round CH election process is purely energy based. This CH election is independent of other sensor nodes and number of CHs in the network is based on binomial distribution [7].

$$P = \frac{k}{N} \tag{3.1}$$

Where ***k*** is the required number of CHs and ***N*** is the total number of sensor nodes in the network. These ***k*** CHs are elected initially pseudo-randomly among all nodes in the network. Each sensor node generates a pseudo random number between 0 and 1 in the start and if the number is less than the threshold it is elected as a CH:

$$T(n) = \begin{cases} \frac{P}{1-P\times(r \bmod (1/P))} & n \in G \\ 0 & \text{Otherwise} \end{cases} \tag{3.2}$$

Where ***P*** is the percentage of CHs, ***r*** is current round number.

Each CH then advertises its status to all cluster nodes so that they can join more suitable CH. If non-cluster nodes receive more than one advertisement messages, it selects a CH which has greater RSS .In response of advertisements, cluster nodes sends join message to their respective CH directly. This cluster formation continues until all clusters are formed refer to figure below.

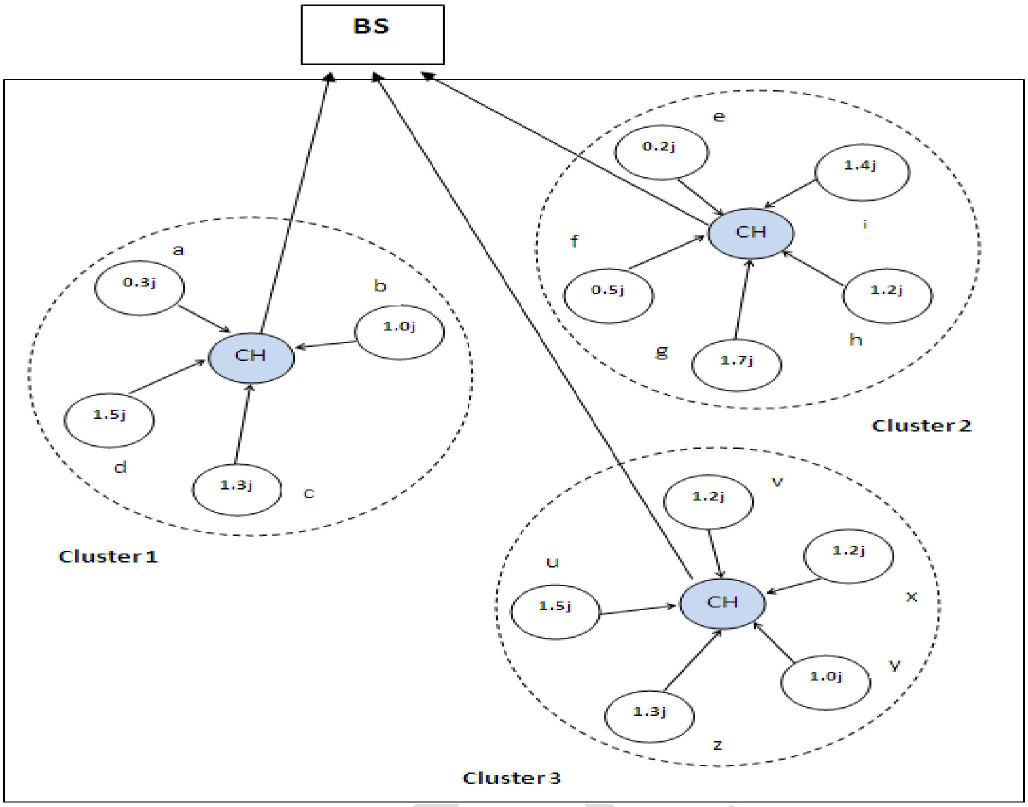


Figure3.2 Cluster formation of First round

Once all clusters are formed, steady-state phase started where each sensor node sends its data to their respective CH and then gathered data from each CH is sent to BS.

In mWSN sensor nodes can leave their clusters at any time due to the high mobility and can join any other cluster described below in figure 3.3

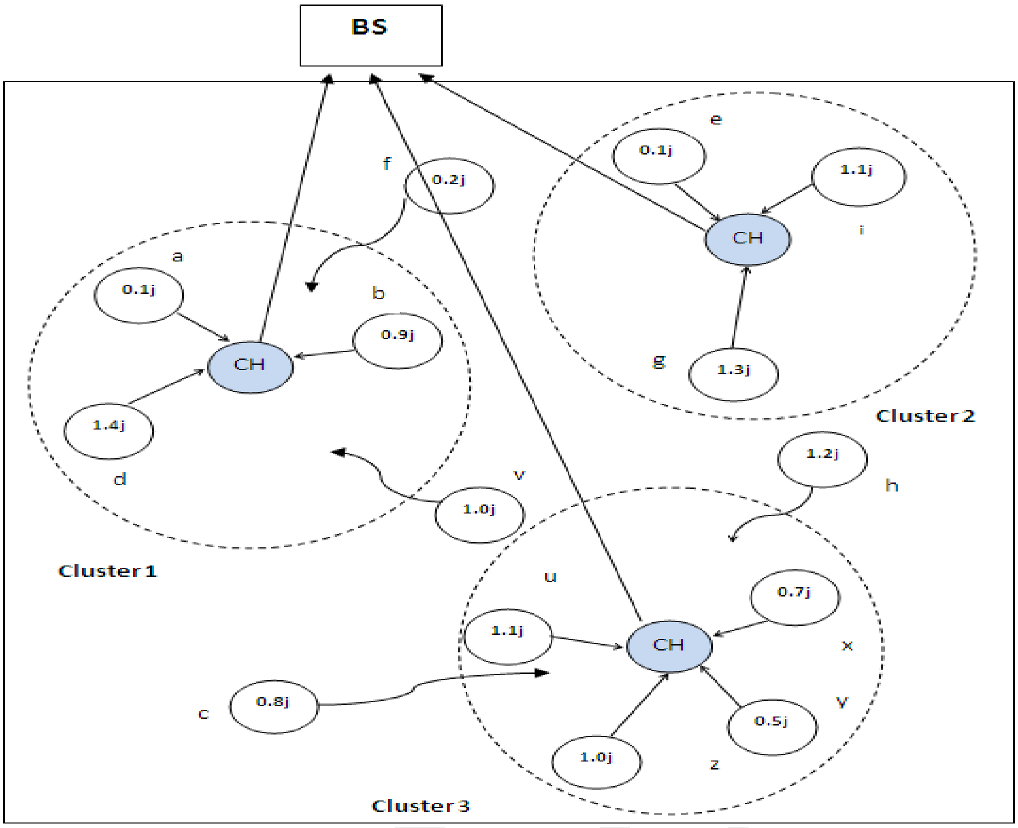


Figure3.3 Mobile nodes changing their location and joining other clusters

After entering to the new clusters, neighbors of sensor nodes are checked to ensure if the nodes have moved out from previous cluster which is shown below in figure 3.4.

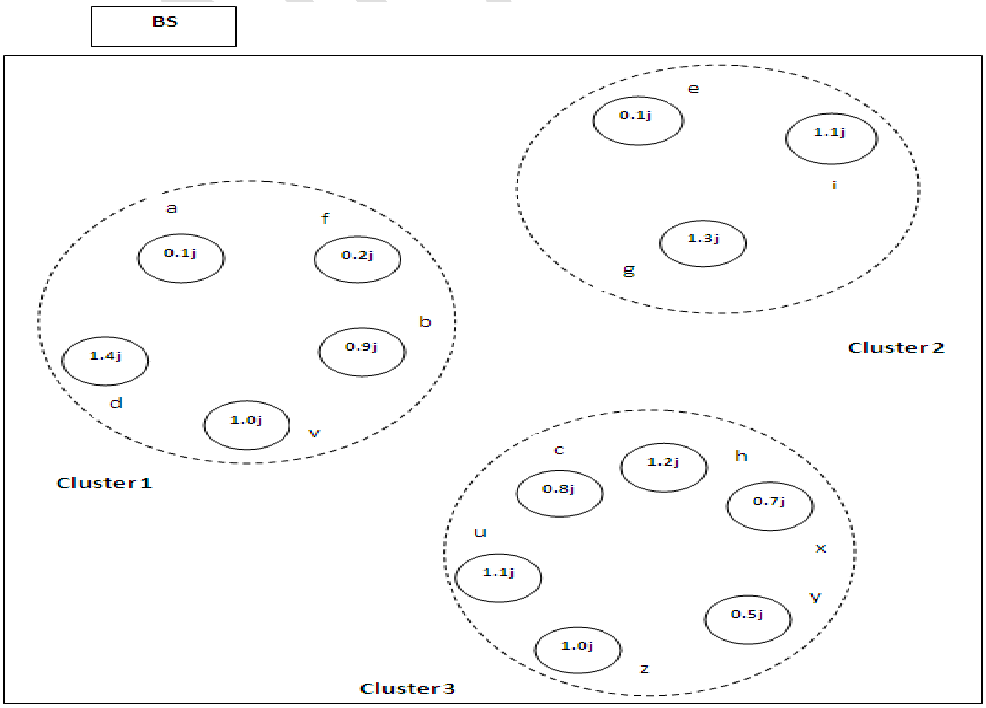


Figure3.4 Neighbors' checking in new clusters

Once all the nodes joined new clusters, nodes having maximum amount of residual energy are elected as CHs for current round shown in figure 3.5

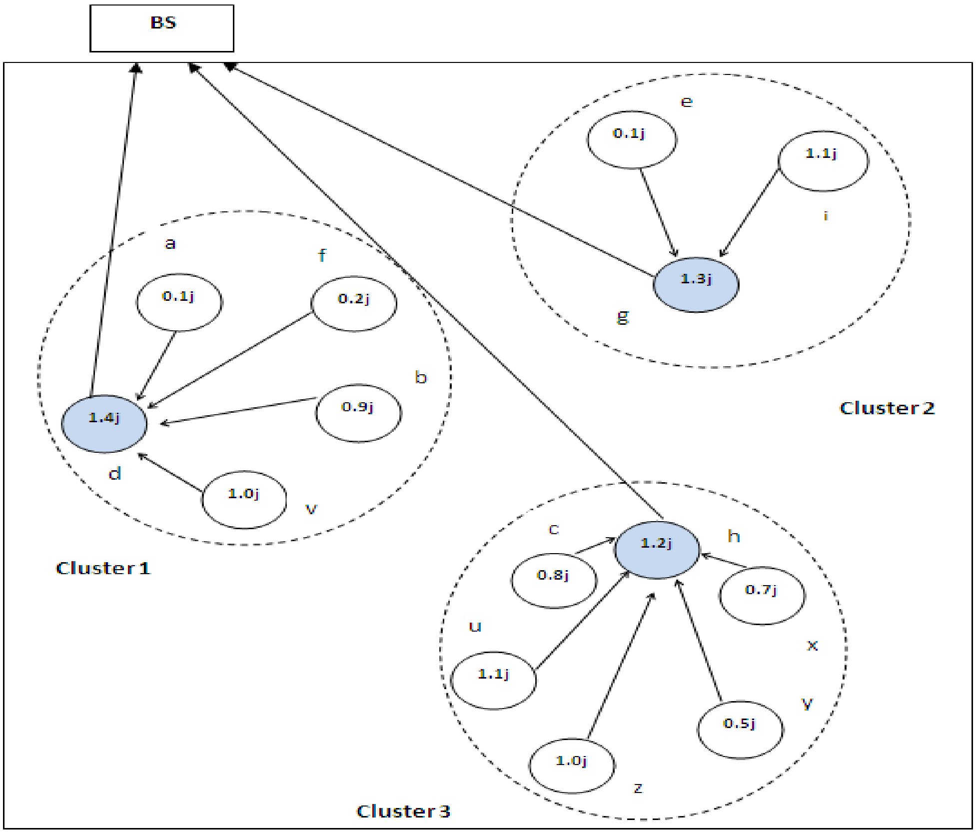


Figure3.5 High Energy nodes are elected as CHs

We proposed that after first round CH election is based on the maximum energy nodes in respective clusters fulfilling the threshold condition.

$$T(n) = \frac{E_{tot}}{CN_{tot}} \tag{3.3}$$

Where ***Etot*** is total energy of all the nodes in one cluster, and ***CNtot*** is total number of cluster nodes.

During setup phase of each round, mobility of sensor node is also taking into account, during each round new TDMA schedules are created by removing the time slots of the nodes which have moved out from the cluster and on the basis of received join requests hence prevents from communication failure. This cluster formation continues till all nodes of the network reaches power failure.

However we compared our simulation results (section 6.3) with LEACH-M protocol using various network scenarios.

On the basis of research studies we analyzed that LEACH-M has some drawbacks:

- There might be some isolated nodes moving far away from the CHs.
- Fixed number of clusters might not be able to facilitate all the other non cluster nodes.
- Random selection of CH nodes doesn't take account of residual energy of nodes. It might cause quick failure of nodes hence result in minimum network lifetime.

3.4 Radio Propagation Model

In mobile networks, transmission between transmitter and receiver can be line-of-sight or can be obstructed due to buildings, trees and mountains. Radio channels are random as compared to the wired channels and speed of a mobile unit effects the signal level. Propagation models are designed to predict the received signal strength at a given distance from the transmitter.

In MWSN two propagation models are used for CH to BS transmission as well as for the cluster node to CH transmission.

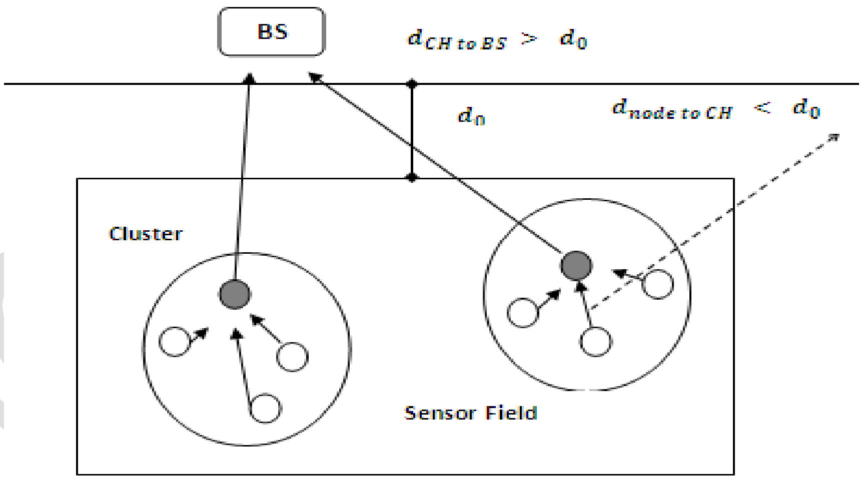


Figure3.6 Sensor Network when BS is located outside the field

In above figure 3.6, a sensor network is represented where BS is located outside the sensor field at some distance and this distance is greater than threshold ***do.*** When data is transmitted within large distance i.e. greater than ***do***, two ray ground reflected model is used.

For our simulation, we used same propagation models as in basic LEACH (section 2.7) protocol.

The free space model is considered with direct line-of-sight propagation and two ray ground models is considered with the ground reflected signals, depending upon the distance between transmitter and receiver.

$$d_o = \frac{4 \times \pi \times \sqrt{L} \times h_r \times h_t}{\lambda} \qquad (3.4)$$

Where ***L=1*** is the loss factor, h_t and h_r are the heights of transmitting and receiving antennas respectively.

For both above models transmit power is described by following equation

$$P_r(d) = \begin{cases} \dfrac{P_t * G_{tr} * G_t * \lambda^2}{(4 * \pi * d)^2 * L} & if\ d < d_o \\ \dfrac{P_t * G_t * G_r * h_t^2 * h_r^2}{d^4} & if\ d \geq d_o \end{cases} \qquad (3.5)$$

Here ***Pr*** and ***Pt*** are received and transmitted power respectively at distance d, ***Gt*** and ***Gr*** are transmitting antenna gain and received antenna gain.

3.5 Energy Dissipation Model

For the Sensor network represented in figure 4.5 when data is transmitted within the distance i.e. greater than d_o, for the transmission of ***k*** bits message. In LEACH [1] a simple radio energy dissipation model is assumed, where energy is dissipated due to transmitter electronics and power amplifiers on the transmitter side and on radio side by receiving electronics.

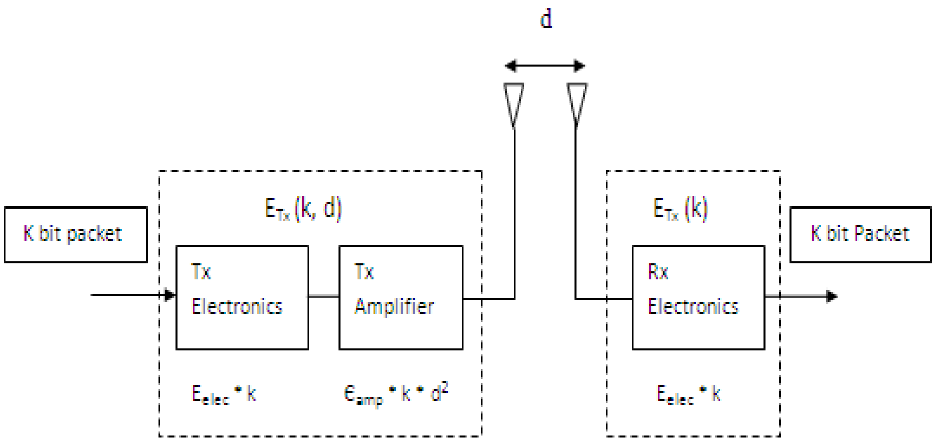


Figure3.7 Energy Dissipation Model

To transmit a ***k*** bit message at distance ***d***, this radio model is used following equation:

$$E_{Tx}(k,d) = E_{elec} \times k + \epsilon_{amp} \times k \times d^2 \tag{3.6}$$

and k-bit message is received by following:

$$E_{Rx}(k) = E_{elec} \times k \tag{3.7}$$

It is analyzed that communication cost is higher in terms of energy consumption at large distances.

Our proposed algorithm helps to conserve energy through more than k clusters formation. This technique not only minimizes the distance but also helps to maintain the communication among all network nodes. Due to the mobility, more energy is depleted and it's hard to communicate all the nodes to the BS at a greater distance. Our proposed algorithm design for mobile sensor nodes reduces transmission costs. Data aggregation at CH combines all the packets from cluster nodes to produce a single packet stream therefore minimizes the energy cost consumption.

3.6 LEACH-M Algorithm Design

Basic LEACH [3] is designed for static WSN due to which in mobility centric environment it causes many problems such as efficient energy resource allocation, mobility handling and self-organization. Therefore an extension of LEACH-M has been proposed in basic LEACH to handle the mobility in sensor nodes. LEACH-M [28] uses same set-up phase as in basic LEACH. All the sensor nodes are divided into multiple clusters where one node is acting as CH. CH position is random and rotated among all the nodes but due to overloading, higher data processing and maximum data transmission these CH die out more quickly. In MWSN, nodes can leave the cluster anytime and join another which causes serious network issues. Once all the nodes join their clusters CH maintain the TDMA schedule for each cluster node to transmit data.

LEACH-M ensures the communication between nodes which are deployed randomly over the field.

Although LEACH-M has some drawbacks discussed in section (3.3).

CHAPTER 4

RESEARCH METHODOLOGY

4.1 Introduction

This section describes the techniques and methods for the design and implementation of the software simulation used to carry out the whole research work. Various methodologies can be used to analyze the WSNs routing protocols. For our research work we used literature survey about the WSN and MWSNs protocols specifically *hierarchical routing protocols* discussed in previous section and software simulation to analyze the behavior of MWSN for the proposed algorithm.

4.2 Simulation of WSN

WSN are quite latest technology and evolving day by day due to which experiment equipments are not commonly available, hence to analyze them software simulation is only best approach in research. It provides the prevailing technique for the study of computer network's behavior. To observe the behavior of our proposed algorithm and network performance of routing protocol, simulation experiments has been carried out by generating different network scenarios. Routing protocols with the basic algorithm and with proposed algorithm were implemented and observed on the software.

4.3 Network Simulators

To analyze the behavior of different routing protocols and the performance of wired or wireless network, network simulators play a significant role in research studies. Network simulation provides generalization, cost effectiveness, easy installation, and reliability.

To make research studies easier many simulation tools have been introduced such as NS-2, TOSSIM, OpNet, Omnet ++, SenSe, GloMoSim, SENSE, J-sim and NetSim [10].

In our thesis protocol implementation and evaluation is based on simulations using NS-2. As NS-2 is open source software, it is more popular tool compared to the other simulators. Following section discuss brief introduction of NS-2 and its working schemes.

4.4 Network Simulator 2 (NS-2)

It is one of the widely used software for the evaluation of routing protocols. As it is free of cost software it can be downloaded easily. It can be run in UNIX and Windows but to run NS-2 in Windows, Cygwin software is required. For our thesis we used Red hat and CentOS operating system for NS-2 installation.

It is discrete event simulator and based on two programming languages: an object oriented simulator which is written in C++, and OTcl (Object Tool Command Language) interpreter which is extension of Tcl (Tool Command Language), used to run command scripts.

For our simulation, we worked on OTcl scripts, which are the programs to initiate the event scheduler, to set up network topology, to generate traffic, and to set the start and stop time for packet transmission.

NS-2 works as interpreter for OTcl and sets environment parameters for the executed OTcl script. As output of the simulation one or more trace files are generated containing details about the simulation data, this traced information can be analyzed by directly ,by using Network Animator (NAM) or XGraph tools provided by NS-2.[9]

4.4.1 NS-2 Simulation

Simulation methodology in NS-2 can be defined by [11]

Simulation design: User describes the purpose of the simulation, network configuration, assumptions, performance measures and output format and type to initiate the network simulation.

Configuration and execution: It is divided into two phases

Phase1: Here all the network components: protocols, network model, and proposed scenarios are implemented according to the system design. Also scheduling of events like data transmission and reception, start and stop time of simulation are scheduled.

Phase2: In this phase, simulation starts according to the configuration described above. It keeps up simulation time and executes all the configured events until the end time of simulation if reached.

NS-2 provides many great features:

- Queue management techniques e.g. Drop tail, RED, FQ etc
- Multiple traffic types e.g. CBR, VBR
- Unicasting and multicasting
- Wireless network modeling for WLAN,GPRS, cellular, IEEE 802.11
- Network topology management
- Packet flow
- Packet tracing
- Applications such as FTP, Telnet.

4.4.2 NS-2 Packet Tracing

Packet tracing is provided by NS-2 by recording packet flow in details during the whole simulation. NS-2 provides text based packet tracing and a NAM packet tracing. Only trace file is not sufficient, one must have to extract desired data from the trace file as this trace file contain some irrelevant information. [9]

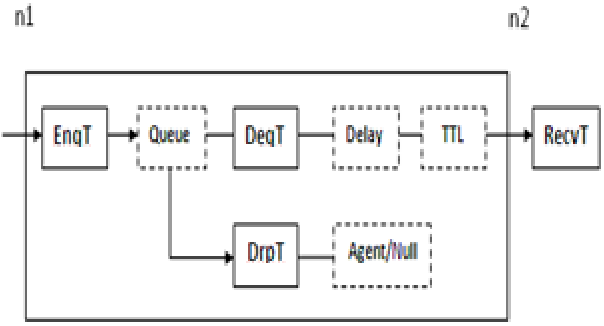


Figure4.1 Packet Tracing

NS-2 provides two trace format old trace format Figure and a new trace file format figure respectively

```
s 0.000079587 _92_ AGT  --- 0 rca 2 [0 5c000000 0 0] ------- [A 92 -1 -1]
s 0.000084588 _86_ AGT  --- 1 rca 2 [0 56000000 0 0] ------- [A 86 -1 -1]
s 0.000109811 _11_ AGT  --- 2 rca 2 [0 b000000 0 0] ------- [A 11 -1 -1]
s 0.000132910 _22_ AGT  --- 3 rca 2 [0 16000000 0 0] ------- [A 22 -1 -1]
r 0.000305597 _53_ AGT  --- 0 rca 2 [0 5c000000 ffff0008 0] ------- [A 92 -1 -1]
r 0.000305616 _63_ AGT  --- 0 rca 2 [0 5c000000 ffff0008 0] ------- [A 92 -1 -1]
r 0.000305620 _95_ AGT  --- 0 rca 2 [0 5c000000 ffff0008 0] ------- [A 92 -1 -1]
r 0.000305628 _85_ AGT  --- 0 rca 2 [0 5c000000 ffff0008 0] ------- [A 92 -1 -1]
r 0.000305629 _21_ AGT  --- 0 rca 2 [0 5c000000 ffff0008 0] ------- [A 92 -1 -1]
r 0.000305635 _32_ AGT  --- 0 rca 2 [0 5c000000 ffff0008 0] ------- [A 92 -1 -1]
r 0.000305637 _15_ AGT  --- 0 rca 2 [0 5c000000 ffff0008 0] ------- [A 92 -1 -1]
```

Figure4.2 NS-2 Old Trace Formats

Old trace format provides a 12 column pattern of a trace file. In post analysis phase desired data is extracted to analyze further

Event Type	Time	Source Node	Dest Node	Packet name	Packet Size	Flags	Flow ID	Source Address	Dest Address	Seq Number	Packet Unique ID

Figure4.3Line Format Of a Trace File

New trace file format of NS-2 contains more information then old trace format.

```
r -t 0.122447268 -Hs 28 -Hd -1 -Ni 28 -Nx 136.44 -Ny 328.35 -Nz 0.00 -Ne 9.998925 -Nl RT
r -t 0.122447274 -Hs 21 -Hd -1 -Ni 21 -Nx 203.59 -Ny 496.22 -Nz 0.00 -Ne 9.998656 -Nl RT
r -t 0.122447329 -Hs 100 -Hd -1 -Ni 100 -Nx 40.81 -Ny 676.50 -Nz 0.00 -Ne 9.999194 -Nl R
r -t 0.122447361 -Hs 89 -Hd -1 -Ni 89 -Nx 106.50 -Ny 276.50 -Nz 0.00 -Ne 9.998925 -Nl RT
r -t 0.122447428 -Hs 19 -Hd -1 -Ni 19 -Nx 209.40 -Ny 606.12 -Nz 0.00 -Ne 9.998387 -Nl RT
```

Figure4.4New Trace Format

This new trace format of NS-2 have following fields as mentioned in [11] :

Event Type: It describes the type of the event occur at the node, it can be

-s Send

-r receive

-d drop

-f forward

Time: It is the time at which one of the above mentioned event occur

Node Characteristics: It denotes node-id, node-x,y co-ordinates' information, node's energy etc. The tag starts with ***–N*** as follows:

-Ni Node id,

-Nx node's x-coordinate

-Ny node's y-coordinate

-Ne Energy level

IP level packet information:

-Is Source Address: Source Port

-Id Dest address: Dest port

-Il size

-If flow id

-Ii Unique id of packet

Next Hop Information:

-Hs Current node id

-Hd next hop id

Packet detail at MAC:

-Ma: duration

-Md   dest physical address

-Ms   source's physical address

-Mt   Ethernet Type

4.4.3 Network Animator (NAM)

NS-2 provides an important tool which helps to visualize the network topology. It can be directly executed from the Tcl scripts, presenting the the information about throughput and number of packets on link. It is easy to use as it provides drag and drop facility to create topology directly in the window.

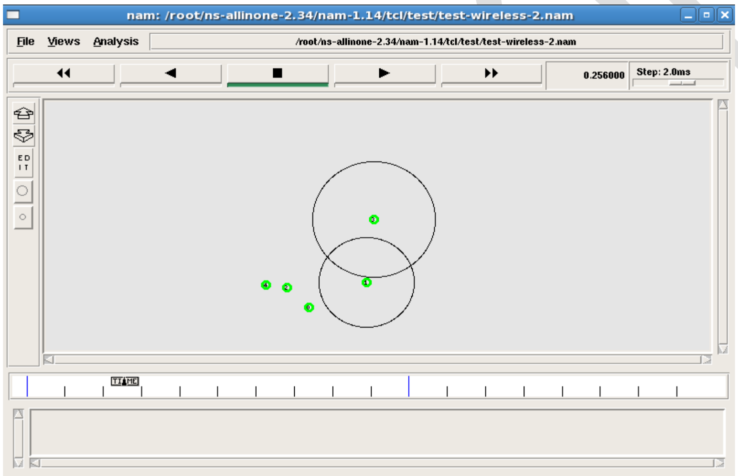


Figure4.5 Network Animator Window

4.4.4 Trace Data Analyzer

It is to analyze the traced data from an output file, Xgraph utility helps to generate graphs, Microsoft excel sheets also can be used to analyze the data.

4.4.5 Data Extraction Tools

In our thesis, to extract desired data from the trace file for our proposed algorithm we used:

Awk scripts: to calculate the desired data

Grep, Perl: Helps to filter the necessary information

CHAPTER 5

IMPLEMENTATION AND SIMULATION

5.1 Chapter Outline

In chapter 3, the design of proposed clustering algorithm is explained. The simulation of proposed algorithm and implementations are described in current chapter. First, we simulated network model for the basic algorithm, based on the drawbacks findings we made some improvements and results are analyzed for different scenarios.

Our proposed algorithm was tested for different number of nodes, different field size, speed variations of sensor nodes and for variable pause time during the movement of the nodes over the field. It is also important to note that there is no fixed number of clusters and the size of the cluster is also not fixed.

For the analysis of both algorithms we simulate each scenario ***20*** times then on the basis of average values we generated ***95% CI*** graphs. The results demonstrated that proposed algorithm results in better network performance.

5.2 Implementation of LEACH protocol

As discussed in chapter 3 we have proposed our algorithm on the basis of LEACH protocol. Therefore we first implemented LEACH protocol using the MIT's LEACH patch (section 3.7) in NS-2.

To implement it successfully some changes have been made within multiple files refer to (APPENDIX I) hence it has been guaranteed that LEACH is working fine giving out the results as in [1].

For the results comparison analysis we used following parameters:

Table5.1 Main Parameters of Routing Protocols

Parameters	Description
Antenna Model	Antenna/Omni Directional
Channel Type	Channel/Wireless Channel
Radio Propagation model	Two Ray Ground
Interface queue type	Queue/Drop tail/Priqueue
Link Layer type	LL
Communication model	Bi-Direction
IFQ length	50 packets
Simulation time	200 sec

To analyze the behavior of proposed algorithm multiple scenarios were generated with parameters' variations described in table below:

Table5.2 Simulation parameters

Parameters	Scenario 1	Scenario 2	Scenario 3
Field Size	100m *100m	100m*100m	50m*50m
BS Location coordinates (x,y)	50m,175m	50m, 175m	25m, 75m
Number of Nodes	100	50	100
Initial energy	2j	2j	2j
E_{elec}	50 nJ/bit	50 nJ/bit	50 nJ/bit
E_{amp}	0.0013 pJ/bit/m^4	0.0013 pJ/bit/m^4	0.0013 pJ/bit/m^4
Packet Size	1460 bits	1460 bits	1460 bits

In order to analyze the network behavior we used same energy level for all the nodes to introduce homogeneity. Eelec is energy dissipated from transmitter and receiver electronics. E_{amp} is energy dissipated when packet is amplified to the destination. For LEACH-M and LEACH-MAE we used a spreading factor of ***8***.

5.3 Implementation of Mobility models

For mobile sensor networks it is important to note that sensor nodes are moving through some mobility model discussed in section (2.8).

As the basic LEACH protocol is for static nodes, we generated mobility scenario so that mobility in sensor nodes is achieved. We made multiple variations in speed of sensor nodes and their pause time as described in table below:

Table5.3Pause time for Different Node Speed

Protocol	Pause Time			Speed of Sensor Nodes				
LEACH-M	0 sec	1 sec	5 sec	5 m/s	10 m/s	20 m/s	30 m/s	40 m/s
LEACH-MAE	0 sec	1 sec	5 sec	5 m/s	10 m/s	20 m/s	30 m/s	40 m/s

To generate mobility scenarios with the parameters described in table 5.2, we used ***setdest*** tool provided by NS-2 refer to (Appendix III).

The output file contains the information about the initial coordinates of all the network nodes, and information about their next change location along with the speed of each individual node. As LEACH uses single hop technique it is necessary that each node must be at the distance of one hop to each node.

5.4 Implementation of Topological files for different Number of Nodes

In NS-2, topology files are created separately for the different number of nodes using a tool named ***genscen*** refers to (APPENDIX II)

This tool helps to generate initial topology of sensor nodes over the field and is used by main Tcl script for simulation.

5.5 Implementation of LEACH-MAE algorithm

In previous chapter we discussed the design of our proposed algorithm LEACH-MAE. As LEACH-MAE is based on LEACH protocol therefore it uses same CH election technique for the initial round. All the nodes are divided into multiple clusters with one node acting as a CH. Initially the selection of CH is random due to the homogeneity of nodes but in further rounds it is to be ensured that only high energy nodes can be elected as CHs. LEACH-MAE extends the LEACH-M algorithm by further introducing mobility model for all the sensor nodes.

For our research work we used MIT code of LEACH and modified further for LEACH-MAE algorithm refers to APPENDIX IV. We used multiple scenario files and different tools of NS-2 to implement our algorithm

CHAPTER 6

RESULTS AND PERFORMANCE ANALYSIS

6.1 Overview

In cluster based routing protocols there is no standard trademark to evaluate the performance of WSNs. In general research studies some of the parameters are used to determine the effect of hierarchical routing protocols as follows: Total number of alive nodes in each round, energy consumption at the end of network lifetime and packet delivery ratio. For energy aware protocols the time of first dead node appears is important to estimate the network lifetime. In these networks, working time is defined as the duration between the start of node's activity and the death of first node [1].

In this chapter we discussed the findings of our research work based on the proposed algorithm design and selected mobility model. We used different simulation parameters described in (section 5.2 and 5.3), and experiment metrics to analyze the network performance.

6.2 Experiment metrics

As described in section 2.7, Heinzelman et al [1] used energy consumption and network lifetime metrics for the analysis of their simulation experiments which is also used for LEACH-M performance analysis in [29]. For our simulation we used energy consumption of network and the speed of all mobile nodes in the network.

In WSN, one of the important factors is Energy consumption of each individual node, as the network lifetime is the time of last node death i.e. the complete energy depletion of the last active node in the network. In sensor networks batteries of sensor nodes cannot be recharged or replaced, therefore research is aiming to prolong the network lifetime.

In our simulation experiments, network lifetime is defined as the time until all the sensor nodes are dead.

The speed of mobile nodes is important because when the nodes moves continuously in random directions with random speed there is a possibility that they are unable to communicate to CH or to the BS due to increasing and decreasing distance as well faces the delay or failure of data transmission. When the sensor nodes moves with certain speed they consume more energy hence are more likely to quick death. When the speed is varied with some pause time, cluster nodes can send more data, and can consume less energy.

6.3 Scenario 1 for 100 nodes placed over (100m*100m)

6.3.1 Experiment A

Continuous nodes' movement

In this scenario, we used 100 mobile sensor nodes placed randomly over the field with the initial energy of 2J per node. The movement of sensor nodes is continuous without introducing pause time as described in section 5.3 (table5.3).

Figures below compare the energy consumption of LEACH-M and our proposed algorithm for different node's speed.

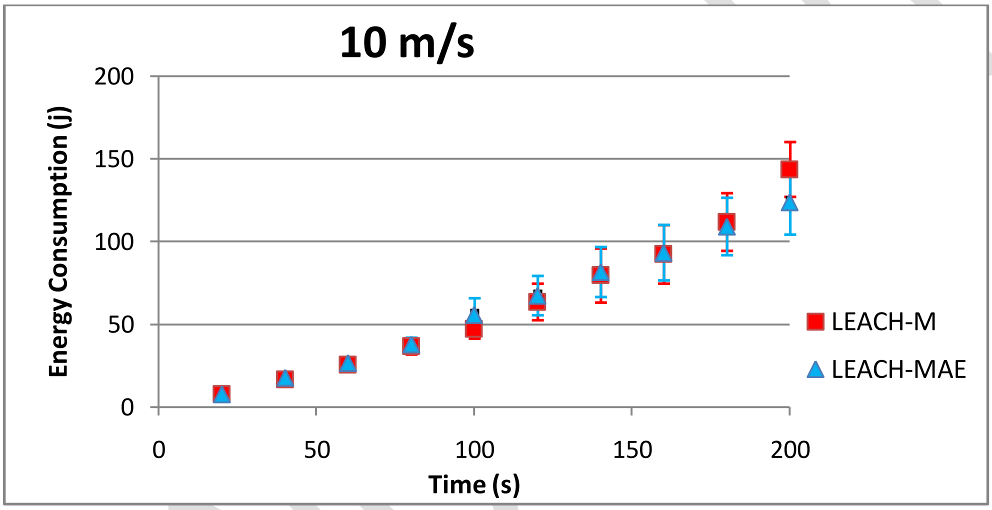


Figure 6.1 Energy Consumption at 10m/s for 0sec Pause Time

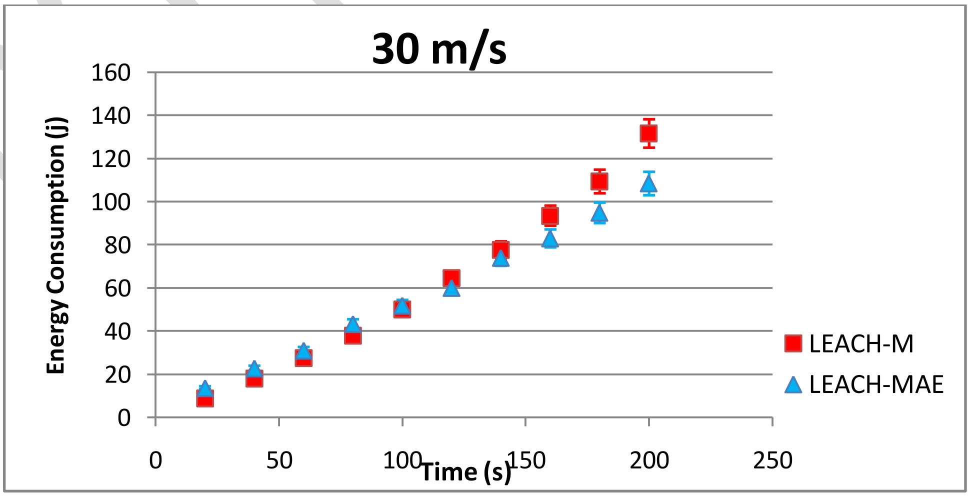


Figure6.2 Energy Consumption at 30 m/s for 0sec Pause Time

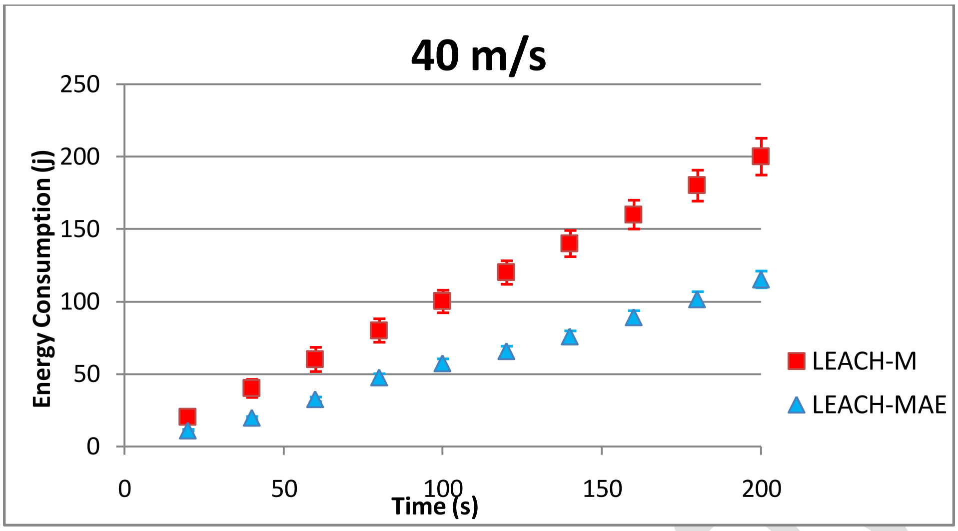


Figure6.3 Energy Consumption at 40 m/s 0sec Pause Time

The amount of energy consumption is less in our proposed algorithm than in LEACH-M because of the CH selection procedure which avoids random selection of CH even in high mobility scenario. TDMA based scheduling also helps to overcome the energy dissipation issue by avoiding the overhearing [26]. As the energy consumption of a network is inversely proportional to the network lifetime. CH selection on the basis of more residual energy helps to improve the performance of network.

We calculated the percentage difference of energy consumption by using:

$$Percentage\ Difference = \left| \frac{E_{LEACH_{MAE}} - E_{LEACH_M}}{E_{LEACH_M}} \right| \qquad (6.1)$$

Where $\boldsymbol{E_{LEACH_{MAE}}}$ is the energy consumption of our proposed algorithm and $\boldsymbol{E_{LEACH_M}}$ is the total energy consumption of LEACH-M at the end of simulation time.

The figures below shows the throughput rate of both algorithms when all the mobile nodes are moving with the maximum speed. The amount of delivered data is far better in our algorithm as all the mobile nodes are able to communicate with the CH even with the high speed which helps to minimize the packet loss.

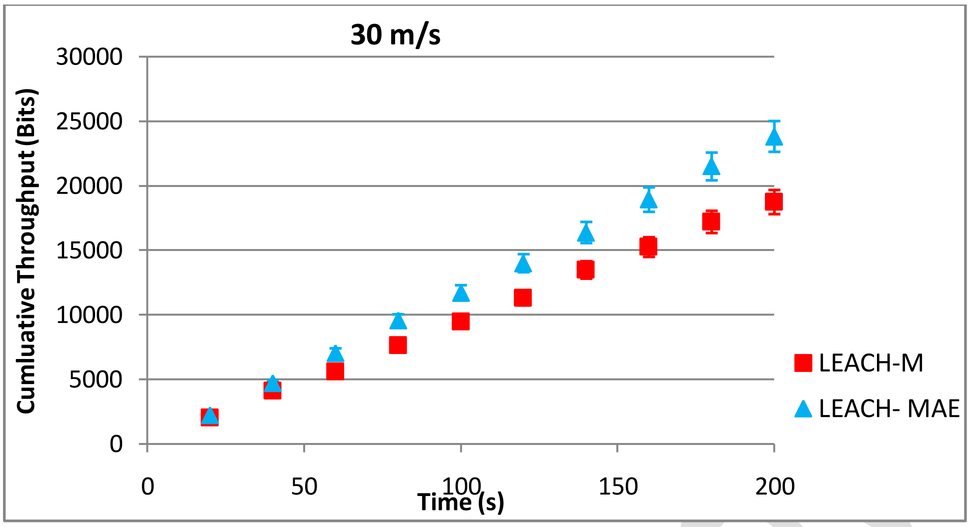


Figure6.4 Cumulative Throughput at 30 m/s for 0sec Pause Time

When high energy node becomes cluster head, there are fewer chances for the quick death due to energy depletion which helps to maintain the communication between the nodes therefore data is successfully transferred to the destination without any loss (resulting in high throughput for LEACH-MAE protocol). Further use of random waypoint and congestion avoidance mechanism also helps to achieve maximum throughput [23].

The figures below describe the number of nodes alive during the network operation. As the network lifetime depends upon the first dead node or last alive node therefore we used remaining number of nodes alive in the network with time.

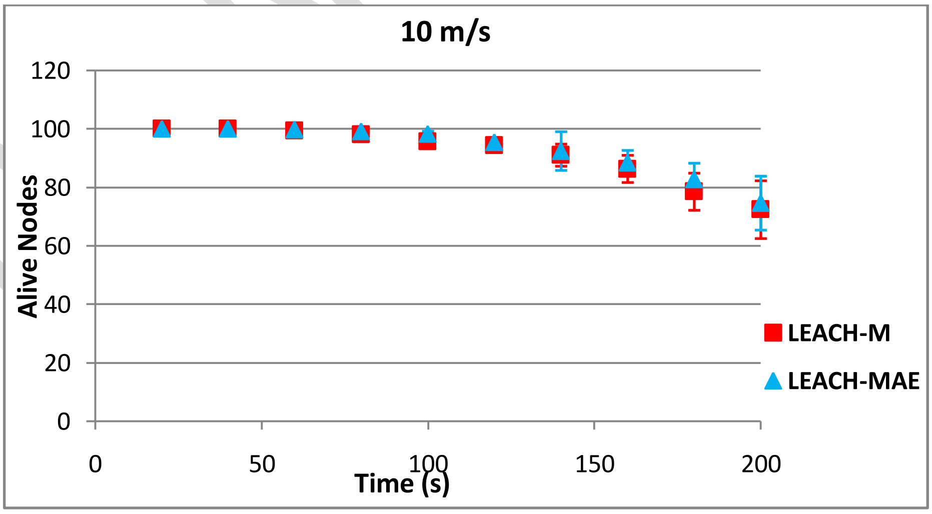


Figure6.5 Number of Alive nodes at 10 m/s for 0sec Pause Time

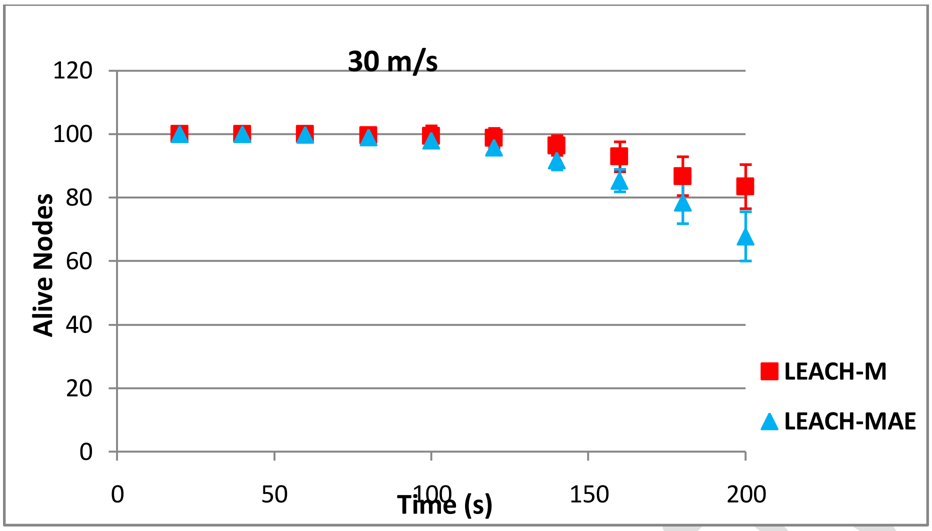


Figure6.6 Number of Alive nodes at 30 m/s for 0sec Pause Time

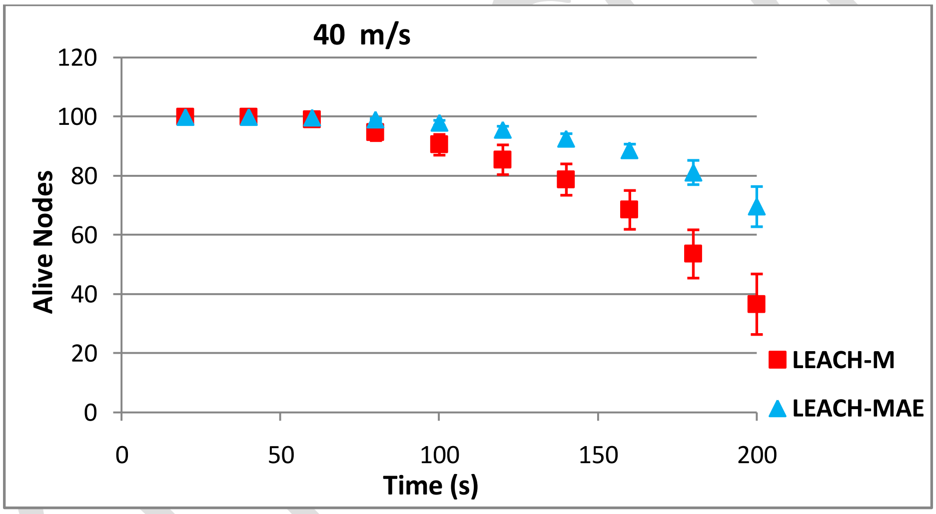


Figure6.7 Number of Alive nodes at 40 m/s for 0sec Pause Time

In WSN, energy consumption is directly proportional to the number of alive nodes. Above figures shows that LEACH-MAE provides better throughput rate with minimum energy consumption. It is important to analyze that energy consumption is the cause of another parameter i.e. mobility. Therefore above figures described that for high mobility LEACH-MAE algorithm ensure maximum data transfer hence the major cause of node's death is the amount of data delivered to the destination and not only speed of mobile nodes.

6.3.2 Experiment B

Nodes' movement with pause time of 1sec

Here we set pause time parameter for all mobile nodes so that each node will stop its movement for the minimum time of 1sec. According to RWPM during this pause time all the mobile nodes chooses new destinations and moves with the constant speed towards new destination points[20].

In figures below we analyzed the performance of mobile nodes using three performance metrics i.e. energy consumption, throughput and number of alive nodes for different speeds with certain pause time.

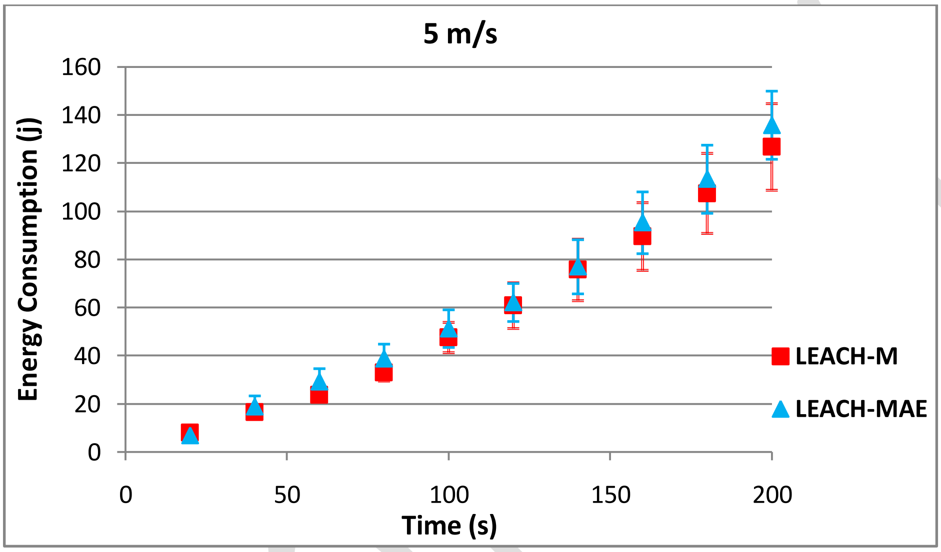


Figure6.8 Energy consumption 5m/s for 1sec Pause Time

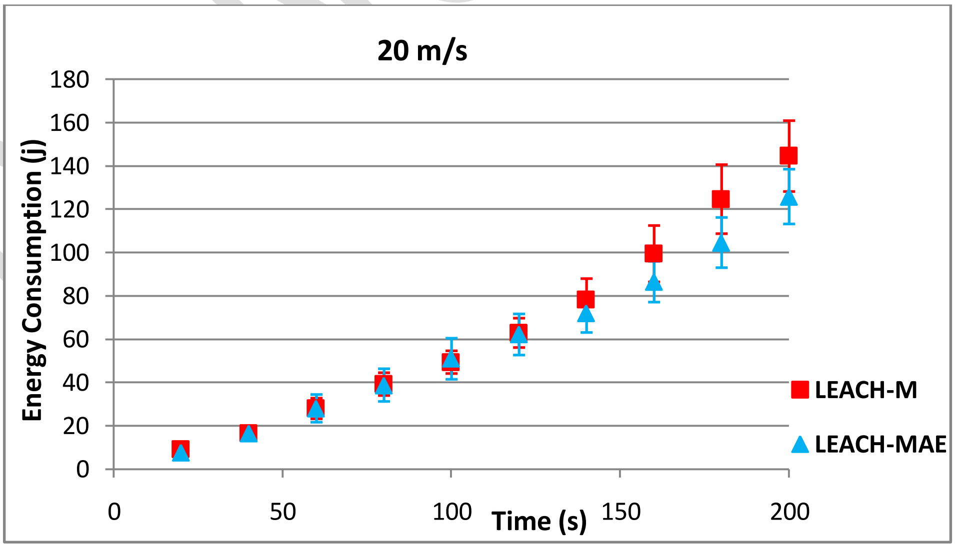


Figure6.9 Energy consumption at 20 m/s for 1sec Pause Time

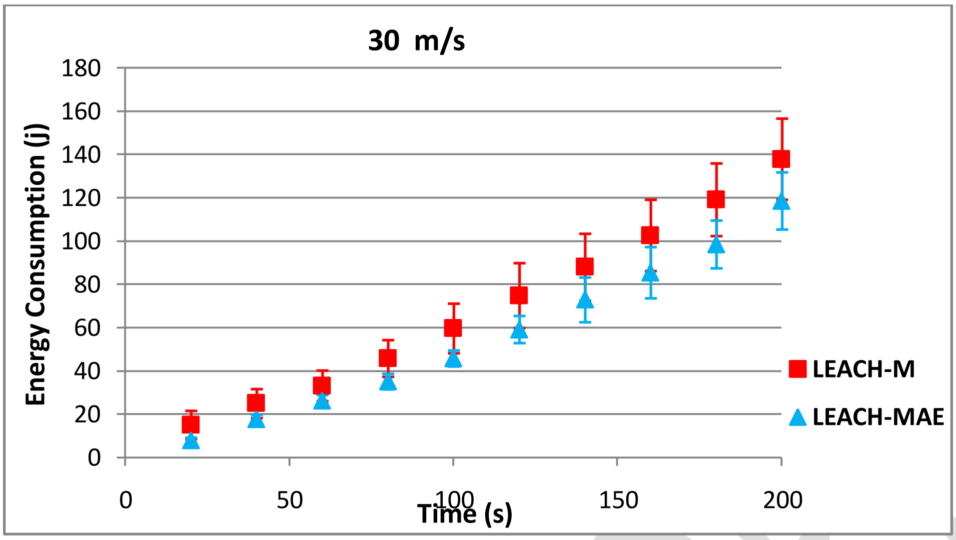


Figure6.10 Energy consumption at 30 m/s for 1sec Pause Time

Above figures shows even if there is a partial static behavior of nodes, Energy consumption is less in LEACH-MAE.

In figures below we analyzed the throughput rate with time. This throughput rate in our LEACH-MAE proves the improved results by ensuring the maximum packet delivery amount as compared to LEACH-M.

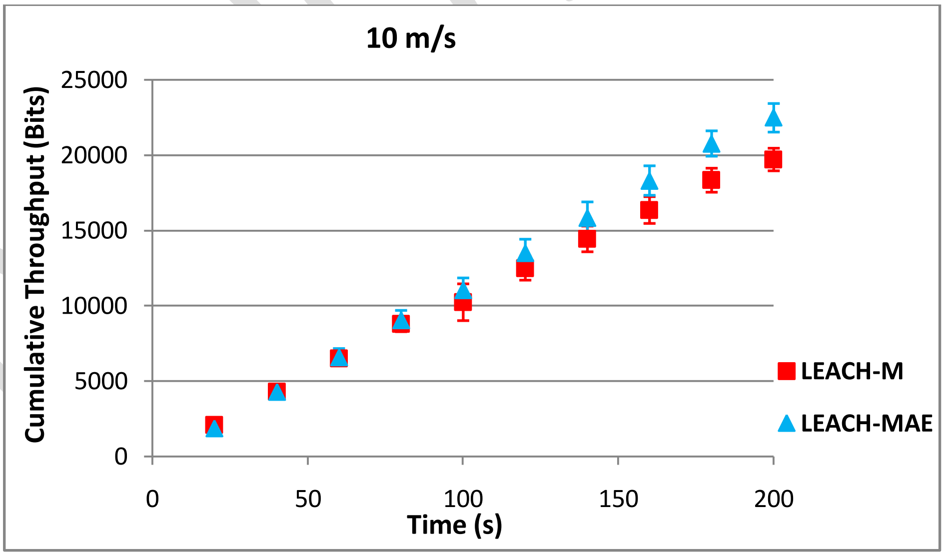


Figure6.11 Cumulative Throughput at 10 m/s for 1sec Pause Time

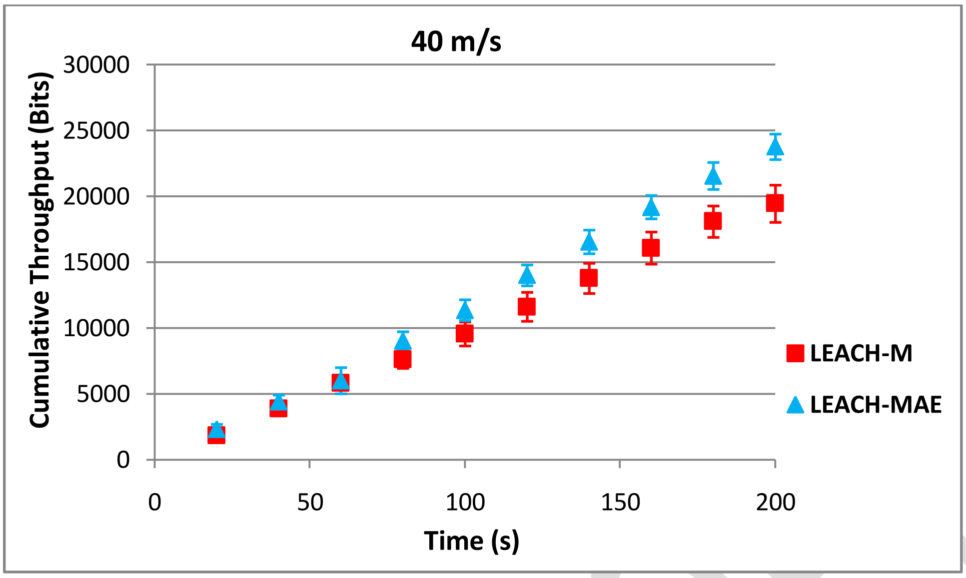


Figure6.12 Cumulative Throughput at 40 m/s for 1sec Pause Time

Since mobility of sensor nodes cause greater link breakage, the appropriate mobility model helps to maintain the links for a long time and in cluster based protocol inter-cluster communication minimizes the data loss due to minimum distance.

Here in figures below network performance has been analyzed in terms of number of alive nodes with time using the parameters defined in table 5.2.

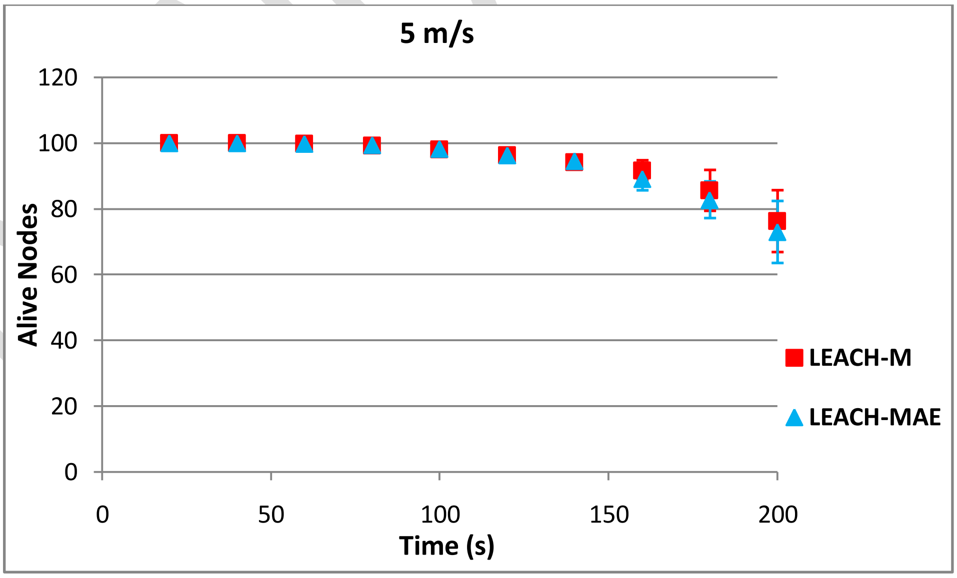


Figure6.13 Number of Alive nodes at 5 m/s for 1sec Pause Time

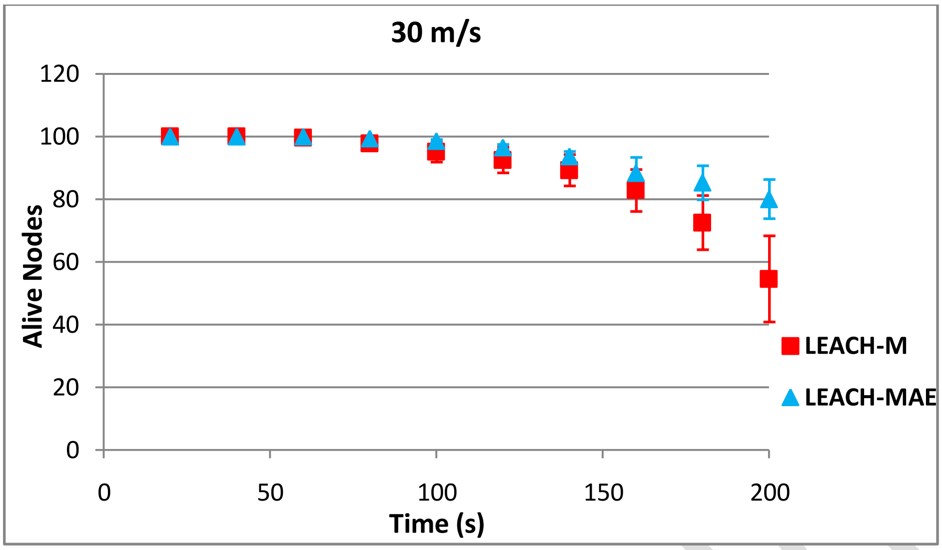


Figure6.14 Number of Alive nodes at 30 m/s for 1sec Pause Time

6.3.3 Experiment C

Nodes' movement with pause time of 5sec

In current scenario we increased the pause time for all mobile nodes to 5 sec and analyzed the behavior of network for all the parameters used in table 5.2.

To compare both LEACH-M and LEACH-MAE, figures below are used to analyze the network behavior for both routing algorithms in terms of throughput and number of nodes appearing dead with time.

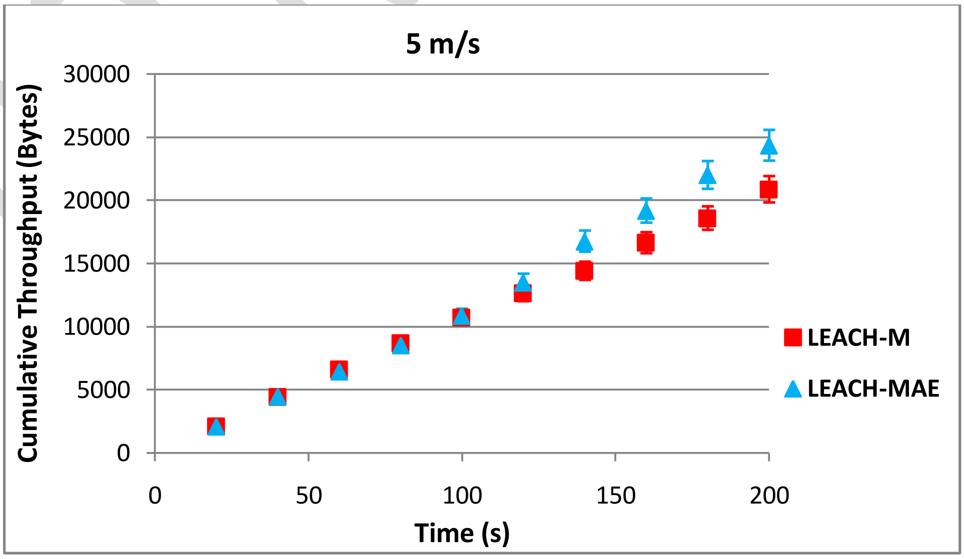


Figure6.15 Cumulative Throughput at 5 m/s for 5 sec Pause Time

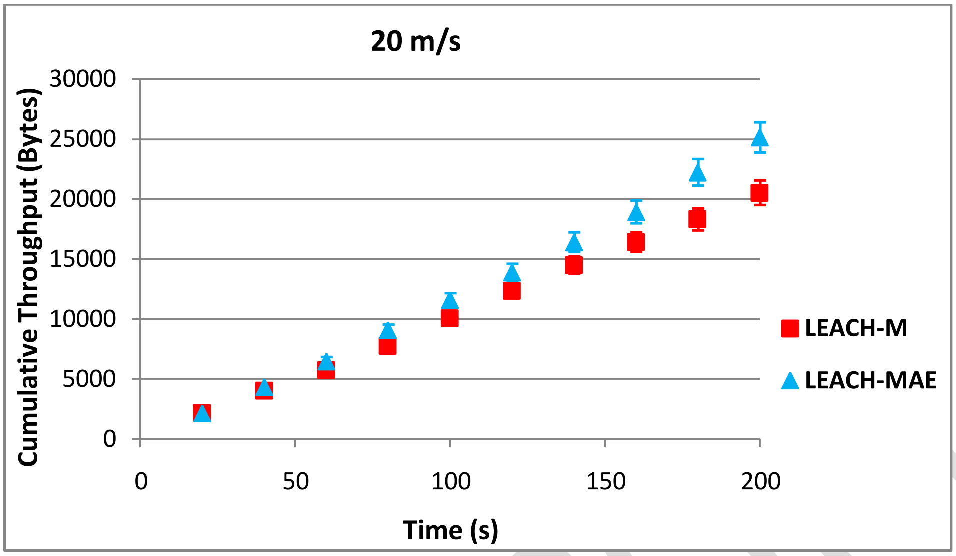


Figure6.16 Cumulative Throughput at 20 m/s for 5sec Pause Time

As already described in section LEACH-MAE introduces minimum link breakage to avoid maximum packet loss, maximum throughput rate has been observed better in all scenarios .

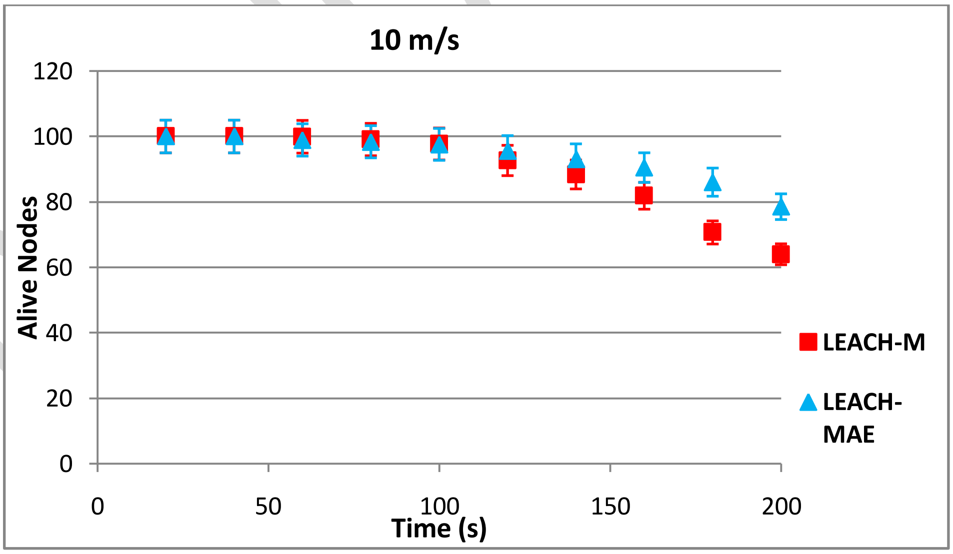


Figure6.17 Number of Alive nodes at 10 m/s for 5sec Pause Time

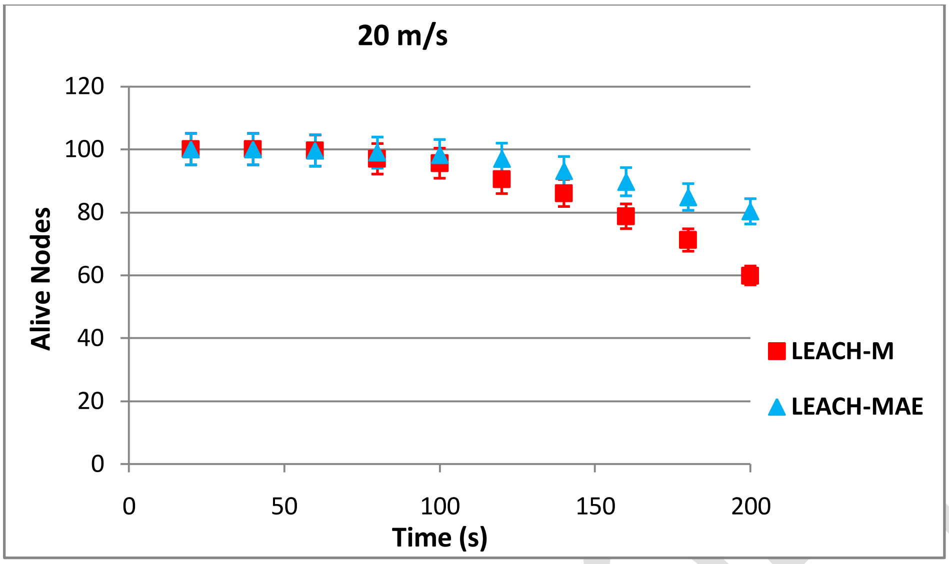


Figure6.18 Number of Alive nodes at 20 m/s for 5sec Pause Time

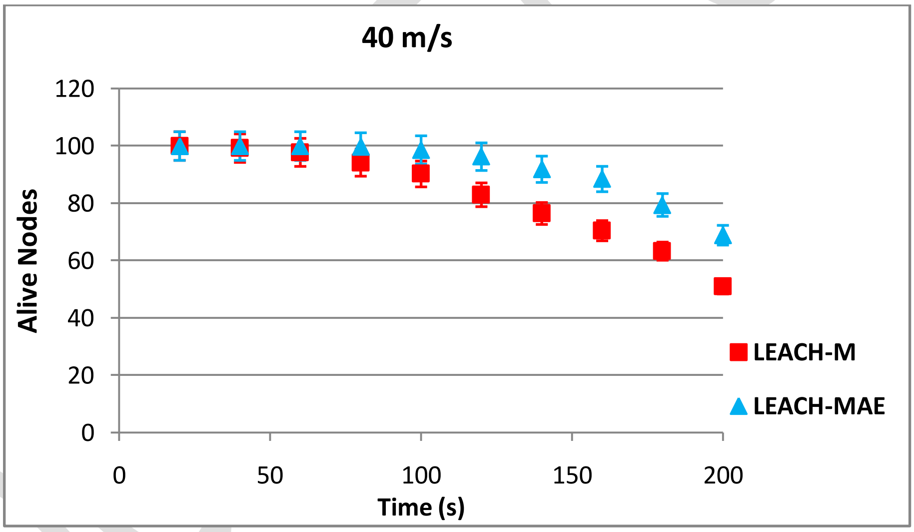


Figure6.19 Number of Alive nodes at 40 m/s for 5sec Pause Time

In this scenario we analyzed that as LEACH-M uses the same technique used in LEACH protocol, random selection of cluster heads can cause maximum link failure due to the probability that low energy level node may be selected which leads to the quick death of nodes in the network. While LEACH-MAE is designed in a way that only high residual energy nodes are eligible to be the CH in every round therefore it is observed that nodes have maximum lifetime and links can be broke down only if the nodes move out of the cluster.

6.4 Scenario 2 for 50 nodes placed over (100 m*100m)

LEACH protocol performance is influenced by the number of nodes and the area on which they are deployed [38]. In this experiment we used 50 mobile nodes deployed over the field area of 100*100m^2 where we assumed that BS is located far away from the field described in table3. As the BS is located at (50m,175m), due to limited amount of energy and wide area these sensor nodes are not able to communicate to the BS. High mobility speed is also observed which causes maximum energy depletion due to which nodes are failed to communicate at a greater distance. Consequently no communication has been observed among the nodes as the maximum amount of energy is consumed while moving over the field leading to the deficiency of power for data transmission.

6.5 Scenario 3 for 100 nodes placed over (50 m*50m)

In previous scenario (section 5.7), It is also observed that LEACH-MAE performs best in terms of all three performance metrics when there is maximum nodes' mobility without pause time.

In current experiment we changed the network field size while keeping the same number of nodes, variable speed (table 5.2) along with no pause time and analyzed the network behavior alike prior experiment.

The following table is used to analyze the network behavior in terms of energy dissipation for variable node speeds assuming maximum mobility is presented in nodes.

Table6.1Energy Consumption (J) Comparison for Both Algorithms

Protocols	**Node's Speed**				
	5 m/s	***10 m/s***	***20 m/s***	***30 m/s***	***40 m/s***
LEACH-MAE	**119.98**	**122.48**	**107.95**	**105.21**	**121.23**
LEACH-M	**115.62**	**103.2**	**112.95**	**97.52**	**104.59**

In above table it is shown that energy consumption in LEACH-MAE is more than LEACH-M because of the data transfer rate shown in table (6.2). As the maximum amount of energy is consumed while transmitting data to BS because there is minimum link breakage due to RWP mobility model.

In following table we analyzed the packet delivery ratio for both algorithms when nodes are moving without any pause time.

Table6.2 Throughput Rate (Bytes) Comparison For Both Algorithms

Protocols	Node's Speed				
	5 m/s	*10 m/s*	*20 m/s*	*30 m/s*	*40 m/s*
LEACH-MAE	21468.35	23076.4	23507.6	24145.05	25159.25
LEACH-M	19317.55	20698.8	17983	20190.75	20268.15

Table6.3 Number Of Alive Nodes Comparison For Both Algorithms

Protocols	Node's Speed				
	5 m/s	*10 m/s*	*20 m/s*	*30 m/s*	*40 m/s*
LEACH-MAE	73.3	71.4	82.95	91.45	90.35
LEACH-M	84.95	95.1	75.6	78.65	86.55

In above table (6.2) it can be seen that throughput rate in LEACH-MAE is greater. The RWP mobility model causes to push the nodes towards center whenever they hit the boundary or other node. Hence the less isolated nodes on the edge of the fields are observed and due link better link maintenance more data is transferred to BS.

In current scenario as field size is small, sensor nodes are deployed densely over the field. Due to congested area, minimum link breakage and minimum mobility is observed among all the nodes therefore less energy is consumed to lengthen the nodes' life time hence both algorithms behaves equally likely to each other.

In MWSN, when the nodes move with the maximum speed links are more likely to break and a rapid topological change in the network is observed. In that situation if CH election is made on random the probability of appropriate CH election will decreased. Mobile nodes consume more energy while moving and hence remaining amount of energy is used for data transfer. For continuous movement with high mobility (i.e. 0 pause time), it is observed that LEACH-MAE compensate the energy resource in a way which helps to support more data delivery and increased network lifetime. Same mobility model (RWPM) also helps to support our LEACH-MAE protocol for the long lasting WSN.

As LEACH-MAE distributes energy resource equally among all nodes and supports high mobility, it also helps to keep nodes alive for a longer time hence improves the network performance. LEACH-M supports mobility but it doesn't provide efficient resource allocation so the number of alive nodes is continuously decreasing with the increase in mobility speed.

In our proposed LEACH-MAE algorithm number of CHs in each round can be increased rationally which ensures that if during the movement any node reached to the boundary, it can communicate to the BS via CH near to that node which prevents from the data loss. LEACH-M defines fixed number of cluster heads in each round which increases the probability that all the nodes will not be entertained by those CHs and the data will be lost.

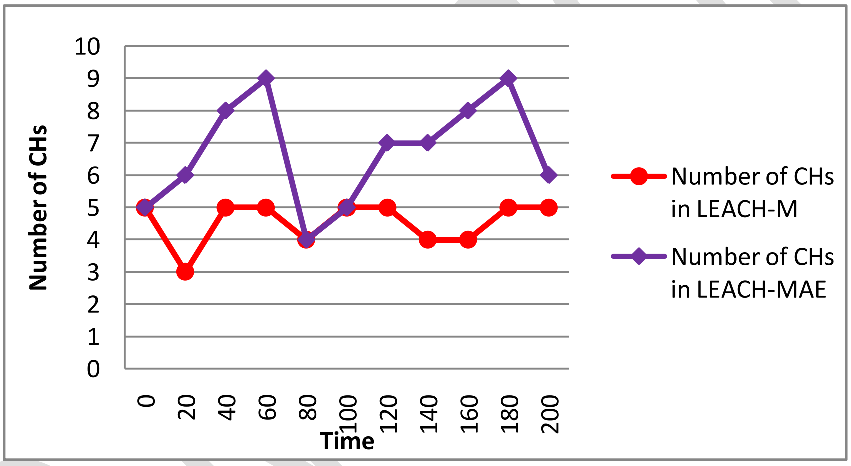


Figure6.20 Comparison between Number of CHs

6.7 Summary

In this chapter we modified LEACH algorithm in terms of LEACH-MAE and showed the simulation results for our proposed algorithm. By evaluating the new criteria for CH election we presented a more effective algorithm for clustering in mobile WSNs. Simulation results shows that network life time in LEACH-MAE is increased as compared to the network life time in LEACH-M. It is more efficient when the speed of mobile nodes is maximum.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The main objective of WSNs is to gather the information from sensors deployed statically and dynamically over the field and transmits this information to the BS. Mobile WSN is a recent advancement comparatively, which adds more applications to meet the challenges. The core operation of WSN for collecting the information is subjected to energy consumption of each individual sensor node. The network life time thus depends upon the amount of nodes' energy dissipated during their movement and communication. Therefore an appropriate routing algorithm is needed to prolong the network lifetime by consuming less energy as well as to ensure the efficient data transmission in the network.

Clustering in WSN is proved to be the best method for improving the network life time. In this thesis we proposed an energy aware clustering algorithm for better network performance. Our algorithm outperforms LEACH and LEACH-M by electing the highest energy mobile node as CH in each round. In LEACH-MAE by considering the mobility of sensor nodes, we reduced the link breakage problem and ensured the highest packet delivery ratio.

The use of mobility model for sensor nodes also helps to improve the network lifetime when node moves in random direction with uniform speed.

The simulation model is designed for mobile sensor nodes which are initially deployed randomly over the field. Simulation results in chapter 6, proves the LEACH-MAE out performs LEACH-M in terms of network lifetime, efficient packet delivery and the amount of energy consumption for high speed sensor nodes.

LEACH-MAE further increases the network lifetime up to 25% and improves the transmission cost and scalability by conserving the more energy.

7.2 Future Work

In our work, we proposed an algorithm which is based on residual energy of nodes. One possibility for improving the current algorithm is to evaluate it for different mobility models. The work can be further extended by choosing the location awareness technique. Another modification which can be done to the current algorithm is to introduce group mobility pattern in order to prolong the network lifetime.

REFERENCES

[1] V. Rajaravivarma, Y. Yang, T. Yang. "An Overview of Wireless Senor Network and Applications" *Proceedings of the 35th Southeastern Symposium on System Theory. 2003. pp. 432-436.*

[2] N Al-Karaki, and A.E. Kamal, "Routing Techniques In Wireless Sensor Networks: A Survey", *IEEE Wireless Communications, Vol. 11, No. 6, pp.6-28, December 2004.*

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Routing Protocols For Wireless Microsensor Networks," *in Proc. 33rd Hawaii Int. Conf. System Sciences (HICSS), Maui, HI, 2000.*

[4] K. Chan, F. Fekri, and H. Pishro-Niki, "Analysis Of Hierarchical Algorithms For Wireless sensor Network Routing Protocols", *IEEE Wireless Communications and Networking Con-ference, vol. 3, pp. 1830 – 1835, 2005*.

[5] K. Ma, Y. Zhang, and W. Trappe, "Managing the Mobility of a Mobile Sensor Network Using Network Dynamics", *IEEE Trans. Parallel Distrib. Syst.2008, pp.106-120.*

[6] Imad Mahgboub and Mohammad Ilyas."Sensor Network Protocols", *Handbook of Sensor networks: Compact wireless and wired Sensing systems published in 2006*.

[7] Sriram Chellappan,"On Deployment and Security in Mobile Wireless Sensor Networks:Algorithms Design, Vulnerability Assessment and Analysis" , *ISBN: 363918257X,pages 7–9.VDM Verlag, July 19,200*

[8] A. Chandrakasan, R. Amirtharajah, Seonghwan Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang, "Design Considerations for Distributed Microsensor Systems", *Proceedings of IEEE 1999 Custom Integrated Circuits Conference (CICC), pp. 279-286, San Diego, CA, May 1999*

[9] L. P. Clare, G. J. Pottie, and J. R. Agre, "Self-Organizing Distributed Sensor Networks", *Proceedings of Society of Photo Optical Engineering (SPIE) Conference on Unattended Ground Sensor Technologies and Applications, pp. 229-237, Orlando, FL, April 1999*.

[10] S. Dai and X. Jing, " Research And Analysis On Routing Protocols For Wireless Sensor Networks," *International Conference on Communications, Circuits and Systems Proceedings,vol. 1, pp. 407–411, 27-30 May, 2005*.

[11] Jennifer Yick, Biswanath Mukherjee and Dipak Ghosal, "Wireless Sensor Network Survey, Computer Networks", *Vol. 52, No. 12, pp. 2292-2330, Aug. 2008.*

[12] G. Santhosh Kumar, Vinu Paul M V, K. Poulose Jacob, "Mobility Metric based LEACH-Mobile Protocol", *IEEE ADCOM 2008*.

[13] S. Duan and X. Yuan, "Exploring Hierarchy Architecture For Wireless Sensor Networks Management" *IFIP International Conference on Wireless and Optical Communications Net-works, p. 6, 11-13 April,2006*

[14] Sri Chusri Haryanti and Riri Fitri Sari," Effect of RPGM and Gauss Markov Node Mobility for Resource Allocation Performance in Mobile Ad Hoc Computational Grid", *IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.6, June 2011.*

[15] S. Lindsey, C. S. Raghavendra, and K. Sivalingam, " Data Gathering in Sensor Networks using the Energy*Delay Metric," *in Proc. of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing, San Francisco, CA, 2002*.

[16] Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison Of Multi-Hop Wireless Ad Hoc Network Routing Protocols", *in Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom98), ACM, October 1998.*

[17] X. Hong, M. Gerla, G. Pei, and C. Chiang. "A Group Mobility Model for Ad Hoc Wireless Networks". *In Proceedings of the ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems (MSWiM), August 1999*

[18] Christian Bettstetter, Hannes Hartenstein, Xavier P´erez–Costa, "Stochastic Properties of the Random Waypoint Mobility Model", *ACM/Kluwer Wireless Networks, Special Issue on Modeling & Analysis of Mobile Networks. Authors' preprint. Mar 10, 2003*

[19] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research, in Wireless Communication and Mobile Computing (WCMC)": *Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483-502, 2002.*

[20] F. Bai, A. Helmy, " A Survey of Mobility Modeling and Analysis in Wireless Adhoc Networks ", *Book Chapter in the book on "Wireless Ad Hoc and Sensor Networks", Kluwer Academic Publishers,June 2004.*

[21] P. Johansson, T. Larsson, N. Hedman, B.Mielczarek, and M. Degermark, "Scenario-Based Performance Analysis Of Routing Protocols For Mobile Ad-Hoc Networks", *in International Conference on Mobile Computing and Networking (MobiCom'99), 1999, pp. 195--206*

[22] B. Liang, Z. J. Haas," Predictive Distance-Based Mobility Management for PCS Networks", *in Proceedings of IEEE Information Communications Conference (INFOCOM 1999), Apr. 1999*

[23] C. Tracy, B. Jeff, D.Vanessa, "A Survey of Mobility Models for Ad hoc Network Research, Special Issue on Mobile Adhoc Networking: Research, Trends and Applications," *Journal of Wireless Communications and Mobile Computing, 2002, pp. 483-502.*

[24] H. Xiaoyan, G. Mario, P. Guangyu, C.A. Ching-Chuan,"Group Mobility Model for Ad Hoc Wireless Networks," *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*

[25] A.S. Bhatia, R.K. Cheema, "Analysing And Implementing The Mobility Over MANETS Using Random Waypoint Mobility Model", *International Journal of Computer applications (0975- 8887), Volume 68-No.17 ,April 2013.*

[26] Samer A.B.Awwad· Chee Kyun Ng· NorK.Noordin·Mohd. Fadlee A. Rasid, "Cluster Based Routing Protocol for Mobile Nodes in Wireless Sensor Network", *Wireless Pers Commun (2011) 61:251–281*

[27] B. Liu, P. Bras, O. Dousse, P. Nain, and D.F. Towsley, "Mobility Improves Coverage Of Sensor Networks", *in Proc. MobiHoc, 2005, pp.300-308*.

[28] Do-Seong Kim,Yeong-Jee Chung, "Self-Organization Routing Protocol Supporting Mobile Nodes for Wireless Sensor Network", *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 2006*

[29] Xinlian Zhou, Bin Xiao, "A Energy-efficient Scheduling Algorithm in MWSN", *Proceedings of the Second International Symposium on Networking and Network Security (ISNNS '10) Jinggangshan, P. R. China, 2-4, April. 2010, pp. 149-152*

[30] MIT µAMPS projectns2 code extensions.
http://www-mtl.mit.edu/research/icsystems/uamps/research/leach/leach_code.shtml

[31] Marc Greis' Tutorial for the UCB/LBNL/VINT Network Simulator "ns", *http://www.isi.edu/nsnam/ns/tutorial/*

[32] R.Alagu Pushpa, A.Vallimayil, V.R.Sarma Dhulipala, "Impact of Mobility models on Mobile Sensor Networks", *International Journal of Communication Network & Security, Volume-1, Issue-1, 2011*

[33] Eitan Altman, Tania Jimenez, "NS Simulator for Beginners", University of de Los Andes, Merida, Venezula and ESSI, Sophia Antipolis, France, Lecture notes, 2003

[34] K. Chan, F. Fekri, and H. Pishro-Niki, "Analysis Of Hierarchical Algorithms For Wireless Sensor Network Routing Protocols," *IEEE Wireless Communications and Networking Conference, vol. 3, pp. 1830 – 1835, 2005*

[35] K. Fall and K. Varadhan, "The Network Simulator NS-2 Documentation," January 2008.[Online]. Available: *http://www.isi.edu/nsnam/ns/nsdocumentation.html*

[36] Teerawat Issariyakul Ekram Hossain, "Introduction to Network Simulator-2", *DOI: 10.1007/978-0-387-71760-91, Springer Science+Business Media, LLC 2009*

[37] Dhaval K. Patel, Milin P.Patel, Ketul S. Patel, "Scalability Analysis in Wireless Sensor Network with LEACH Routing Protocol",*IEEE 2011*

[38] Sri Chusri Haryanti and Riri Fitri Sari, "Effect of RPGM and Gauss Markov Node Mobility for Resource Allocation Performance in Mobile Ad Hoc Computational Grid", *IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.6, June 2011*

[39] M. Bani Yassein, A. Al-zou'bi, Y. Khamayseh, W. Mardini, Improvement on LEACH Protocol of Wireless Sensor Network (VLEACH), *International Journal of Digital Content Technology and its Applications Volume 3, Number 2, June 2009*

[40] V. Loscrì, G. Morabito, S. Marano, A Two-Levels Hierarchy for Low-Energy Adaptive Clustering Hierarchy (TL-LEACH), *IEEE 2005*

[41] V. Jolly and S. latifi, "Comprehensive study of routing management in wireless sensor", *March 2006. [Online]. Available: www.larces.uece.br/ celestino/RSSF%20II*

[42] F L.Lewis, "Wireless Sensor Networks", mart Environments: Technologies, Protocols, and Applications ed. D.J. Cook and S.K. Das, John Wiley, Newyork,2004

[43] Constantine G. Kakoyiannis, Penelope Gika and Philip Constantinou, Compact Antennas with Mutual Coupling for Wireless Sensor Networks, May 2009, High Frequency Electronics Copyright © 2009 Summit Technical Media, LLC

APPENDICES

APPENDIX I: LEACH Implementation and Compilation in NS-2.34

Installing LEACh on NS-2.34 requires the installation of any LINUX distro on the system along with the NS-2.34 containing all the packages namely: otcl,TclCL,Tcl/Tk,nam-1 and XGraph. These packages can be directly found by installing ns-allinone-2.34.

MIT's LEACH can be installed by using following steps:

1. Download the package "ns-234-leach.tar.gz" into the directory "**/root/ns2/ns-allinone-2.34/ns-2.34**"
2. Download the bash file "leach-setup.sh" into the directory "**/root/ns2/ns-allinone-2.34/ns-2.34**".
3. If the home directory of your "NS-2.34" is other than "**/root/ns2/ns-allinone-2.34**", then you need to find "ns-allinone-2.34" and replace with "**yourpath/ns-allinone-2.34**" for the file "leach-setup.sh".
4. Move to the directory "**/root/ns2/ns-allinone-2.34/ns-2.34**" and patch the file "leach-setup.sh" by run this command :

```
[root@localhost ~]# cd /root/ns2/ns-allinone-2.34/ns-2.34
[root@localhost ~]# /root/ns2/ns-allinone-2.34/ns-2.34
                    # bash leach-setup.sh
```

5. If the home directory of your "ns-2.34" is other than "**/root/ns2/ns-allinone-2.34**", then you need to find ".../ns-allinone-2.34" and replace with "yourpath/ns-allinone-2.34" for the following two files:

   "**Makefile**":found in the directory"**/root/ns2/ns-allinone-2.34/ns- 2.34**

   "**Makefile.in**":found in the directory"**/root/ns2/ns-allinone-2.34/**ns-2.34".
6. Edit both "Makefile" & "Makefile.in" as following:

```
CC = gcc-4.3
CPP = g++-4.3
```

7. Run the following commands

```
[root@localhost ~]# /root/ns2/ns-allinone-2.34/ns-2.34
                    # ./configure
                    # make clean
                    # make
```

8. Now modify a file named **cmu-trace.h** and add these lines:

```
// start from line 165
#ifdef MIT_uAMPS

void format_rca(Packet *p, int offset);
#define ADV_CHAR 'A'
#define REQ_CHAR 'R'
#define DATA_CHAR 'D'
#endif
```

9. Modify another file named **ns-default.tcl** (line 765)by adding following lines:

```
#——————————————————
Phy/WirelessPhy set alive_ 1
Phy/WirelessPhy set Efriss_amp_ 100e-12
Phy/WirelessPhy set Etwo_ray_amp_ 0.013e-12

Phy/WirelessPhy set EXcvr_ 50e-9
Phy/WirelessPhy set sleep_ 0
Phy/WirelessPhy set ss_ 1
Phy/WirelessPhy set dist_ 0
# ——————————————————
```

10. After that we need to re-make.
11. If the previous commands passed successfully then "Leach" by running ./test:

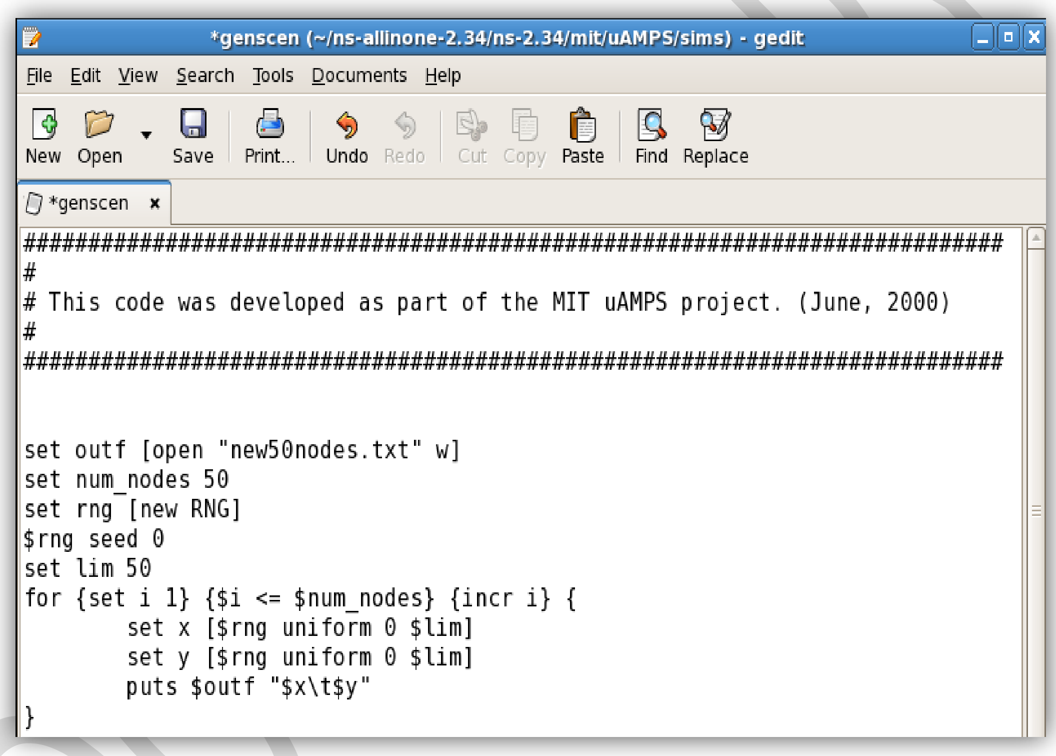
```
[root@localhost ~] # root/ns2/ns-allinone-2.34/ns-2.34
                   # ./test
```

12. Check the file "leach.err", if no error reported then go to "leach.out" and enjoy your results!

APPENDIX II: To Generate A Topological File

This file contains the information about the initial coordinates of all the nodes and is created by using "genscen" command tool as following:

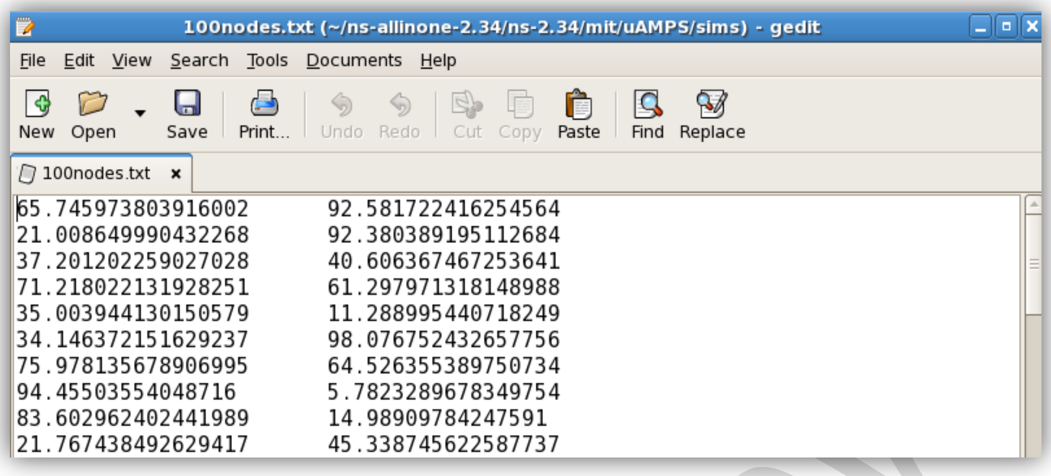
1. Change working directory to
   `[root@localhost ~] # cd ns-allinone-2.34/ns-2.34/mit/uAMPS/sim`
2. Modify the file by changing following lines according to your desired number of nodes and set output file name:
   ```
   [root@localhost  sim] # gedit genscen
   Set  outf  [ open  " output_filename.txt"  w]
   Set  num_nodes  100  to  e.g.  50
   ```




```
############################################################################
#
# This code was developed as part of the MIT uAMPS project. (June, 2000)
#
############################################################################


set outf [open "new50nodes.txt" w]
set num_nodes 50
set rng [new RNG]
$rng seed 0
set lim 50
for {set i 1} {$i <= $num_nodes} {incr i} {
        set x [$rng uniform 0 $lim]
        set y [$rng uniform 0 $lim]
        puts $outf "$x\t$y"
}
```

3. After saving the file, go to terminal and run the following command tool
   `[root@localhost genscen] # ns genscen`
4. Output file is generate as below:



5. After file is generated, call that file in **"leach_test"** file.

APPENDIX III: Mobility Scenario Generation

To implement simple RWP(Random Waypoint Mobility) model, NS-2 offers a command tool named **"setdest"** . These mobility scenario files can be implemented by setting different parameters showing below:

1. Go to terminal as follows:

```
root@localhost:~/ns-allinone-2.34/ns-2.34/indep-utils/cmu-scen-gen/setdest
File  Edit  View  Terminal  Tabs  Help
[root@localhost ns-2.34]# cd indep-utils/cmu-scen-gen/setdest
[root@localhost setdest]# ./setdest

usage:

<original 1999 CMU version (version 1)>
 ./setdest      -v <1> -n <nodes> -p <pause time> -M <max speed>
                -t <simulation time> -x <max X> -y <max Y>
```

It will generate an output file according to the desired parameters, it contains information about initial nodes' coordinates, global information of all the nodes using "General Operation director (god)" which helps to store shortest number of hopes required to reach from one node to other.

```
scen-p5-5
#
# nodes: 100, pause: 5.00, max speed: 5.00, max x: 100.00, max y: 100.00
#
$node_(0) set X_ 21.121272908809
$node_(0) set Y_ 94.991372050300
$node_(0) set Z_ 0.000000000000
$node_(1) set X_ 86.371555031576
$node_(1) set Y_ 37.412816188733
$node_(1) set Z_ 0.000000000000
$node_(2) set X_ 77.843901443186
$node_(2) set Y_ 13.827712968030
$node_(2) set Z_ 0.000000000000
$node_(3) set X_ 68.027946737086
$node_(3) set Y_ 59.649329284919
$node_(3) set Z_ 0.000000000000
```

```
$god_ set-dist 0 4 1
$god_ set-dist 0 5 1
$god_ set-dist 0 6 1
$god_ set-dist 0 7 1
$god_ set-dist 0 8 1
$god_ set-dist 0 9 1
$god_ set-dist 0 10 1
$god_ set-dist 0 11 1
$god_ set-dist 0 12 1
```

```
scen-p5-5
0.000000000000"
$ns_ at 97.468520460389 "$node_(81) setdest 92.654100729165 9.614544338506
0.000000000000"
$ns_ at 97.689480790662 "$node_(66) setdest 15.624600427672 9.514740337400
0.000000000000"
$ns_ at 98.013458147774 "$node_(47) setdest 16.130468169026 29.395900670054
0.000000000000"
```

The last part of the output file contains information about new destination (new coordinates) of nodes with their respective time.

APPENDIX IV: LEACH-MAE Implementation in NS-2.34

```
###########################################################################
#
# This code was developed as part of the MIT uAMPS project. (June, 2000)
#
###########################################################################


# Message Constants
set ADV_CH         0
set JOIN_REQ       1
set ADV_SCH        2
set DATA           3
set MAC_BROADCAST  0xffffffff
set LINK_BROADCAST 0xffffffff
set BYTES_ID       2


###########################################################################
#
#                              LEACH Application
#
###########################################################################

Class Application/LEACH -superclass Application

Application/LEACH instproc init args {

  global opt

  $self instvar rng_ isch_ hasbeench_ next_change_time_ round_
  $self instvar clusterChoices_ clusterDist_ clusterNodes_ currentCH_
  $self instvar xmitTime_ TDMAschedule_ dist_ code_
  $self instvar now_ alive_ frame_time_ end_frm_time_
  $self instvar begin_idle_ begin_sleep_
  $self instvar myADVnum_ receivedFrom_ dataReceived_

  set rng_ [new RNG]
  $rng_ seed 0
  set isch_ 0
  set hasbeench_ 0
  set next_change_time_ 0
  set round_ 0
  set clusterChoices_ ""
  set clusterDist_ ""
  set clusterNodes_ ""
  set currentCH_ ""
  set xmitTime_ ""
  set TDMAschedule_ ""
  set dist_ 0
  set code_ 0
  set now_ 0
  set alive_ 1
  set frame_time_ $opt(frame_time)
  set end_frm_time_ 0
  set begin_idle_ 0
  set begin_sleep_ 0
```

```
  set myADVnum_ 0
  set receivedFrom_ ""
  set dataReceived_ ""

  $self next $args
}

Application/LEACH instproc start {} {
  [$self mac] set node_num_ [$self nodeID]
  $self decideClusterHead
  $self checkAlive
}


############################################################################
#
#             Helper Functions
#
############################################################################

Application/LEACH instproc getRandomNumber {llim ulim} {
  $self instvar rng_
  return [$rng_ uniform $llim $ulim]
}

Application/LEACH instproc node {} {
  return [[$self agent] set node_]
}

Application/LEACH instproc nodeID {} {
  return [[$self node] id]
}

Application/LEACH instproc mac {} {
  return [[$self node] set mac_(0)]
}

Application/LEACH instproc getX {} {
  return [[$self node] set X_]
}

Application/LEACH instproc getY {} {
  return [[$self node] set Y_]
}

Application/LEACH instproc getER {} {
  set er [[$self node] getER]
  return $er
}

Application/LEACH instproc GoToSleep {} {
  global opt ns_
  $self instvar begin_idle_ begin_sleep_

  [[$self node] set netif_(0)] set sleep_ 1



  # If node has been awake, remove idle energy (e.g., the amount of energy
  # dissipated while the node is in the idle state).  Otherwise, the node
```

```
  # has been asleep and must remove sleep energy (e.g., the amount of
  # energy dissipated while the node is in the sleep state).

 if {$begin_idle_ > $begin_sleep_} {
    set idle_energy [expr $opt(Pidle) * [expr [$ns_ now] - $begin_idle_]]
    [$self getER] remove $idle_energy
  } else {
    set sleep_energy [expr $opt(Psleep) * [expr [$ns_ now] -
$begin_sleep_]]
    [$self getER] remove $sleep_energy
  }
  set begin_sleep_ [$ns_ now]
  set begin_idle_ 0
}

Application/LEACH instproc WakeUp {} {
  global opt ns_
  $self instvar begin_idle_ begin_sleep_

  [[$self node] set netif_(0)] set sleep_ 0

  # If node has been asleep, remove sleep energy (e.g.the amount of energy
  # dissipated while the node is in the sleep state).  Otherwise, the node
  # has been idling and must remove idle energy (e.g., the amount of
  # energy dissipated while the node is in the idle state).

  if {$begin_sleep_ > $begin_idle_} {
    set sleep_energy [expr $opt(Psleep) * [expr [$ns_ now] -
$begin_sleep_]]
    [$self getER] remove $sleep_energy
  } else {
    set idle_energy [expr $opt(Pidle) * [expr [$ns_ now] - $begin_idle_]]
    [$self getER] remove $idle_energy
  }
  set begin_idle_ [$ns_ now]
  set begin_sleep_ 0
}

Application/LEACH instproc setCode code {
  $self instvar code_
  set code_ $code
  [$self mac] set code_ $code
}

Application/LEACH instproc checkAlive {} {

  global ns_ chan opt node_
  $self instvar alive_ TDMAschedule_
  $self instvar begin_idle_ begin_sleep_

  # Check the alive status of the node.  If the node has run out of
  # energy, it no longer functions in the network.
  set ISalive [[[$self node] set netif_(0)] set alive_]
  if {$alive_ == 1} {
    if {$ISalive == 0} {
      puts "Node [$self nodeID] is DEAD!!!!"
      $chan removeif [[$self node] set netif_(0)]
      set alive_ 0
      set opt(nn_) [expr $opt(nn_) - 1]

      if {$opt(rcapp) == "LEACH-C/StatClustering" && \
```

```
        [$self isClusterHead?]} {
      foreach element $TDMAschedule_ {
        if {$element != [$self nodeID]} {
          puts "Node $element is effectively DEAD!!!!"
          $chan removeif [$node_($element) set netif_(0)]
          [$node_($element) set netif_(0)] set alive_ 0
          [$node_($element) set rca_app_] set alive_ 0
          set opt(nn_) [expr $opt(nn_) - 1]
        }
      }
    }
  } else {
    $ns_ at [expr [$ns_ now] + 0.1] "$self checkAlive"
    if {$begin_idle_ >= $begin_sleep_} {
      set idle_energy [expr $opt(Pidle) * [expr [$ns_ now] -
$begin_idle_]]
      [$self getER] remove $idle_energy
      set begin_idle_ [$ns_ now]
    } else {
      set sleep_energy [expr $opt(Psleep) * [expr [$ns_ now] -
$begin_sleep_]]
      [$self getER] remove $sleep_energy
      set begin_sleep_ [$ns_ now]
    }
  }
}
if {$opt(nn_) < $opt(num_clusters)} "sens_finish"
}

############################################################################
#
# Distributed Cluster Set-up Functions
#
############################################################################

Application/LEACH instproc decideClusterHead {} {

  global chan ns_ opt node_

  $self instvar next_change_time_ round_ clusterNodes_ clusterDist_
  $self instvar now_ TDMAschedule_ beginningE_ alive_
  $self instvar myADVnum_ CHheard_

  set CHheard_ 0
  [$self mac] set CHheard_ $CHheard_
  set myADVnum_ 0
  [$self mac] set myADVnum_ $myADVnum_

  # Check the alive status of the node.  If the node has run out of
  # energy, it no longer functions in the network.
  set ISalive [[[$self node] set netif_(0)] set alive_]
  if {$alive_ == 1 && $ISalive == 0} {
    puts "Node [$self nodeID] is DEAD!!!! Energy = [[$self getER] query]"
    $chan removeif [[$self node] set netif_(0)]
    set alive_ 0
    set opt(nn_) [expr $opt(nn_) - 1]
  }
  if {$alive_ == 0} {return}

  set now_ [$ns_ now]
  set nodeID [$self nodeID]
```

```
set beginningE_ [[$self getER] query]

$self setCode 0
$self WakeUp

set tot_rounds [expr int([expr $opt(nn_) / $opt(num_clusters)])]
if {$round_ >= $tot_rounds} {
  set round_ 0
}
```

Following is the contributed code of our proposed algorithm defined for Cluster Head selection.

```
  if { $round_ == 0} {
  #
  # Pi(t) = k / (N - k mod(r,N/k))
  # where k is the expected number of clusters per round
  # N is the total number of sensor nodes in the network
  # and r is the number of rounds that have already passed.
  #
  set nn $opt(nn_)
  if {[expr $nn - $opt(num_clusters) * $round_] < 1} {
    set thresh 1

  } else {
    set thresh [expr double($opt(num_clusters)) /  \
      [expr $nn - $opt(num_clusters) * $round_]]
  }
 puts "THRESH = $thresh"
 set clusterNodes_ ""
 set TDMAschedule_ ""
 if {[$self getRandomNumber 0 1] < $thresh} {
   puts "$nodeID: ********************************************"
   puts "$nodeID: Is a cluster head at time [$ns_ now]"
   $self setClusterHead
   set random_access [$self getRandomNumber 0 $opt(ra_adv)]
   $ns_ at [expr $now_ + $random_access] "$self advertiseClusterHead"

} else {
   puts "$nodeID: ********************************************"
   $self unsetClusterHead
}

} else {

   set Etotal 0

for {set num_clusters 0} {$opt(num_clusters) <= [expr $opt(num_clusters)
-1]} {incr num_clusters}  {

     set id [$num_clusters($id)]
     set $self clusterDist_
     for {set id 0} {$id < [expr $opt(nn)-1]} {incr id} {
       set $self clusterNodes_
```

```
        set E [[$app getER] query]
        set E [[$self getER] query]
        set cn_energy [list $E $node_($id)]
        set app [$node_($id) set rca_app_]
        while {$node_($id) <= $clusterNodes_} {
        #incr $self clustersNodes_
        set Etotal [expr $Etotal + $E]
      }
      }

      set MeanEnergy [expr [$Etotal / $self clusterNodes_]]
        puts "MeanEnergy = $MeanEnergy"
       lsearch $cn_energy max($E)
       if {$E > $MeanEnergy } {
        set app [$node_($id) set rca_app_]
        puts "ClusterHead = $node_($id) with max energy $E"
        set clusterNodes_ ""
        set TDMAschedule_ ""
       puts "$nodeID: *******************************************"
       puts "$nodeID: Is a cluster head at time [$ns_ now]"
       $self setClusterHead
       set random_access [$self getRandomNumber 0 $opt(ra_adv)]
       $ns_ at [expr $now_ + $random_access] "$self advertiseClusterHead"
       }  else {
    puts "$nodeID: *******************************************"
    $self unsetClusterHead
       }
        }
     }

  incr round_
  set next_change_time_ [expr $now_ + $opt(ch_change)]
  $ns_ at $next_change_time_ "$self decideClusterHead"
  $ns_ at [expr $now_ + $opt(ra_adv_total)] "$self findBestCluster"
}


Application/LEACH instproc advertiseClusterHead {} {

  global ns_ opt ADV_CH MAC_BROADCAST LINK_BROADCAST BYTES_ID
  $self instvar currentCH_ code_

  set chID [$self nodeID]
  set currentCH_ $chID
  pp "Cluster Head $currentCH_ broadcasting ADV at time [$ns_ now]"
  set mac_dst $MAC_BROADCAST
  set link_dst $LINK_BROADCAST
  set msg [list $currentCH_]
  set datasize [expr $BYTES_ID * [llength $msg]]

  # Send beacons opt(max_dist) meters so all nodes can hear.
  $self send $mac_dst $link_dst $ADV_CH $msg $datasize $opt(max_dist)
$code_
}

Application/LEACH instproc findBestCluster {} {

  global ns_ opt

  $self instvar now_ dist_ myADVnum_
  $self instvar clusterChoices_ clusterDist_ currentCH_
```

```
  set nodeID [$self nodeID]
  set min_dist 10000
  if [$self isClusterHead?] {
    # If node is CH, determine code and create a TDMA schedule.
    set dist_ $opt(max_dist)
    set currentCH_ $nodeID
    set myADVnum_ [[$self mac] set myADVnum_]
    # There are opt(spreading) - 1 codes available b/c need 1 code
    # for communication with the base station.
    set numCodesAvail [expr 2 * $opt(spreading) - 1]
    set ClusterCode [expr int(fmod($myADVnum_, $numCodesAvail)) + 1]
    $ns_ at [expr $now_ + $opt(ra_adv_total) + $opt(ra_join)] \
        "$self createSchedule"
  } else {
    # If node is not a CH, find the CH which allows minimum transmit
    # power for communication.  Set the code and "distance" parameters
    # accordingly.
    if {$clusterChoices_ == ""} {
      puts "$nodeID: Warning!!! No Cluster Head ADVs were heard!"
      set currentCH_ $opt(nn)
      $self SendMyDataToBS
      return
    }
    foreach element $clusterChoices_ {
      set chID [lindex $element 0]
      set clustID [lindex $element 2]
      set ind [lsearch $clusterChoices_ $element]
      set d [lindex $clusterDist_ $ind]
      if {$d < $min_dist} {
          set min_dist $d
          set currentCH_ $chID
          set numCodesAvail [expr 2 * $opt(spreading) - 1]
          set ClusterCode [expr int(fmod($ind, $numCodesAvail)) + 1]
      }
    }
    set dist_ $min_dist

    set random_access [$self getRandomNumber 0 \
                            [expr $opt(ra_join) - $opt(ra_delay)]]
    $ns_ at [expr $now_ + $opt(ra_adv_total) + $random_access] \
          "$self informClusterHead"
    $self GoToSleep
  }

  $self setCode $ClusterCode
  puts "$nodeID: Current cluster-head is $currentCH_, code is $ClusterCode,
\
        dist is $dist_"

  set clusterChoices_ ""
  set clusterDist_ ""
}

Application/LEACH instproc informClusterHead {} {

  global ns_ opt JOIN_REQ MAC_BROADCAST BYTES_ID
  $self instvar currentCH_ dist_ code_

  set nodeID [$self nodeID]
  set chID $currentCH_
```

```
  pp "$nodeID: sending Join-REQ to $chID (dist = $dist_) at time [$ns_
now]"
  set mac_dst $MAC_BROADCAST
  set link_dst $chID
  set msg [list $nodeID]
  set spreading_factor $opt(spreading)
  set datasize [expr $spreading_factor * $BYTES_ID * [llength $msg]]
  $self WakeUp

  # NOTE!!!! Join-Req message sent with enough power so all nodes in
  # the network can hear the message.  This avoids the hidden terminal
  # problem.
  $self send $mac_dst $link_dst $JOIN_REQ $msg $datasize $opt(max_dist)
$code_
}

Application/LEACH instproc createSchedule {} {

  global ns_ opt ADV_SCH MAC_BROADCAST BYTES_ID

  $self instvar clusterNodes_ TDMAschedule_
  $self instvar dist_ code_ now_ beginningE_

  set numNodes [llength $clusterNodes_]
  set chID [$self nodeID]
  if {$numNodes == 0} {
    set xmitOrder ""
    puts "Warning!  There are no nodes in this cluster ($chID)!"
    $self SendMyDataToBS
  } else {
    # Set the TDMA schedule and send it to all nodes in the cluster.
    set xmitOrder $clusterNodes_
    set msg [list $xmitOrder]
    set spreading_factor $opt(spreading)
    set datasize [expr $spreading_factor * $BYTES_ID * [llength
$xmitOrder]]
    pp "$chID sending TDMA schedule: $xmitOrder at time [$ns_ now]"
    pp "Packet size is $datasize."
    set mac_dst $MAC_BROADCAST
    set link_dst $chID
    $self send $mac_dst $link_dst $ADV_SCH $msg $datasize $dist_ $code_
  }

  set TDMAschedule_ $xmitOrder
  set outf [open $opt(dirname)/TDMAschedule.$now_.txt a]
  puts $outf "$chID\t$TDMAschedule_"
  close $outf

  set outf [open $opt(dirname)/startup.energy a]
  puts $outf "[$ns_ now]\t$chID\t[expr $beginningE_ - [[$self getER]
query]] "
  close $outf

}


############################################################################
#
# Receiving Functions
#
############################################################################
```

```
Application/LEACH instproc recv {args} {

  global ADV_CH JOIN_REQ ADV_SCH DATA ns_

  $self instvar currentCH_

  set msg_type [[$self agent] set packetMsg_]
  set chID [lindex $args 0]
  set sender [lindex $args 1]
  set data_size [lindex $args 2]
  set msg [lrange $args 3 end]

  set nodeID [$self nodeID]

  if {$msg_type == $ADV_CH && ![$self isClusterHead?]} {
    $self recvADV_CH $msg
  } elseif {$msg_type == $JOIN_REQ && $nodeID == $chID} {
    $self recvJOIN_REQ $msg
  } elseif {$msg_type == $ADV_SCH  && $chID == $currentCH_} {
    $self recvADV_SCH $msg
  } elseif {$msg_type == $DATA && $nodeID == $chID} {
    $self recvDATA $msg
  }

}

Application/LEACH instproc recvADV_CH {msg} {

  global ns_
  $self instvar clusterChoices_ clusterDist_
  set chID [lindex $msg 0]
  set nodeID [$self nodeID]
  pp "$nodeID rcvd ADV_CH from $chID at [$ns_ now]"
  set clusterChoices_ [lappend clusterChoices_ $msg]
  set clusterDist_ [lappend clusterDist_ [[$self agent] set distEst_]]
}

Application/LEACH instproc recvJOIN_REQ {nodeID} {

  global ns_
  $self instvar clusterNodes_
  set chID [$self nodeID]
  pp "$chID received notice of node $nodeID at time [$ns_ now]"
  set clusterNodes_ [lappend clusterNodes_ $nodeID]
}

Application/LEACH instproc recvADV_SCH {order} {

  global ns_ opt
  $self instvar xmitTime_ next_change_time_ now_
  $self instvar beginningE_ frame_time_ end_frm_time_

  set nodeID [$self nodeID]
  set ind [lsearch [join $order] $nodeID]

  set outf [open $opt(dirname)/startup.energy a]
  puts $outf "[$ns_ now]\t$nodeID\t[expr $beginningE_ - [[$self getER]
query]]"
  close $outf
```

```
  if {$ind < 0} {
    puts "Warning!!!!  $nodeID does not have a transmit time!"
    puts "Must send data directly to BS."
    set outf [open $opt(dirname)/TDMAschedule.$now_.txt a]
    puts -nonewline $outf "$nodeID\t"
    close $outf
    $self SendMyDataToBS
    return
  }
  # Determine time for a single TDMA frame.  Each node sends data once
  # per frame in the specified slot.
  set frame_time_ [expr [expr 5 + [llength [join $order]]] *
$opt(ss_slot_time)]
  set xmitTime_ [expr $opt(ss_slot_time) * $ind]
  set end_frm_time_ [expr $frame_time_ - $xmitTime_]
  set xmitat [expr [$ns_ now] + $xmitTime_]
  pp "$nodeID scheduled to transmit at $xmitat.  It is now [$ns_ now]."
  if {[expr $xmitat + $end_frm_time_] < \
      [expr $next_change_time_ - 10 * $opt(ss_slot_time)]} {
    $ns_ at $xmitat "$self sendData"
  }

  $self GoToSleep
}

 Application/LEACH instproc recvDATA {msg} {

  global ns_ opt
  $self instvar TDMAschedule_ receivedFrom_ dataReceived_

  set chID [$self nodeID]
  set nodeID [lindex $msg 0]
  pp "CH $chID received data ($msg) from $nodeID at [$ns_ now]"
  set receivedFrom_ [lappend receivedFrom_ $nodeID]

  set last_node [expr [llength $TDMAschedule_] - 1]
  if {$chID == [lindex $TDMAschedule_ $last_node]} {
    set last_node [expr $last_node - 1]
  }
  if {$nodeID == [lindex $TDMAschedule_ $last_node]} {
    # After an entire frame of data has been received, the cluster-head
    # must perform data aggregation functions and transmit the aggregate
    # signal to the base station.
    pp "CH $chID must now perform comp and xmit to BS."
    set num_sigs [llength $TDMAschedule_]
    set compute_energy [bf $opt(sig_size) $num_sigs]
    pp "\tcompute_energy = $compute_energy"
    [$self getER] remove $compute_energy
    set receivedFrom_ [lappend receivedFrom_ $chID]
    set dataReceived_ $receivedFrom_
    set receivedFrom_ ""

    $self SendDataToBS
  }
}


############################################################################
#
# Sending Functions
#
```

```
###########################################################################

Application/LEACH instproc sendData {} {

  global ns_ opt DATA MAC_BROADCAST BYTES_ID

  $self instvar next_change_time_ frame_time_ end_frm_time_
  $self instvar currentCH_ dist_ code_ alive_

  set nodeID [$self nodeID]
  set msg [list [list $nodeID , [$ns_ now]]]
  # Use DS-SS to send data messages to avoid inter-cluster interference.
  set spreading_factor $opt(spreading)
  set datasize [expr $spreading_factor * \
              [expr [expr $BYTES_ID * [llength $msg]] + $opt(sig_size)]]

  $self WakeUp

  pp "$nodeID sending data $msg to $currentCH_ at [$ns_ now] (dist =
$dist_)"
  set mac_dst $MAC_BROADCAST
  set link_dst $currentCH_
  $self send $mac_dst $link_dst $DATA $msg $datasize $dist_ $code_

  # Must transmit data again during slot in next TDMA frame.
  set xmitat [expr [$ns_ now] + $frame_time_]
  if {$alive_ && [expr $xmitat + $end_frm_time_] < \
                [expr $next_change_time_ - 10 * $opt(ss_slot_time)]} {
    $ns_ at $xmitat "$self sendData"
  }
  set sense_energy [expr $opt(Esense) * $opt(sig_size) * 8]
  pp "Node $nodeID removing sensing energy = $sense_energy J."
  [$self getER] remove $sense_energy

  if {$currentCH_ != $nodeID} {
    $self GoToSleep
  }
}

Application/LEACH instproc send {mac_dst link_dst type msg
                                    data_size dist code} {
  global ns_
  $self instvar rng_

  #set random_delay [expr 0.005 + [$rng_ uniform 0 0.005]]
  #$ns_ at [expr [$ns_ now] + $random_delay] "$self send_now $mac_dst \
  #  $link_dst $type $msg $data_size $dist"
  $ns_ at [$ns_ now]  "$self send_now $mac_dst \
      $link_dst $type $msg $data_size $dist $code"
}

Application/LEACH instproc send_now {mac_dst link_dst type msg \
                                        data_size dist code} {
    [$self agent] set packetMsg_ $type
    [$self agent] set dst_addr_ $mac_dst
    [$self agent] sendmsg $data_size $msg $mac_dst $link_dst $dist $code
}

Application/LEACH instproc SendDataToBS {} {

 global ns_ opt bs MAC_BROADCAST DATA BYTES_ID
```

```
	$self instvar code_ rng_ now_

	# Data must be sent directly to the basestation.
	set nodeID [$self nodeID]
	set msg [list [list [list $nodeID , [$ns_ now]]]]
	# Use DS-SS to send data messages to avoid inter-cluster
interference.
	set spreading_factor $opt(spreading)
	set datasize [expr $spreading_factor * \
	                 [expr $BYTES_ID * [llength $msg] +
$opt(sig_size)]]
	set dist [nodeToBSDist [$self node] $bs]

	set mac_dst $MAC_BROADCAST
	set link_dst $opt(bsID)
	set random_delay [expr [$ns_ now] + [$rng_ uniform 0 0.01]]
	pp "Node $nodeID sending $msg to BS at time $random_delay"
	$ns_ at $random_delay "$self send $mac_dst $link_dst $DATA \
	                     $msg $datasize $dist $opt(bsCode)"
}

Application/LEACH instproc SendMyDataToBS {} {
	global ns_ opt
	$self instvar next_change_time_ alive_
	puts "Data being sent to the Base Station"
	$self SendDataToBS
	puts "Data was sent to the base station"
	set xmitat [expr [$ns_ now] + $opt(frame_time)]
	if {$alive_ && [expr $xmitat + $opt(frame_time)] < \
	          [expr $next_change_time_ - $opt(frame_time)]} {
	  $ns_ at $xmitat "$self SendMyDataToBS"
	}
}
```