M. Sc. Engineering Thesis

A NOVEL APPROACH TO CLUSTER HETEROGENEOUS SENSOR NETWORK (CHSN)

by

A. B. M. Alim Al Islam

Submitted to
the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology (BUET)
Dhaka 1000

May 30, 2009

The thesis titled **"A NOVEL APPROACH TO CLUSTER HETEROGENEOUS SENSOR NETWORK (CHSN)"**, submitted by A. B. M. Alim Al Islam, Roll No. 100605047P, Session October, 2006, to the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

1. ____________ Dr. Md. Humayun Kabir Associate Professor Department of CSE BUET, Dhaka 1000	Chairman (Supervisor)
2. ____________ Dr. Md. Saidur Rahman Professor & Head Department of CSE BUET, Dhaka 1000	Member (Ex-Officio)
3. ____________ Dr. Mahmuda Naznin Assistant Professor Department of CSE BUET, Dhaka 1000	Member
4. ARahman 30.05.09 Dr. A. K. M. Ashikur Rahman Assistant Professor Department of CSE BUET, Dhaka 1000	Member
5. ____________ Dr. Hafiz Md. Hasan Babu Professor Department of CSE Dhaka University, Dhaka	Member (External)

Candidate's Declaration

This is to certify that the research work entitled "A Novel Approach to Cluster Heterogeneous Sensor Network (CHSN)" is the outcome of the investigations carried out by me under the supervision of Dr. Md. Humayun Kabir in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka-1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

A. B. M. Alim Al Islam
Candidate

Acknowledgments

All praise is to almighty ALLAH. I thank ALLAH (SWT) for providing me the knowledge, capability and perseverance to persist and successfully complete this thesis. I express my utmost gratefulness to my supervisor Dr. Md. Humayun Kabir, Associate Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology for his encouragement, witty guidance and priceless suggestions. I also want to show my gratitude to Dr. Mahmuda Naznin and Dr. A. K. M. Ashikur Rahman for their serenity and valuable suggestions. It has been a great opportunity to work with them. Their vast experience and in-depth knowledge, scholarly guidance with precious advice and amusing support have been a great impetus to the accomplishment of my thesis work. I also want to thank the other members of my thesis committee: Dr. Md. Saidur Rahman and Dr. Hafiz Md. Hasan Babu for their valuable suggestions. I would like to express my ever gratefulness to my parents and family members for their continuous support. Finally, I remember some names (Sayeed, Adnan, Sumon, Atif, Rifat, Faiz, and Kuntal) for their supports. May ALLAH bless them all in here and hereafter.

Abstract

Network lifetime is one of the important metrics in performance evaluation of sensor network. It depends on both the rate of energy consumption and the relative distribution of the energy consumption among the sensor nodes. Among various clustering solutions to elongate the network lifetime, LEACH (Low-Energy Adaptive Clustering Hierarchy) is one of the most widely cited solutions due to its simplicity and effectiveness. However, LEACH considers only homogeneous sensor network. Moreover, there is no known complete mathematical model derived for LEACH that can be used to tune various LEACH parameters in order to achieve better performance. In this thesis, we first formulate a complete mathematical model for LEACH and verify its correctness through simulation. Next, we present three heuristics to enhance the energy efficiency of LEACH and propose an energy efficient modification of LEACH, CHSN (Cluster Heterogeneous Sensor Networks), considering the heterogeneity of sensor nodes in terms of residual energy. Our simulation results show that CHSN improves the network lifetime significantly. The increase in network lifetime has been shown in terms of the First Node Dies (FND), the Half of the Nodes Die (HND) and the Last Node Dies (LND).

Table of Contents

List of Figures

List of Tables

1 Introduction

With the advent of new technology and low production costs, wireless sensor networks (WSN) have been proved to be useful in myriad of diversified applications although its original development was motivated by military applications, such as battlefield surveillance, enemy tracking and monitoring. Most of the WSN applications involve monitoring, tracking, or controlling, e.g., habitat monitoring, object tracking, nuclear reactor control, fire detection, and traffic monitoring etc.

In a typical WSN application, sensor nodes are scattered in a region from where they collect data to achieve certain goals. Data collection may be continuous, periodic or event based. Irrespective of data collection type, different kinds of management, such as power management, dynamic topology (due to node failure) management, self-configuration management, resource management, and security management are necessary for WSN. Power management deals with the optimum energy usage in order to increase the network lifetime. Dynamic topology management dynamically adjusts the topology in case of the death of an existing node or the arrival of a new node. Self configuration management enables the nodes to tune its parameters on the fly. Resource management takes the role to ensure effective resource (CPU and memory) sharing among multiple tasks. Security management guarantees protection against any theft or intrusion in the network.

Among all of these, power management is very important since the sensor nodes come with pre-installed limited powered battery. Moreover, the batteries cannot be replaced in the sensor nodes once they are in operation. For these reasons, the algorithms and protocols used in WSN have to be energy efficient in order to have better power management. Different techniques are used to achieve energy efficiency like clustering, data compression, dynamic power management etc.

Clustering is a technique in which some nodes act as the cluster heads and the others act as the followers. The followers collect data and send it to their corresponding cluster heads. The cluster heads aggregate its own data with the data received from its followers. Aggregated data is then sent to a sink to accomplish a specific goal. Cluster heads remain closer to their follower sensor nodes compared to the sink. It takes less energy to transmit data to the cluster head instead of the sink, which allows the sensor nodes to conserve more energy and live longer in WSN.

1.1 Motivation

There are different clustering techniques already established for ad-hoc networks. However, those techniques cannot be directly used in WSN because of the fact that WSN imposes strict requirements on the energy efficiency than that ad-hoc networks do. As a result, many techniques have been proposed for clustering in WSN. LEACH [1] is one of the simple and popular clustering techniques used for WSN. However, LEACH does not consider the heterogeneity of the sensor nodes in terms of residual energy when it selects the cluster heads. LEACH has some tunable parameters that can be tuned to achieve optimal energy consumption. Due to the absence of a mathematical model, it is also hard to tune these parameters.

In this thesis, we have proposed a mathematical model for LEACH and proved its correctness by simulation results. We have also tuned LEACH parameters using our proposed mathematical model in order to achieve optimal energy consumption goal. Finally, we have proposed a new clustering approach namely Cluster Heterogeneous Sensor Network (CHSN) and we give the mathematical model for CHSN. Simulation results prove that CHSN performs better than LEACH [1] and its variations [16], [17].

1.2 Disposition

This thesis consists of seven chapters. In the second chapter, the general concepts of sensor node, wireless sensor network and clustering are briefly discussed. The third chapter describes some existing popular clustering techniques in sensor networks. In the fourth chapter, the formulation of a proposed mathematical model for LEACH is elaborated. In the fifth chapter, a novel approach to Cluster Heterogeneous Sensor Network (CHSN) is proposed. The sixth chapter presents the simulation results to verify the correctness of the proposed LEACH mathematical model and the performance of our

CHSN. Finally, we conclude the thesis in Chapter seven with shedding some light on the future works.

2 General Concepts

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices. Using sensors these devices cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants at different locations. These autonomous devices are called sensor nodes. The main challenge for WSN is the power management due to the one-time low power batteries installed in the sensor nodes. To meet power management challenge, different techniques are used. Clustering is one of the most prominent one among them. In this chapter, we briefly describe sensor node, WSN, and clustering concept.

2.1 Sensor Node

A sensor node is an electronic device which is capable of gathering sensory information, processing the information, and communicating the information with similar type of other devices. Figure 2-1 shows a widely known sensor node Berkeley Mote. There are different commercial sensor nodes like Mica, IMote, Kmote, Dot etc.

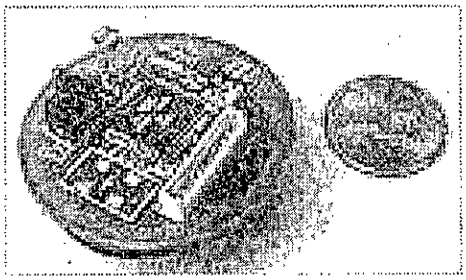


Figure 2-1 Berkeley Mote

2.1.1 Architecture

The main components of a sensor node are microcontroller, transceiver, external memory, power source and one or more sensors are shown in Figure 2-2. Microcontroller processes data and controls the functionality of the other components in the sensor node.

Transceiver transmits and receives data. Memory contains programs and all sort of data. Data can be application related or used to identify the device if necessary. Power sources supplies power required for data processing and communication. Sensors are hardware components that produce measurable response to a change in a physical condition like temperature and pressure. Sensors sense or measure physical data of the area to be monitored.

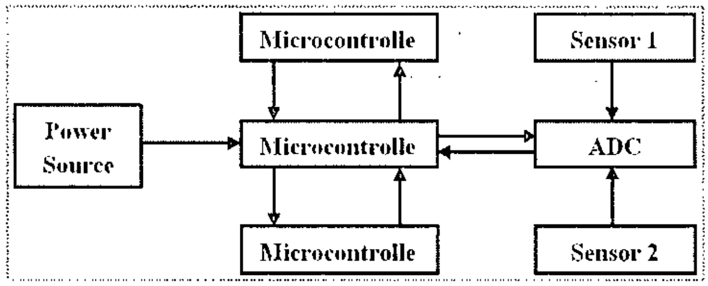


Figure 2-2 Basic Architecture of a typical sensor node

Most of the sensor nodes are designed to be low cost and small so that their deployment can be arbitrary and in a big number in a WSN. For this reason, they come with following limitations:

1. Limited power
2. Limited memory and processing capabilities

2.2 Wireless Sensor Network (WSN)

WSN consists of a set of sensor nodes capable of sensing their surroundings, i.e. gathering, processing, transmitting, and relaying data in order to monitor a specific phenomenon. The sensors in a WSN may be of the same or different capabilities or characteristics. The first one is called homogeneous WSN and the later one is called heterogeneous WSN. The heterogeneity may arise from different ways, e.g., different energy levels, different transmission ranges, different application logics etc. In a heterogeneous WSN, there may be a small set of costly, but more powerful sensor nodes, called relay nodes. The main purpose to deploy relay nodes is to prolong network lifetime while preserving network connectivity. The relay nodes are capable of receiving and aggregating data packets from neighboring sensor nodes and transmitting them to the sink node directly or via multi-hop wireless paths. Figure 2-3 shows typical homogeneous and heterogeneous WSN.

Sensors are deployed in a region using any of the following three approaches –

1. Random deployment
2. Regular deployment
3. Dynamic deployment

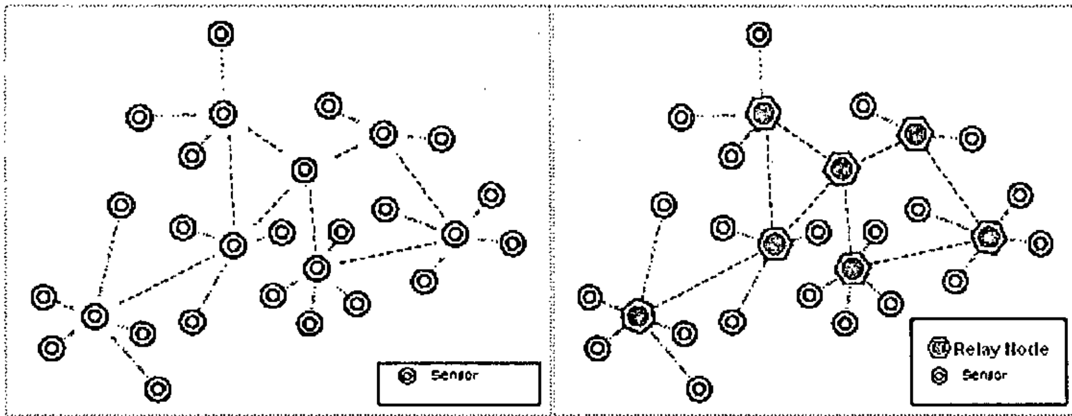


Figure 2-3 (a) Homogeneous Wireless Sensor Network (b) Heterogeneous Wireless Sensor Network

In the random deployment sensor nodes can be dropped from an aircraft. Regular deployments are well planned and the sensor nodes are deployed in the fixed locations. If the deployed sensor nodes are allowed to move then it becomes dynamic. Deployed sensors can communicate to the real world via Internet gateways. Figure 2-4 shows this type of operation.

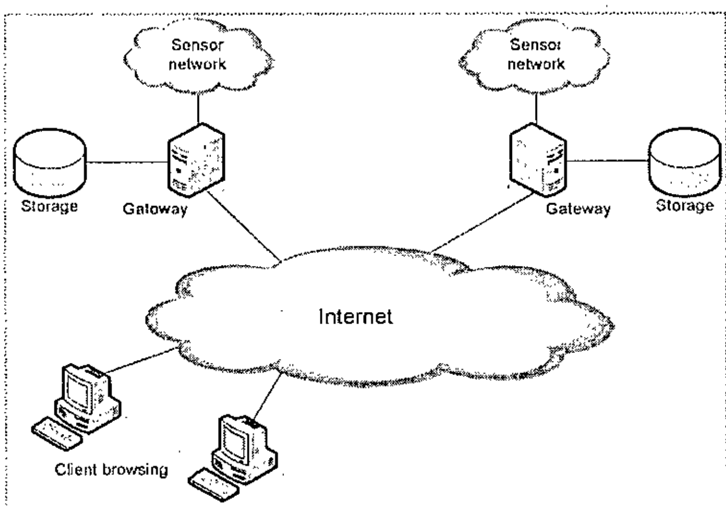


Figure 2-4 Communication of WSN into the real world

2.2.1 Challenges

After deploying the sensor nodes in a WSN it is necessary to ensure that the network is functioning effectively. There are many issues to consider in order to ensuring effective operation of a WSN. We are briefly discussing those issues in this section.

Energy efficiency is a major issue for WSN operation and management. In most of the cases, the size of the sensor nodes is very small. Hence, the batteries installed in the sensor nodes to supply the power are also very small and have limited power supply capability from the very beginning. Moreover, sensors are usually deployed in the areas which are not very easy to access. i.e., energy replenishment of sensor nodes is not possible. Network may contain a huge number of sensor nodes. Also, the deployment of the sensor nodes may be dense or sparse or combination of both. Variation in the network size and density imposes several difficulties to devise common algorithms for WSN. Transmission range and the sensing range impose another kind of challenge in the deployment and management of WSN. Connectivity and coverage are two other issues need to be considered in conjunction with transmission and sensing ranges. Different types of communication techniques such as broadcast and multicast may also be needed to incorporate. Due to the small size of the sensor nodes, the hardware installed in it may not have high capability. Specially, the processing power and the memory are limited for the nodes. This hardware limitation imposes lot of difficulty to develop good operational algorithm for WSN.

There are many types of varying conditions or network dynamics that may arise in the environment or in the network respectively. This can impose following challenges:

- Sensor nodes are prone to failure. Connectivity and coverage must be maintained in the similar fashion after the death of a sensor node.
- The environment in which the sensor nodes are deployed may be changed due to different natural phenomenon like storm, rainfall etc. The nodes must be dynamically adapted to these changes in the environment.
- The topology of sensor networks may be changed very frequently due to the displacement or death of existing sensor. Also, new sensor nodes may be deployed. These changes in the topology must be dynamically maintained.

It is not feasible to manually configure thousands of sensors. Hence, the nodes must be capable to be configured themselves on the fly. Also, the sensors may need to change location and move to a foreign network. These changes are also required to be dynamically configured.

2.2.2 Applications

Now-a-days, WSNs have many applications although the original purpose of developing WSN was military sensing. In a military sensing application, different security issues are monitored by the sensor nodes. Sensor nodes are very useful in the movement tracking in the battle field. Multiple targets can be tracked using sensor nodes. Perimeter protection can also be achieved using boarder tracking with the help of these sensor nodes. There are many WSN deployments for different types of environmental monitoring. These include but not limited to habitat, temperature, pressure, and humidity monitoring. WSN is also used for wildlife conservation.

WSN has a number of applications in the industrial sensing and diagnostics. In the hazardous and risky environment in an industry WSN play a significant role. Different types of applications in the industry include:

- Manufacturing automation
- Chemical products tracking
- Disaster prevention and recovery

WSN has an important role in infrastructure protection. There are many applications to protect the infrastructure using WSN. Some of them are:

- Traffic management and control
- Roads/vehicle safety
- Electricity distribution in power grids
- Water distribution

WSN is now frequently used for different context-aware computing such as remote monitoring of a building to ensure its security. It is also used for intelligent home applications. Baby-sitting and children monitoring are some other applications of WSN.

Healthcare is another field of application for WSN. Different types of biosensors are being used for life signs monitoring, remote tracking of patients, and in-home elderly care. WSN has also important commercial applications like inventory control, product quality control, smart office spaces, environmental control in office buildings etc.

2.3 Clustering

Clustering technique subdivides a WSN into multiple parts. In each part there will be only one cluster head and the other nodes will become the followers of the head. A follower can only communicate to its head in the cluster. However, a head can communicate to any of its follower or to any other cluster head or to the base station. A cluster head can aggregate data before transmitting it to the base station directly or through other cluster heads. Figure 2-5 shows a typical example of forwarding data in a clustered WSN.

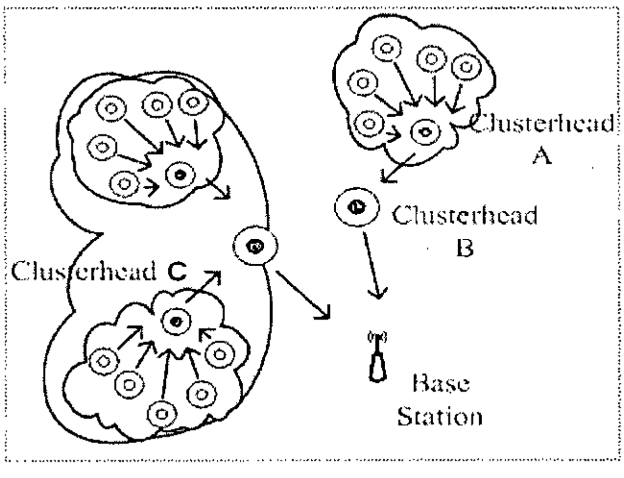


Figure 2-5 Data forwarding in a clustered WSN

2.3.1 Objectives

Clustering is done in a sensor network with following objectives:

- To improve network lifetime through reducing the energy consumption rate by decreasing the distances to which data are to be transmitted.
- To limit the required number of bits in data to be transmitted.
- To reduce network traffic and the contention for the channel.
- To aggregate and update data in cluster heads.
- To facilitate the proportionate usability of the resources by choosing cluster heads from the sensor nodes with higher capability.

- To design efficient upper layer functionalities like broadcast.
- To enable inter cluster routing by forming a virtual backbone with cluster heads and gateway nodes.
- To make the network more stable.

Since clustering promises many benefits towards WSN, many researchers have devoted their effort to build good clustering algorithm. We discuss some of prominent clustering algorithms in the next chapter.

3 Related works

Clustering a WSN yields many benefits, which have been discussed in the previous chapter. Several clustering techniques have also been proposed for partitioning nodes in wireless ad-hoc networks, mobile ad-hoc networks and sensor networks. Some of the early but not widely accepted clustering techniques are - Hierarchical Clustering [2], Distributed Clustering Algorithm (DCA) [3], Spanning Tree (or BFS Tree) based Clustering [4], Clustering With On-Demand Routing [5], Clustering based on Degree or Lowest Identifier Heuristics [6], and Distributed and Energy-Efficient Clustering [7], Adaptive Power-Aware Clustering [8]. Some of the recently developed clustering techniques are PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [9], Energy Efficient Clustering Routing [10], PEACH (Power Efficient And Adaptive Clustering Hierarchy) [11], Optimal Energy Aware Clustering [12], ACE (Algorithm For Cluster Establishment) [13], HEED (Hybrid Energy-Efficient Distributed Clustering) [14], PADCP (Power Aware Dynamic Clustering Protocol) [15], LEACH (Low-Energy Adaptive Clustering Hierarchy) [1], SEP (Stable Election Protocol) [16], and LEACH with Deterministic Cluster Head Selection [17]. We are briefly introducing the recently developed clustering techniques in this chapter.

In [9] PEGASIS introduces a near optimal chain-based protocol. Here, each node communicates only with a close neighbor and takes turns transmitting to the base station, thus reducing the amount of energy spent per round. It assumes that all nodes have global knowledge of the network and employ the greedy algorithm. It maps the problem of having close neighbors for all nodes to the traveling salesman problem. The greedy approach to constructing the chain of nodes for communication is done before the starting of the first round. The construction of the chain starts from the furthest node from the BS. This node is chosen in order to make sure that nodes farther from the BS have close neighbors, as in the greedy algorithm the neighbor distances will increase gradually since nodes already on the chain cannot be revisited. When a node dies, the chain is reconstructed in the same manner to bypass the dead node. PEGASIS is a greedy chain protocol that is near optimal for a data-gathering problem in sensor networks. It limits the number of transmissions and receptions within the chain, and uses only one transmission to the BS per round. Greedy approach considers the physical distance only, ignoring the capability of a prospective node on the chain. Hence, a node with a shorter distance but less residual energy may be chosen in the chain and may die quickly.

In [10] a routing algorithm is proposed which combines hierarchical routing and geographical routing. The process of packet forwarding from the source nodes in the target region to the base station consists of two phases—inter-cluster routing and intra-cluster routing. For inter-cluster routing, a greedy algorithm is adopted to forward packets from the cluster heads of the target regions to the base station. While picking a next hop, a cluster head compares the costs of its neighbor cluster heads to reach the destination. The cluster head with the lowest cost to the destination is chosen as the next-hop node. For intra-cluster routing, a simple flooding is used to flood the packet inside the cluster when the number of intra-cluster nodes is less than a predetermined threshold. Otherwise, the recursive geographic forwarding approach is used to disseminate the packet inside target cluster, that is, the cluster head divides the target cluster into some sub-regions, creates the same number of new copies of the query packet, and then disseminates these copies to a central node in each sub region. It repeats this recursive splitting and forwarding procedure until the number of nodes in a sub-region reach the threshold. This approach let the sensor node to conserve energy by not transmitting data directly to the base station. Like [9], it uses greedy algorithm based on the distance only but not on the capability or the residual energy. Although it deals with the optimal forwarding approach the criteria to choose the cluster heads optimally is not clearly explained.

PEACH [11] is a cluster formation technique based on overheard information from the sensor nodes. In PEACH, a node set namely $NodeSet(N_i, N_j)$ and a cluster set namely $ClusterSet(N_i, N_j)$ have been defined when a node N_i transmits packet to a node N_j. $NodeSet(N_i, N_j)$ is the set of all nodes in a circle (or sphere in a 3-dimensional space), where the center of the circle is the sender node N_i. The radius of the circle is the distance between the node N_i and the node N_j. Similarly another node set $NodeSet(Sink, N_j)$ has been define keeping the sink node at the center of a circle having a radius equal to the distance between the sink and node N_j. $ClusterSet(N_i, N_j)$ is set of all nodes that are included in $NodeSet(N_i, N_j)$ but not in $NodeSet(Sink, N_j)$. $ClusterSet(N_i, N_j)$ is the cluster set of the overheard nodes where node N_j is the cluster head. Figure 3-1 shows how PEACH forms clusters on wireless sensor networks. In this example, the *NodeSet(A, B)* and the *ClusterSet(A ,B)* are {*A, B, C, D, E, F, G*} and {*A, C, D, E*}, respectively. The node *B* becomes a cluster head of the *ClusterSet(A, B)*.

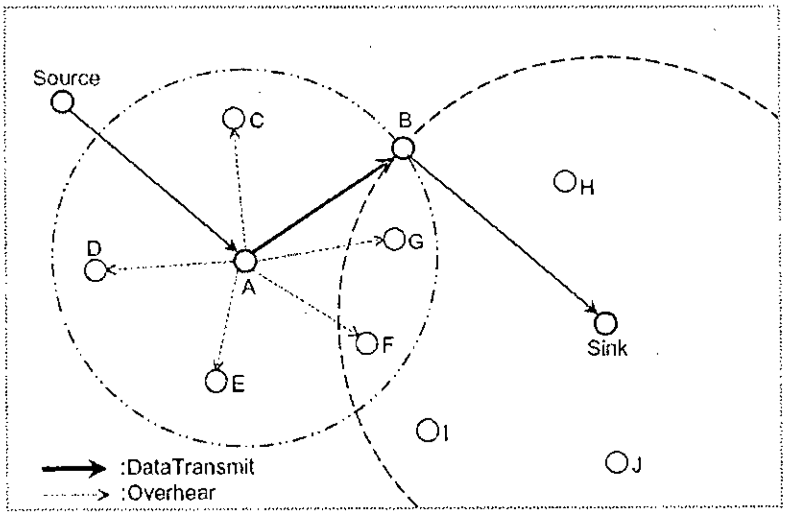


Figure 3-1 Overhearing and cluster formation in PEACH

A cluster head node N_j first sets the sink node as its next hop. Then it sets a timer to receive and aggregate multiple packets from the nodes in the cluster set for a pre-specified time T_{delay}. If node N_j overhears a packet destined to a node N_{dest}, it checks whether the distance between N_j and N_{dest} is shorter than that of between N_j and already selected next hop node. If the distance is shorter, the N_j joins to the cluster of N_{dest} and the next hop of the N_j is changed to node N_{dest}. PEACH is an adaptive clustering approach for multi-hop inter-cluster communication. However, it suffers from almost the same limitations of PEGASIS.

Optimal energy aware clustering [12] solves the balanced *k*-clustering problem optimally, where *k* signifies the number of master nodes that can be in the network. The balanced *k*-clustering problem tries to group the sensor nodes into some clusters such that each cluster is balanced by the number of member sensor nodes and has exactly one master. The algorithm is based on the minimum weight matching. It optimizes the sum of spatial distances between the member sensor nodes and the master nodes in the whole network. This helps in balancing the load on each master. It also reduces the energy dissipation by the sensor nodes to communicate with the respective master node. Each sensor node and each master node is represented by a vertex in a graph G. A sensor node and a master node pair, such as (x, a_i), is represented by a directed edge from x to a_i in G. Each edge has a weight equal to the energy dissipation required to transmit a message from one vertex to the other vertices of the edge. For example, an edge connecting x and a_i has weight $f(x, a_i)$. A source node S and a sink node T are also added to G as the starting point and the ending point for a message transmission respectively. There are n directed edges from the source node S to n vertices correspond to n sensor nodes. Similarly, there are k directed edges from k vertices assumed to be the master nodes to the sink node T. All edges incident from S or to T are assumed to have weight 0. Finally, the vertices correspond to the sensor nodes are assumed to have the capacity 1, while the vertices correspond to the master nodes are assumed to have the capacity n/k, where $n/k >> 1$. S and T both are assumed to have infinite capacity. Each flow solution in the above graph corresponds to a k-clustering solution. The cost of each flow solution is also equal to the cost of corresponding k-clustering solution, since all the edges adjacent to S or T have zero cost and the other edges have cost equal to energy dissipation between the corresponding sensor and master nodes. Figure 3-2 shows an example graph built on a sample sensor network.

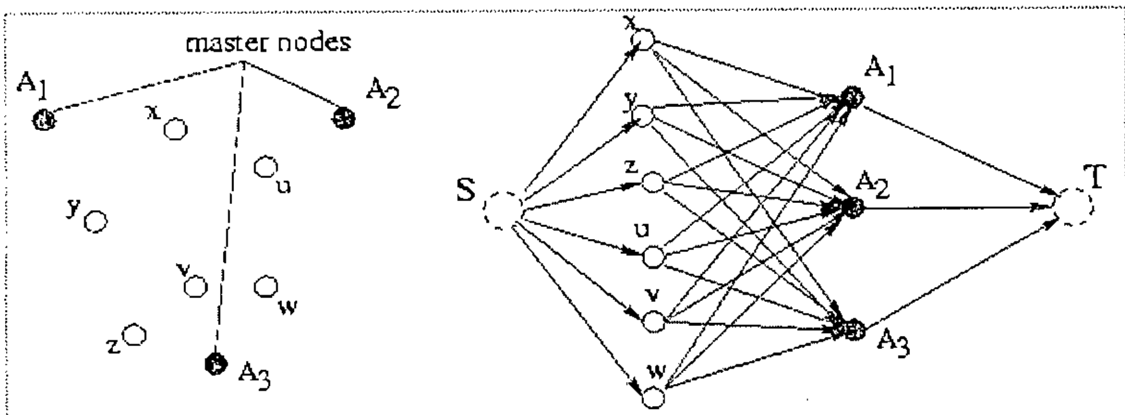


Figure 3-2 Transforming a balanced k-clustering instance to a minimum cost flow instance. Each sensor node has unit capacity, while each master node has capacity n/k.

This approach illustrates an optimal algorithm for clustering the sensor network such that each cluster is balanced and the overall distance between the sensor and the master nodes is minimized. It effectively distributes the network load on all the masters and reduces the communication overhead and the energy dissipation. However, this research work does not consider of residual energy level while choosing a node as the master. Hence, the choice of the master or cluster head is far away from the optimal energy efficient distribution of the cluster heads.

ACE [13] is a distributed clustering algorithm which establishes clusters into two phases-spawning and migration. There are several iterations in each phase and the gap between two successive iterations follows uniform distribution. During the spawning phase, new clusters are formed in a self-elective manner. Every node discovers its neighbors first. A node will elect itself as a temporary cluster head if the number of its neighbors is greater than a pre-specified threshold. When a node decides to become a cluster head, it will broadcast a message to its neighbors to become its followers. A node can receive broadcast messages from more than one cluster head. It randomly chooses a single cluster head from them and broadcasts this information periodically. During migration phase, existing clusters are maintained and rearranged, if required. Migration of an existing cluster is controlled by the cluster head. Each cluster head will periodically poll all of its followers to determine which could be the best candidate to elect as a new leader for the cluster. The best candidate is one which has the largest sum of the same cluster neighbors and cluster free neighbors. The neighbors who are the member of a different cluster will not be included in the sum. This selection of a new cluster head will help to minimize the level of overlapping among the existing clusters. Current cluster head will promote the best candidate as the new cluster head and abdicate itself from its position. ACE results in uniform cluster formation with a packing efficiency close to hexagonal close-packing. ACE clusters are an efficient cover of the network with significantly less overlapping. However, ACE does not consider the residual energy of the nodes while selecting cluster heads. Hence, the clustering is far away from the optimal energy efficient.

HEED [14] introduces a distributed algorithm considering the residual energy of sensor nodes. It results in some clusters by uniformly distributing the cluster heads across the network. It periodically selects cluster heads according to a hybrid parameter which consists of a primary parameter, the residual energy of a node, and a secondary parameter, such as propinquity of a node to its neighbors or node degree. Here, an initial percentage, C_{prob}, is set randomly such that only C_{prob} percentage of nodes from n nodes can become cluster heads. Individual node sets its probability of becoming a cluster head, CH_{prob}, as follows:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{\max}}$$

where, $E_{residual}$ is the estimated current residual energy in the node and E_{max} is a reference maximum energy corresponds to the energy level of a fully charged battery. The CH_{prob} value of a node, however, is not allowed to fall below a certain threshold p_{min}. During any iteration i, every "uncovered" node elects itself as the cluster head with the probability CH_{prob}. If a node elects itself as a cluster head, it sends an announcement message. At the end of iteration i, the set of tentative cluster heads S_{CH} contains the new heads elected in this iteration and the cluster heads from iteration i - 1. A non-cluster-head node v_i selects its cluster head from S_{CH} to which it needs minimum energy to transmit a packet. Every node then doubles its CH_{prob} and goes to the next iteration. A cluster head can relegate itself to a regular node in a later iteration if it finds itself covered by another cluster head using average minimum energy in transmission. HEED converges in $O(1)$ iterations using low messaging overhead and achieves fairly uniform cluster head distribution across the network. With the appropriate bounds on node density and intra-cluster and inter-cluster transmission ranges, HEED can asymptotically guarantee connectivity of clustered networks. However, the random choice of the initial percentage of cluster heads, C_{prob}, remains as a severe limitation of this algorithm.

PADCP [15] uses several adaptive schemes like dynamic cluster range, dynamic transmission power and cluster head re-election to form clusters. In this approach, the sensor nodes are assumed to have the same transmission capability and the ability to adjust transmission power in five levels. Each cluster head can choose the minimum transmission power from level 1 to 5 to connect to different cluster heads of different distances. Level 5 is used to guarantee the connectivity between cluster heads. PADCP has four major phases. In the first phase, each node collects the neighbor information and creates a look-up table by broadcasting messages with first four transmission power levels. In the second phase, cluster heads are elected based on a cost function. The cost function is as follows:

$$C_v = w_A A_v + w_E E_v + w_M M_v$$

where, A_v is the average minimum power that indicates the intra-cluster communication cost, E_v is the ratio between the maximum energy and the remaining energy that indicates the impact of residual energy and M_v indicates the probability of becoming cluster head due to mobility. w_A, w_E, and w_M are their respective weights. In the third phase, clusters are formed using the same method that has been used in HEED and discussed earlier. In the final phase, cluster head is re-elected if its residual energy falls below a pre defined threshold value. PADCP improves the load balance when the sensor nodes are non-uniformly dispersed. The mobility of the sensor nodes is also considered in cluster formation. However, it suffers from the same randomly chosen initial probability limitations of HEED as it completely follows HEED algorithm for cluster formation in its phase 3. It is also hard to know which weight values in the cost function and which threshold value in the re-election phase will give optimum results.

LEACH [1] introduced a simple mechanism for localized coordination and control for cluster set-up and operation. It also introduces the randomized rotation of the cluster heads and the corresponding clusters. However, it does not consider the variation of the initial energy nor the residual energy of sensors during cluster head selection. Other limitations of LEACH have been discussed later in this chapter. SEP [16], a LEACH variant, modifies the equation of the threshold. However, it considers two types of nodes only, normal and advanced, instead of many types that can be encountered in the wireless sensor network after a significant amount of time of operation. Deterministic Cluster Head Selection [17], another variant of LEACH also modifies the threshold to accommodate the heterogeneity of residual energy based on some heuristics. It has several limitations discussed later in this chapter. LEACH-C, proposed by the same authors of LEACH in [18], is a centralized technique which selects the cluster heads based on their positions. It considers uniform distribution of the cluster heads based on their positions and the average residual energy in the network. They did not consider the relative residual energy in each sensor node. Adaptive Cluster Head Selection [19], a distributed clustering technique based on LEACH, considers the positions but not the relative residual energies of the sensor nodes. We explored LEACH and its variants in this research work. For this reason, we describe LEACH and those variants in detail in the following sections.

3.1 LEACH: Low Energy Adaptive Clustering Hierarchy

LEACH is a self-organizing, and adaptive clustering protocol [1]. It dynamically creates clusters in order to distribute the energy load evenly among all of the sensor nodes. This algorithm needs time synchronization. Cluster heads are randomly rotated during each time interval. The resultant cluster heads directly communicate with the base station.

3.1.1 Mechanism

In LEACH, the lifetime of the network is divided into some discrete, disjoint time intervals. Each interval is again divided into some subintervals as shown in Figure 3-3. Each subinterval begins with an advertisement phase followed by a cluster set up phase. In the advertisement phase, each node independently decides whether to become a cluster head or not. In the cluster set-up phase, the clusters are organized based on the decisions made in the advertisement phase. Then a steady-state phase follows. In this phase, the followers, i.e., the sensor nodes except cluster heads, will send data to the corresponding cluster head. The cluster heads accumulate and compress the received data with its own data. Cluster heads send the compressed data to the base station. In order to minimize cluster establishment overhead, the duration of steady-state phase must be longer than that of cluster set-up phase.

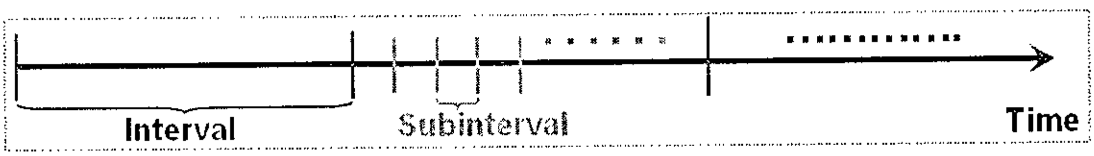


Figure 3-3 discrete and disjoint intervals in the whole network lifetime; discrete and disjoint subintervals in an interval.

At the very beginning of advertisement phase, each node decides whether it wants to become a cluster head for the current round. This decision is based on the suggested percentage of cluster heads for the network, which is set a priori. This decision also depends on the number of times the node has already been a cluster head. This decision is made by a node n choosing a random number between 0 and 1. If the number is less than a threshold $T(n)$, the node decides to become a cluster head. The threshold is calculated as follows:

$$T(n) = \begin{cases} \dfrac{P}{1 - P \times \left(r \bmod \dfrac{1}{P} \right)} & \mathit{if}\ n \in G \\ 0 & \mathit{otherwise} \end{cases}$$

where,

P = the percentage of nodes that can become cluster heads (e.g. P = 0.05);

$1/P$ = the number of subintervals in an interval;

r = the current subinterval;

G = the set of nodes that have not been cluster heads yet in the current interval.

Using this threshold, a node can be a cluster head in any one of $1/P$ subintervals in an interval. At the first subinterval of an interval (r = 0), each node has a probability P to become a cluster head. The nodes that are cluster heads in the first subinterval cannot be cluster heads in the next ($1/P$ – 1) subintervals of the same interval. Thus the probability that the remaining nodes are becoming cluster heads is increasing. After the completion of $1/P$ subintervals, a new interval will start and all the nodes are again eligible to become cluster head.

Each node that has chosen itself as a cluster head in the current subinterval, broadcasts an advertisement message to the rest of the nodes. The non-cluster-head nodes will choose the cluster to which it will belong in this subinterval. This decision is based on the received signal strength of the advertised message. Assuming symmetric propagation channels, the cluster head whose advertisements have been heard with the largest signal strength will be selected by a non-cluster-head sensor node as its cluster head. In case of a tie, a cluster head is chosen randomly.

This algorithm introduced a fairly simple strategy which is more efficient than the direct transmission and the minimum-transmission-energy (MTE) protocol. However, it has several limitations briefly described in the next subsection.

3.1.2 Limitations of LEACH

1. LEACH algorithm uses the desired percentage of cluster heads as a parameter. However, there is no suggestion for which value of this parameter LEACH will ensure optimum energy consumption.

2. LEACH always wants to achieve an even distribution of energy consumption which might not be rational. Residual energy in different nodes is not even or same after a significant amount of time of operation. Nodes with higher residual energy should get preference to be elected as cluster head. Otherwise, longer network stability as well as longer network life cannot be ensured.
3. When the number of live nodes becomes small, the number of prospective cluster heads which is equal to the number of live nodes multiplied by desired percentage of heads will also become very small and in some cases it may become less than one. For example, if the initial number of sensor nodes is 100 and the desired percentage of heads P is 0.05 then the initial number of prospective heads is 100*0.05=5. However, with the same P when the number of live nodes becomes less than 20 the number of prospective heads will become less than one. Under this condition in most of the subintervals, none of the live sensor nodes can become a cluster head by choosing a random number which is less than the current threshold. In other words, there will be no cluster head available to the sensor nodes to which they can become followers. Rather, all the live sensors will force themselves to become a one member cluster head. Thus, there will be very little energy efficiency due to this sort of clustering.

3.2 SEP: A Stable Election Protocol

SEP [16] is variant of LEACH, which elects the cluster heads based on weighted probabilities according to the residual energy of the sensor nodes. It assumes that a percentage of the sensor nodes are coming with higher energy resources and studies the impact of heterogeneity of nodes based on their energy levels.

3.2.1 Mechanism

This approach follows the underlying synchronization approach used in LEACH. In addition, it considers the variation in the residual energy assuming two types of nodes – normal and advanced. It assumes m fractions of the nodes are advanced nodes, which have α times energy than that of the normal nodes. As a result, it assumes $n(1+ \alpha m)$ number of virtual normal nodes in the network. It extends the number of subintervals from $1/P$ to $(1+ \alpha m)/P$ in an interval. The objective of this extension is to elect a normal node once and an advanced node (1+ α) times as the cluster head in an interval. The probability equation to become cluster head has been modified. In fact, two different equations are used for the normal and the advanced nodes. The weighted election probabilities for the normal and the advanced nodes are p_{nrm} and p_{adv} respectively. Their equations are as follows–

$$p_{nrm} = \frac{p_{opt}}{1+\alpha \times m}$$

$$p_{adv} = \frac{p_{opt}}{1+\alpha \times m} \times (1+\alpha)$$

where, p_{opt} is the optimal probability of a node to become a cluster head.

The equation of the threshold has not only been changed, two different equations for the threshold are used. One for the normal nodes called $T(s_{nrm})$ and the other for the advanced nodes called $T(s_{adv})$. $T(s_{nrm})$ and $T(s_{adv})$ are calculated as follows:

$$T(s_{nrm}) = \begin{cases} \dfrac{p_{nrm}}{1 - p_{nrm} \times \left(r \bmod \dfrac{1}{p_{nrm}} \right)} & \text{if } s_{nrm} \in G' \\ 0 & \text{otherwise} \end{cases}$$

and,

$$T(s_{adv}) = \begin{cases} \dfrac{p_{adv}}{1 - p_{adv} \times \left(r \bmod \dfrac{1}{p_{adv}} \right)} & \text{if } s_{adv} \in G'' \\ 0 & \text{otherwise} \end{cases}$$

where, G' is the set of normal nodes that have not become cluster head yet within the last $1/p_{nrm}$ subintervals and G'' is the set of advanced nodes that have not become cluster head yet within the last $1/p_{adv}$ subintervals in an interval. This works introduced the heterogeneity to LEACH in terms of residual energy. However, this introduction was limited to only two types of nodes. Limitations of SEP are briefly discussed in the next subsection.

3.2.2 Limitations of SEP

1. In SEP, the percentage of cluster heads is optimized based on the energy consumption in an interval. However, this value should be optimized on the basis of the long run rate of energy consumption for achieving the higher network stability period.
2. SEP considers two types of nodes only in terms of residual energy. However, during the life cycle of the network the different levels of the residual energies may exist which will not be covered by only two types. More types of nodes are necessary to consider covering numerous residual energy levels in different nodes to achieve maximum network stability.

3.3 LEACH Variant: Deterministic Cluster Head Selection

Deterministic Cluster Head Selection [17] introduces the heterogeneity to LEACH in terms of residual energy. It considers the residual energies of the sensor nodes in order to manage rational power consumption throughout the network.

3.3.1 Mechanism

Deterministic Cluster Head Selection follows the underlying mechanism of LEACH exactly. It has changed the equation of the threshold value only to incorporate the residual energy in cluster head selection process as follows:

$$T(n)_{new} = \frac{P}{1 - P \times \left(r \bmod \frac{1}{P} \right)} \times \frac{E_{n\_current}}{E_{n\_\max}}$$

where, $E_{n\_current}$ is the current energy, E_{n_max} the initial energy of the node. The other parameters have the same definitions as of LEACH. After a significant amount of time of operation,, the residual energies of the sensors would become very low and then this threshold value will be very low. This can result in a situation where all the live sensors are one member cluster head. In this case the energy consumption rate will be very high. To break this stuck condition another modified equation of the threshold value has been proposed –

$$T(n)_{new} = \frac{P}{1 - P \times \left(r \bmod \frac{1}{P} \right)} \times \left[\frac{E_{n\_current}}{E_{n\_\max}} + \left(r_s \ div \frac{1}{P} \right) \times \left(1 - \frac{E_{n\_current}}{E_{n\_\max}} \right) \right]$$

where, r_s is the number of consecutive rounds in which a node has not been cluster head.

3.3.2 Limitations of Deterministic Cluster Head Selection

1. Like LEACH Deterministic Cluster Head Selection uses a random value for the percentage of heads parameter, hence, does not consider the optimal value of this parameter.
2. It does not suggest any optimum value for r_s either.

3.4 LEACH-C: Centralized LEACH

LEACH-C is a centralized technique to cluster sensor nodes based on their positions. In this approach, base station selects cluster heads to get uniformly distributed clusters.

3.4.1 Mechanism

Sensor nodes detect their current locations using GPS (Global Positioning System) receiver or any other technique. At the beginning of each interval, each node informs the base station its current location and residual energy level. After receiving the information from all the sensor nodes, base station computes the average residual energy in the network. It precludes those sensor nodes whose residual energy is below the average residual energy from attaining cluster headship. Base station selects the cluster heads from the remaining nodes using the simulated annealing algorithm [20]. This algorithm minimizes the total sum of squared distances between all the non-cluster-head nodes and the corresponding closest cluster head node. Thus, it minimizes the amount of energy necessary to use to transmit data to the cluster head nodes by the non-cluster-head nodes. Base station also selects corresponding followers for the clusters while selecting the clusters and cluster heads, and the base station broadcasts a message into the network informing these selections.

3.4.2 Limitations of LEACH-C

1. In LEACH-C, the base station selects the cluster heads based on their positions and the average residual energy in the network. Like LEACH, the individual residual energy in each sensor node has little impact on the cluster head selection process in LEACH-C. This centralized algorithm also suffers from non-scalability.
2. Incorporating GPS receiver or similar device in the sensor nodes increases sensor node cost.

3.5 Adaptive Cluster Head Selection

Adaptive Cluster Head Selection [19] assumes that a sensor node knows its distance from another sensor node by observing the signal strengths in the received messages.

3.5.1 Mechanism

At first, this approach randomly selects cluster heads following LEACH. Next it reselects the cluster heads considering the distance between each cluster head and the sensor nodes farthest from the cluster heads. The reselection is done in order to distribute the cluster heads uniformly in the network. When a sensor node is selected as a cluster head by LEACH, it broadcasts an advertisement message to all other nodes. Other sensor nodes respond to the broadcast. From the received responses, a cluster head calculates its distance to its farthest follower node and its distance to the nearest cluster head of neighbor clusters. It subtracts the first distance from the later. Three cases may arise as follows:

Case 1: The result is positive.

Case 2: The result is negative.

Case 3: The result is zero.

In order to place the cluster head in an optimum location, the cluster head is moved to the direction of the closest head in Case 1 and to the direction of the farthest sensor node in Case 2. Cluster head position remains the same in Case 3.

3.5.2 Limitations of Adaptive Cluster Head Selection

1. In this work cluster head movement, if necessary, is not clearly defined.
2. It completely ignores the relative residual energy of each sensor node in the network while selecting the cluster heads.
3. It also suffers from other LEACH limitations.

Though LEACH is widely used clustering technique for WSN, it does not have a complete mathematical model that can be used to tune LEACH performance by selecting different values for different LEACH parameters. In this thesis, we have first proposed a mathematical model of LEACH in the next chapter.

4 Proposed Mathematical Model for LEACH

The primary reason behind the clustering technique is to reduce the rate of energy consumption. Popular clustering techniques LEACH [1] and its variants [16], [17], have achieved a significant amount of lower rate of energy consumption. All of these techniques have been developed based on some heuristics rather than a complete mathematical model. A mathematical model can serve better than those heuristics to achieve an optimal rate of energy consumption. Moreover, a mathematical model can provide the ways to tune application specific parameters. For this reason, we derive a complete mathematical model of energy consumption rate of LEACH. We describe our mathematical model in this chapter.

4.1 Preliminaries

We use some basic assumptions about the sensor nodes and the network settings while developing our mathematical model for LEACH. After describing the basic assumptions, we describe some base models that have also been used in the formulation of our mathematical model.

4.1.1 Assumptions

Original LEACH algorithm uses following assumptions about the sensor nodes and network settings:

- Nodes do not have any location information.
- All nodes can reach the BS.
- The propagation channel is symmetric.

We also use these assumptions while formulating the mathematical model in order to keep our model fully aligned with LEACH algorithm.

4.1.2 Base Models

Heinzelman proposed an energy model namely first order radio model for energy consumption in a wireless network in [21]. Like other research works [1], [16], [17], we use this first order radio model to compute the expected energy consumption rate in sensor networks.

Energy consumption due to the reception and the transmission of data in a sensor network is a stochastic process. We use the Renewal Reward stochastic process to capture the nature of energy consumption due to data transmission and reception by a sensor node. In the following subsections, we briefly describe these two base models.

4.1.2.1 Energy Model: Heinzelman's First Order Radio Model

A sensor node consumes energy to run the circuitry, which is proportional to the number of bits in the message under processing. For example, if the message contains *k* bit and the energy per bit is E_{elec} Joules, then the energy used to run the circuitry will be (E_{elec} * *k*) Joules. A sensor node consumes this energy while receiving and processing a message. Therefore, the energy consumed by a receiving node to receive a *k*-bits message is,

$$E_{R_X}(k) = (E_{elec} * k) \tag{1}$$

The energy needs to send k bit message over a distance *d* is ($\epsilon_{amp} * k * d^{\lambda}$) Joules, where ϵ_{amp} is the energy constant for the radio transmission and λ is the path loss exponent. While transmitting, a sensor node needs energy to run the circuitry as well as to send the message. We consider Heinzelman's first order model where $\lambda = 2$. Therefore, the total energy consumed by a transmitting node to send a *k*-bits message over distance *d* is,

$$E_{T_X}(k,d) = (E_{elec} * k) + (\in_{amp} * k * d^2) \tag{2}$$

This model is shown in Figure 4-1.

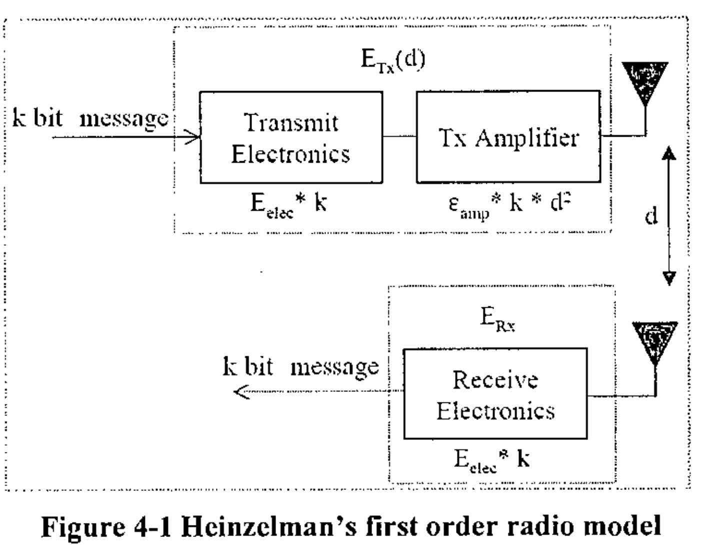


Figure 4-1 Heinzelman's first order radio model

4.1.2.2 Renewal reward process

A renewal process is special counting process $N(t)$ which counts the number of events up to time t and the inter-arrival times of the events are independent and identically distributed (*iid*) random variables. The expected value of inter-arrival times is in between zero and infinity. A renewal reward process is a renewal process such that there are some rewards for each of the inter-arrival times. These rewards are also independent and identically distributed (*iid*) random variables. If, X_i is the ith inter-arrival time and R_i is reward for the inter-arrival time X_i, the total reward earned up to time t will be:

$$R(t) = \sum_{i=1}^{N(t)} R_i \tag{3}$$

According to renewal reward theorem, the rate of reward will be:

$$\lim_{t \to \infty} \frac{R(t)}{t} = \frac{E(R)}{E(X)} \tag{4}$$

This means that the rate of reward is equal to the ratio between the expected reward in a single inter arrival time $E(R)$ and the expected inter-arrival time $E(X)$ in the long run. The theorem can be proved based on Strong Law of Large Numbers and is out of scope of the this thesis. In stochastic process, the inter arrival time is also called a cycle.

4.2 Proposed Mathematical Model

As the part of our mathematical analysis, we calculate the expected energy consumption rate following the renewal-reward process. We consider the energy consumed by the sensor as the reward. Then, the long run rate of reward will essentially be the long run rate of energy consumption. However, to map this problem with renewal-reward process perfectly, we have to define cycle in such a way that both the reward and the cycle can be treated as *iid* random variables.

According to LEACH algorithm, in the first subinterval of an interval each live sensor node will have some non zero probability to become cluster head. However, in the other subintervals a sensor node has zero probability to become cluster head, if it became a cluster head in the first or any other previous subinterval. It must be a follower in all other subsequent subintervals in the same interval. We define a cycle as the number of subintervals between two consecutive subintervals in which a sensor node becomes cluster head. Cluster establishment is probabilistically done in each subinterval. Hence, the cycle or inter-arrival time is an integer number and *iid* random variable. Similarly energy consumption by a sensor node in each cycle is an *iid* random variable. These definitions of cycle and reward map our problem to a renewal-reward process appropriately. Thus, the long run rate of reward in Equation 4 gives the expected energy consumption rate in a subinterval. We need to compute $E(R)$ and $E(X)$ to derive the energy consumption rate. We define following parameters for this purpose.

1. P be the desired percentage of cluster heads,
2. s be the number of subintervals in an interval ($= 1/P$),
3. P_h be the probability of becoming cluster head of a follower node at the start of any subinterval,
4. P_h' be the probability of becoming cluster head of a cluster head node at the start of a subinterval in next interval,
5. Φ_0 be the probability of becoming cluster head of a sensor node at the start of any subinterval,
6. T_h be the currently considered threshold value.
7. N be the total number of sensor nodes in the network.
8. $a * b$ be the sensor area in two dimensions.

4.2.1 Calculation of $E(X)$

We compute expected cycle length, $E(X)$, of Equation 4 in this section. At the beginning of each subinterval new cluster heads are selected and new clusters are generated. Each sensor will generate a random number between 0 and 1 and compares it to a predefined threshold value T_h. If the random number is less than the threshold, the sensor node becomes cluster head. Otherwise, the sensor node acts as a follower. We can show that the transitions between two states (heads: h and follower: f) of a sensor node while changing the subinterval in an interval by following matrix:

$$\begin{array}{cc} & \begin{array}{cc} h & \quad f \end{array} \\ \begin{array}{c} h \\ f \end{array} & \begin{pmatrix} 0 & 1 \\ P_h & 1-P_h \end{pmatrix} \end{array}$$

If the interval is changed then the probability of becoming head while changing the subinterval will be the same irrespective of the previous state. Therefore, we can show the transitions between two states of a sensor node while changing the subinterval as well as the interval by following matrix:

$$\begin{array}{c c} & \begin{array}{cc} h & f \end{array} \\ \begin{array}{c} h \\ f \end{array} & \begin{pmatrix} P_h' & 1-P_h' \\ P_h & 1-P_h \end{pmatrix} \end{array}$$

Above behaviors of sensor nodes in LEACH can be shown by the transition diagrams in Figure 4-2.

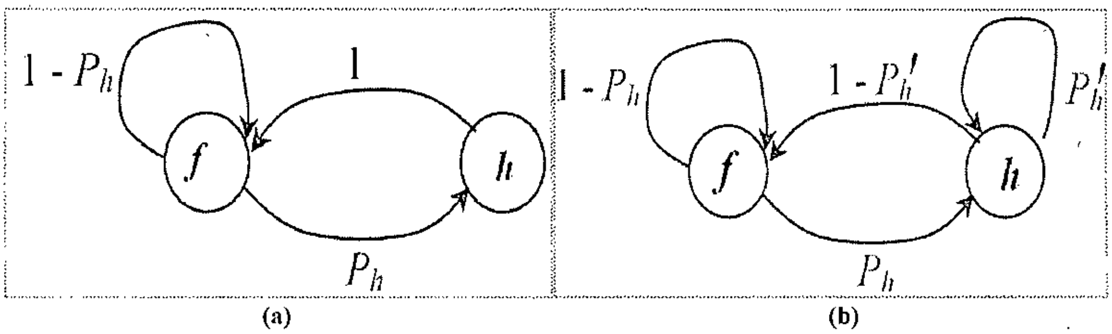


Figure 4-2 State Transition of a Node while (a) Changing Subinterval without changing Interval, (b) Changing Subinterval as well as the Interval

A sensor node can become cluster head at the start of the first subinterval of a new interval based on the picked random number and the threshold. This decision does not depend on whether it was cluster head or follower in the last subinterval of previous interval. In this case, the probability of a follower to become a cluster head and the probability of a cluster head to remain cluster head are same, i.e., $P_h' = P_h$.

The number of subintervals in an interval is s. Therefore, a sensor node remains in the same interval up to (s-1) subinterval transitions and moves to the next interval only at the last subinterval transition. From this observation, we can say that the probability of remaining in the same interval is equal to (s-1)/ s and the probability of changing the interval is equal to $1/s$.

We combine these probabilities with their corresponding transition matrices in order to capture the whole scenario.

$$\frac{s-1}{s}\begin{pmatrix} 0 & 1 \\ P_h & 1-P_h \end{pmatrix} + \frac{1}{s}\begin{pmatrix} P_h & 1-P_h \\ P_h & 1-P_h \end{pmatrix} = \begin{pmatrix} \frac{P_h}{s} & \frac{s-P_h}{s} \\ P_h & 1-P_h \end{pmatrix} \qquad (5)$$

Hence, the combined transition matrix becomes as follows:

1. The probability of a cluster head to remain cluster head, P_{hh} at the start of any subinterval is P_h/s.
2. The probability of a follower to become a cluster head, P_{hf} at the start of any subinterval is P_h.
3. The probability of a cluster head to become a follower, P_{fh} at the start of any subinterval is $(s-P_h)/s$.
4. The probability of a follower to remain follower, P_{ff} at the start of any subinterval is $1-P_h$.

Now, we can compute the probability of becoming a cluster head, Φ_0, at the start of any subinterval by summing up the first two values P_{hh} and P_{hf} as follows:

$$\Phi_0 = \left(\frac{P_h}{s}\right) + P_h = (s+1) * \left(\frac{P_h}{s}\right) \tag{6}$$

We can say that the expected value of the cycle is reciprocal of the probability of becoming cluster head, Φ_0, at the start of any subinterval, i. e. -

$$E(X) = \frac{1}{\Phi_0} = \frac{s}{(s+1) * P_h} \tag{7}$$

In order to compute P_h we need to consider following two conditions -

1. A node can be a cluster head if the picked random number is lower than the threshold. In LEACH, the threshold is maintained in a way such that the mean value of the threshold becomes the percentage of sensor nodes to be elected as the cluster heads in the network. Hence, the probability of becoming cluster head in this way is equal to the said percentage, i.e., $P_{h1} = P$.

2. If none of the nodes pick the random number less than that of the threshold, all nodes act as one-member cluster head. The probability of becoming one-member cluster head in this way is, $P_{h2} = (1 - P)^N$.

Therefore, the ultimate probability of becoming a cluster head, P_h while changing subinterval in an interval will be $P + (1 - P)^N$.

Hence, expected cycle length E(X) can be calculated by substituting P_h from Equation 7.

4.2.2 Calculation of $E(R)$

We compute expected reward (energy consumption), $E(R)$, of Equation 4 in this section. Energy consumption by a sensor node as a cluster head differs from that of a sensor node as a follower. Let -

1. H be the amount of energy consumed by a cluster head in a single subinterval and
2. F be the amount of energy consumed by a follower in a single subinterval.

In a cycle, the expected number of subintervals in which a sensor node remains a follower is $E(X)$ - 1 and the expected number of subintervals in which a sensor node remains a cluster head is 1. Therefore, the amount of energy consumed by a sensor node in a single cycle is -

$$E(R) = (E(X) - 1) * E(F) + E(H) \tag{8}$$

Here, $E(F)$ and $E(H)$ are the expected values of energy consumed by a follower and a cluster head, respectively, in a single subinterval. $E(X)$ has already been calculated in Section 4.2.1. We need to calculate $E(F)$ and $E(H)$ in order to find $E(R)$ of Equation 8.

4.2.2.1 Calculation of $E(F)$

We can compute the expected value of energy, $E(F)$, consumed by a follower in a single subinterval using Heinzelman's first order radio model [21]. Being a follower, a sensor node consumes energy only for transmitting. According to Heinzelman's first order radio model, the total energy to transmit a k-bit message over distance X is

$$E(F|X = x) = (E_{elec} * k) + (\in_{amp} * k * x^2) \tag{9}$$

If $f(x)$ is the distribution function of the distance X of a follower to its nearest cluster head, the energy consumption by a follower will be:

$$\begin{aligned} E(F) &= \int E(F|Distance\ to\ nearest\ cluster\ head = x) f(x)dx \\ &= \int ((E_{elec} * k) + (\in_{amp} * k * x^2)) f(x)dx \\ &= (E_{elec} * k) + (\in_{amp} * k) \int x^2 f(x)dx \end{aligned} \tag{10}$$

Now, we calculate the distribution function of the distance, $f(x)$.

$$f(x) = \frac{d}{dx} P(X \le x) \tag{11}$$

There might be several cluster heads at the nearest distance. Therefore,

$$\begin{aligned} P(X \le x) &= P(at\ least\ 1\ cluster\ head\ is\ at\ distance\ of\ x\ or\ less\ than\ x) \\ &= 1 - P(no\ cluster\ head\ is\ inside\ the\ area\ with\ radius\ x) \\ &= 1 - P(all\ cluster\ heads\ are\ outside\ the\ area\ with\ radius\ x) \end{aligned} \tag{12}$$

Now, if the number of cluster heads is N_c then,

$$P(all\ cluster\ heads\ are\ outside\ the\ area\ with\ radius\ x)$$
$$=\sum_{n=1}^{N} P(all\ cluster\ heads\ are\ outside\ \pi * x^2 | N_c = n)\, P(N_c = n)$$
$$=\sum_{n=1}^{N}\binom{N}{n}\left(\frac{s+1}{s}P_h\right)^n\left(1-\frac{s+1}{s}P_h\right)^{N-n}\left(1-\frac{\pi * x^2}{ab}\right)^n \quad (13)$$

Therefore, we can calculate the distribution function of the distance, $f(x)$ as follows:

$$f(x)=\frac{d}{dx}P(X\le x)$$
$$=\sum_{n=1}^{N}\binom{N}{n}\left(\frac{s+1}{s}P_h\right)^n\left(1-\frac{s+1}{s}P_h\right)^{N-n}\frac{2n\pi x}{ab}\left(1-\frac{\pi x^2}{ab}\right)$$
$$=\sum_{n=1}^{N} g(n) * x * \left(1-\frac{\pi x^2}{ab}\right)^{n-1} \quad (14)$$

where,

$$g(n)=\binom{N}{n}\left(\frac{s+1}{s}P_h\right)^n\left(1-\frac{s+1}{s}P_h\right)^{N-n}\frac{2n\pi}{ab}$$

Let,

$$I = \int x^2 * f(x)dx$$
$$=\sum_{n=1}^{N}\int g(n) * x^3 * \left(1-\frac{\pi x^2}{ab}\right)^{n-1} dx \quad (15)$$

After solving the integration at the right side of the above equation, we get –

$$I=\frac{1}{2}\left(\frac{ab}{\pi}\right)^2\sum_{n=1}^{N} g(n)\left(\frac{y^{n+1}}{n+1}-\frac{y^n}{n}\right) \quad (16)$$

where,

$$y = 1-\frac{\pi x^2}{ab}$$

The subtracted value of y indicates the proportion between two areas of Figure 4-3. Here, the first area is the area inside the circle with the center at the sensor node under consideration and the radius equal to the distance from the sensor node to its nearest cluster head. The second area is the total area covered by all the sensor nodes. If the cluster head position coincides with that of the node, we get the lower limit of x and $\frac{\pi x^2}{ab}$

equal to zero. In this case, y value becomes 1. If the cluster head is positioned at a position such that the first area fully covers the second area, we get the higher limit of x and $\frac{\pi x^2}{ab}$ equal to 1. In this case, y value becomes 0.

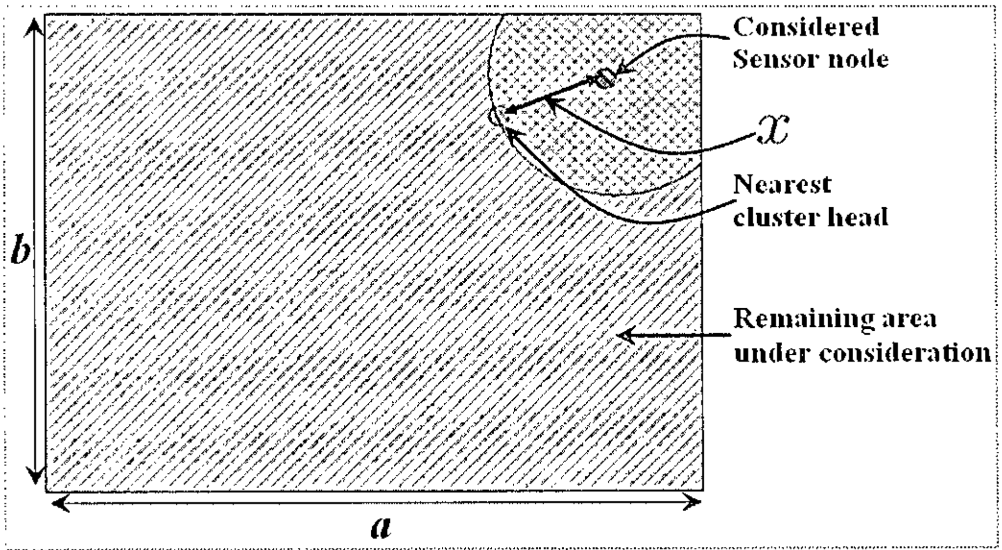


Figure 4-3 y is the ratio between two areas. First one is the remaining area under consideration and the second one is the total area under consideration

Therefore, the integrated value with the limits of y is:

$$[I]_1^0 = \frac{1}{2}\left(\frac{ab}{\pi}\right)^2 \sum_{n=1}^{N} g(n)\left(\frac{1}{n} - \frac{1}{n+1}\right) \tag{17}$$

Combining Equations 10 and 17, we get the expected value of energy consumption of a sensor node as a follower in a subinterval as follows:

$$E(F) = (E_{elec} * k) + (\in_{amp} * k) * \left\{\frac{1}{2}\left(\frac{ab}{\pi}\right)^2 \sum_{n=1}^{N} g(n)\left(\frac{1}{n} - \frac{1}{n+1}\right)\right\} \tag{18}$$

4.2.2.2 Calculation of *E(H)*

We can also compute the expected value of energy, *E(H)*, consumed by a cluster head in a single subinterval using Heinzelman's first order radio model [21]. The cluster head aggregates and compresses the data to be relayed from its followers with its own data before sending them to the base station. Therefore, the actual number of bits sent by a cluster head is less than the summation of the numbers of the bits of all the messages those it handles. Let, γ be the compression ratio. If there are N_f followers and each sensor node generates k bit message, according to Heinzelman's first order radio model, the energy consumption by a cluster head will be:

$$E(H|N_f) = (2N_f+1)kE_{elec} + (N_f+1)\in_{amp} k\gamma x^2 \tag{19}$$

Now,

$$\begin{aligned} E(H) &= \sum_{i=0}^{N-1} E(H|N_f=i)\,P(N_f=i) \\ &= \sum_{i=0}^{N-1}\left[\left\{(2i+1)k\,E_{elec} + (i+1)\in_{amp} k\gamma d_{BS}{}^2\right\} * P(N_f=i)\right] \end{aligned} \tag{20}$$

Since N_c is the total number of cluster heads we can write,

$$P(N_f=i) = \sum_{n=1}^{N-i} P(N_f=i|\,N_C=n)\,P(N_C=n) \tag{21}$$

Here,

$$P(N_C=n) = \binom{N}{n}\Phi_0^n(1-\Phi_0)^{N-n} \tag{22}$$

and

$$P(N_f=i|\,N_C=n) = \binom{N-n}{i}P(A)^i(1-P(A))^{N-n-i} \tag{23}$$

Here, A is an event that ensures that the cluster head under consideration is the nearest cluster head to a follower. If the location of the cluster head is (x_h, y_h) and the location of the follower is (x, y), we can write

$$P(A) = \frac{\pi r^2}{ab}p_a\left(1-\frac{\pi r^2}{ab}p_a\right)^{n-1} \tag{24}$$

where, $r = \sqrt{((x - x_h)^2 + (y - y_h)^2)}$ and p_a is the percentage of the circular area (centered at the follower and with radius r) falls within the area covered by the sensor network. Let, $P(A) = h(r, n)$. Combining Equations 21, 22, and 23, we get

$$P(N_f=i) = \sum_{n=1}^{N-i}\left[\binom{N-n}{i}h(r,n)^i(1-h(r,n))^{N-n-i}\binom{N}{n}\Phi_0^n(1-\Phi_0)^{N-n}\right] \tag{25}$$

Combining Equations 6, 20, and 25, we get –

$$E(H) = \sum_{i=0}^{N-1}\left[\left\{(2i+1)kE_{elec} + (i+1)\in_{amp} k\gamma {d_{BS}}^2\right\} * \sum_{n=1}^{N-i}\left\{\binom{N-n}{i} h(r,n)^i (1-h(r,n))^{N-n-i} \binom{N}{n}\left(\frac{s+1}{s}P_h\right)^n \left(1-\frac{s+1}{s}P_h\right)^{N-n}\right\}\right] \quad \textbf{(26)}$$

4.3 Energy Consumption Rate

Combining Equations 7, 8, 18, and 26, we can get energy consumption rate as follows:

$$\begin{aligned}\lim_{t\to\infty}\frac{R(t)}{t} &= \frac{E(R)}{E(X)} \\ &= \left(1-\frac{s+1}{s}P_h\right)*\left[(E_{elec}\,k)+(\in_{amp} k)\frac{1}{2}\left(\frac{ab}{\pi}\right)^2\sum_{n=1}^{N} g(n)\left(\frac{1}{n}-\frac{1}{n+1}\right)\right] \\ &\quad + \frac{s+1}{s}P_h\sum_{i=0}^{N-1}\left[p(i)*\sum_{n=1}^{N-i} q(i,n)\right]\end{aligned} \quad \textbf{(27)}$$

where,

$$p(i) = (2i+1)kE_{elec} + (i+1)\in_{amp} k\gamma {d_{BS}}^2$$

and,

$$q(i,n) = \binom{N-n}{i} h(r,n)^i (1-h(r,n))^{N-n-i}\binom{N}{n}\left(\frac{s+1}{s}P_h\right)^n\left(1-\frac{s+1}{s}P_h\right)^{N-n}$$

Equation 27 concludes the formulation of our mathematical model. This equation evaluates the expected energy consumption rate in a wireless sensor network. The optimal number of cluster heads can also be determined using this equation. In the next chapter, we propose a new clustering technique for WSN based on some heuristics and modify the mathematical model accordingly.

5 Proposed Protocol: Cluster Heterogeneous Sensor Network (CHSN)

In this chapter, we propose a new algorithm namely "Cluster Heterogeneous Sensor Network (CHSN)" to cluster sensor nodes of a heterogeneous sensor network. We consider the heterogeneity of sensor nodes in terms of their residual energy levels, which makes our work more practical and useful over the original LEACH algorithm. We use three heuristics on LEACH to enhance the performance. We also modify the mathematical model derived in the previous chapter accordingly. After describing those heuristics, we describe our clustering algorithm in details in this chapter.

5.1 Heuristic 1

In the original LEACH algorithm if a node becomes cluster head in a subinterval, it cannot become cluster head again in any of the subsequent subintervals of the same interval. However, if a sensor node with higher residual energy can become cluster head again in the same interval it might be more energy efficient for the whole network. For this reason, we make the subintervals completely memory less and don't use a separate set of nodes that have not been cluster head yet in the current interval. In this case, the probability of becoming cluster head of a sensor node in a subinterval does not depend on its status in the previous subintervals.

5.2 Heuristic 2

It will be more energy efficient for the sensor network if the nodes with higher residual energy have the higher probability to become cluster head. For this reason, we consider relative residual energy of a sensor node while selecting the cluster heads. Accordingly, we map the relative residual energy of a sensor node in its threshold computation so that it keeps its expected value at the optimal percentage of cluster heads P. At the beginning of each subinterval, each node knows its own residual energy (E_{cur}) and the maximum (E_{cur_max}), the minimum (E_{cur_min}), and the average (E_{cur_avg}) residual energies of all the sensor nodes alive in the network. Considering E_{cur_avg} corresponds to P, we map E_{cur_min}, and E_{cur_max} to $(1-P_{range})$ and $(1+P_{range})$ respectively, where P_{range} is the minimum between P and $(1-P)$. If $P \leq (1-P)$, $(P-P_{range})$ becomes zero and if $P \geq (1-P)$, $(P+P_{range})$ becomes one. This has been shown in Figure 5-1.

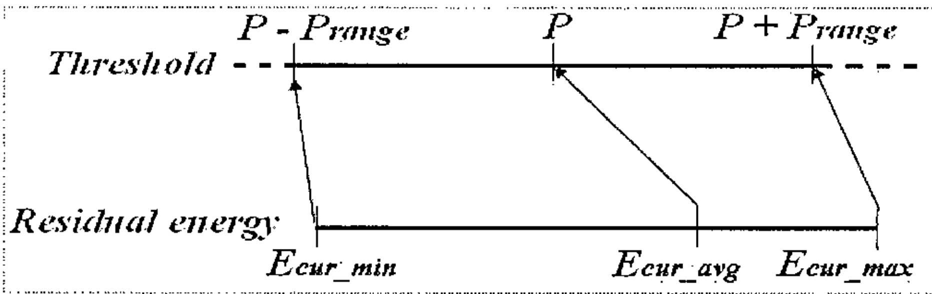


Figure 5-1 Distribution of Threshold Value according to Residual Energy

We define ΔP, the deviation from P that should be considered for a sensor node based on the difference between its residual energy E_{cur} and the average residual energy E_{curr_avg} in the network. Hence,

$$\Delta P = P_{range} * E_r \qquad \textbf{(28)}$$

where,

$$E_r = \begin{cases} \dfrac{E_{cur} - E_{cur\_avg}}{E_{cur\_avg} - E_{cur\_\min}}, & if \ E_{cur} < E_{cur\_avg} \\ 0, & if \ E_{cur} = E_{cur\_avg} \\ \dfrac{E_{cur} - E_{cur\_avg}}{E_{cur\_\max} - E_{cur\_avg}}, & if \ E_{cur} > E_{cur\_avg} \end{cases}$$

In order to make the threshold value proportional to the residual energy of a sensor node, we assign threshold value equal to P plus ΔP, i.e.,

$$T(n) = P + \Delta P \qquad \textbf{(29)}$$

5.3 Heuristic 3

After a long duration of service from the initial deployment, the number of live nodes, N_{live}, becomes so small that the number of probable heads ($N_{live} * \Phi_0$) become very small and even less than one. If this situation occurs, the threshold from Equation 29 may frequently cause the cluster selection algorithm to choose only one member clusters. In this case, all the live sensor nodes become the cluster head of its one member cluster and die very quickly. To prevent this unwanted situation, we preserve the optimal initial number of cluster heads rather than the optimal percentage of cluster heads. For this reason, we multiply the right side of Equation 29 by the ratio between the initial number of sensor nodes (N) and the number of currently live sensor nodes (N_{live}) in the network as follows:

$$T(n) = (P + \Delta P) * \frac{N}{N_{live}} \tag{30}$$

5.4 CHSN Algorithm

As it is done in LEACH, we divide the lifetime of the network into some discrete equal length intervals in CHSN. Each interval has three consecutive phases - advertisement, cluster-setup, and steady-state phase. The algorithm depicted in Figure 5-2 runs independently in each sensor node in each interval. The parameters are initialized at the start of the algorithm. E_{cur} is set to its current residual energy level. E_{cur_max}, E_{cur_min}, and E_{cur_avg} are set to its own current residual energy level, i.e., equal to E_{cur}. The number of live sensor node, N_{live}, is set to one assuming it is the only live sensor node in the network. Advertisement, cluster-setup, and steady-state phases are executed as follows:

I. **Advertisement Phase:** During this phase, each node executes two parallel processes. One of them broadcasts its current residual energy level. Before this broadcast, a node waits for a random amount of time. This random delay is used to make the probability of collision lower. Uniformly distributed random amount of time is chosen for this purpose. Another process receives the current residual energy levels of other sensor nodes. A sensor node may receive multiple copies of a current energy level advertisement message from the same sensor node due to multi-path effect. A receiver sensor node detects these duplicate receptions and ignores them. A receiver sensor node updates the parameters – E_{cur_max}, E_{cur_min}, $E_{cur\_avg,}$, and N_{live} using the fresh advertisement messages only.

II. **Cluster Set-up Phase:** In this phase, each sensor node independently decides whether to become a cluster head or not based on the information gathered in the advertisement phase. At first, it calculates the threshold $T(n)$ using Equation 30. Next, it picks a random number and compares the random number with the threshold. Three cases may arise as follows:

1. CASE 1: The random number is less than the threshold. In this case, the sensor node becomes a cluster head and broadcasts HEAD_EXPOSURE message.
2. CASE 2: The random number is not less than the threshold and it does not receive any HEAD_EXPOSURE message from other sensor nodes. In this case, the sensor node becomes a one member cluster head.
3. CASE 3: The random number is not less than the threshold and it receives one or more HEAD_EXPOSURE messages from other sensor nodes. In this case, the sensor node becomes a follower of the nearest cluster head and sends a FOLLOWER_ACCEPTANCE message to the nearest cluster head.

III. **Steady-state Phase:** In this phase, the followers send data to the corresponding cluster head. The cluster heads accumulate, aggregate, and compress the received data with its own data. Cluster heads send the aggregated and compressed data to the base station. The duration of steady-state phase is significantly longer than the summation of the durations of the advertisement and cluster-setup phases in order to minimize cluster establishment overhead.

5.5 New Mathematical Model

Heuristic 1 of our new clustering algorithm makes the subinterval completely memory less. For this heuristic, the first state transition diagram of Figure 4-2 is no longer applicable. The equation of Φ_0 has been formulated form the weighted combination of two state transition diagrams of Figure 4-2. Therefore, the equation of Φ_0 needs to be changed. After the introduction of heuristic 1 the probability of becoming cluster head of a follower node at the start of any subinterval P_h become equal to the probability of becoming cluster head of a sensor node at the start of any subinterval Φ_0. Therefore, $\Phi_0 = P_h$ in the new mathematical model. With this minor change we can use the mathematical model derived in the previous chapter as the new mathematical model for our new clustering algorithm.

Algorithm CHSN ()
    Set the value of E_{cur}
    Initialize E_{cur_max} to E_{cur}
    Initialize E_{cur_min} to E_{cur}
    Initialize E_{cur_avg} to E_{cur}
    Initialize N_{live} to 1
    Advertisement(); broadcasts and receives current energy levels
    Cluster_Set_Up(); Generates the clusters
    Steady_State(); Receive and transmit data

Advertisement()
    Transmit_Current_Energy_Residual_Level()
    Receive_Current_ Residual_Energy_Level()

Transmit_Current_ Residual_Energy_Level()
    Wait for a random time
    Broadcast own current residual energy level

Receive_Current_ Residual_Energy_ Level()
    For each received current residual energy level, E'_{cur}
        if E'_{cur} is not a repetition from an already received node
            Update_Parameters(N_{live}, E'_{cur})
        endif

Update_Parameters(N_{live}, E'_{cur})
    if $E'_{cur} > E_{cur\_max}$
        $E_{cur\_max} = E'_{cur}$
    endif
    if $E'_{cur} < E_{cur\_min}$
        $E_{cur\_min} = E'_{cur}$
    endif
    $E_{cur\_avg} = \dfrac{N_{live} * E_{cur\_avg} + E'_{cur}}{N_{live} + 1}$
    Increment N_{live}

Figure 5-2 Algorithm to cluster heterogeneous sensor network

Cluster_Set_Up ()

Calculate P_{range} by $P_{range} = \max(P, 1-P)$

if $E_{cur} < E_{cur\_avg}$

Calculate E_r by $E_r = \dfrac{E_{cur} - E_{cur\_avg}}{E_{cur\_avg} - E_{cur\_min}}$

else if $E_{cur} > E_{cur\_avg}$

Calculate E_r by $E_r = \dfrac{E_{cur} - E_{cur\_avg}}{E_{cur\_max} - E_{cur\_avg}}$

else

$E_r = O$

endif

Calculate ΔP by $\Delta P = P_{range} * E_r$

Calculate $T(n)$ by $T(n) = (P + \Delta P) * \dfrac{N}{N_{live}}$

Choose a random number r

if ($r < T(n)$) then

status=head

broadcast HEAD_EXPOSURE message

else

Receive HEAD_EXPOSURE messages from other sensor nodes

if (no HEAD_EXPOSURE message received) then

status=head

else

status=follower

send FOLLOWER_ACCEPTANCE message to nearest cluster head

endif

endif

Figure 5-3 Algorithm to cluster heterogeneous sensor network (contd.)

```
Steady_State ()
    if status=follower
        send self originated data to own cluster head
    else
        receive messages from own followers
        aggregate and compress the received messages with own message
        send to base station
    endif
```

Figure 5-4 Algorithm to cluster heterogeneous sensor network (contd.)

We have presented our clustering algorithm and its mathematical model in this chapter. Our algorithm performs better than LEACH and its variants, which we will proof by some simulation results in the following chapter.

6 Simulation Results

In Chapter 4, we have derived our mathematical model for LEACH based on solid reasoning. As a further validation, we compare the energy consumption rate behavior resulted from our mathematical model with that from simulation runs for a particular network setup. We also compare the performance of our new clustering algorithm, proposed in Chapter 5, with that of LEACH and one of its best variant. The performance is compared in terms of the number of live nodes, the First Node Dies, the Half of the Nodes Die, and the Last Node Dies. Simulation environment is discussed in the next section.

6.1 Network Settings and Simulation Parameters

A visual C++ program is developed for the simulation. We use a network setting as shown in Figure 6-1 in the simulation runs. The network setting compiles with the assumptions stated in Chapter 4 and is as follows:

- The dimension of sensor area is 200 X 200.
- Total number of heterogeneous sensor nodes in the network is 100.
- The sensor nodes are uniformly distributed over the sensor area.
- The base station is located at position (1500, 100).
- Nodes are heterogeneous in terms of their energy levels. Initial energies of the sensor nodes are uniformly distributed between 1.0Joule and 5.0Joule, which has been shown in Figure 6-2.

In the simulation runs the following parameters are used –

1. The amount of energy per bit to run sensor node circuitry, E_{elec} is 0.00000005;
2. The value of energy constant, ϵ_{amp}, for radio transmission, is 0.0000000001;
3. A sensor node generates 0 to 50 units data in an interval;
4. Each data unit contains 8 bits data;
5. The probability that a message successfully arrives at its destination is 90%.

6. The number of data units generated in each subinterval by a sensor node is normally distributed in [0, 50], with the value of mean equal to 25. We applied Box-Muller transformation [22] to achieve this normal distribution from the uniform distribution of the built-in *rand*() function in vc++.

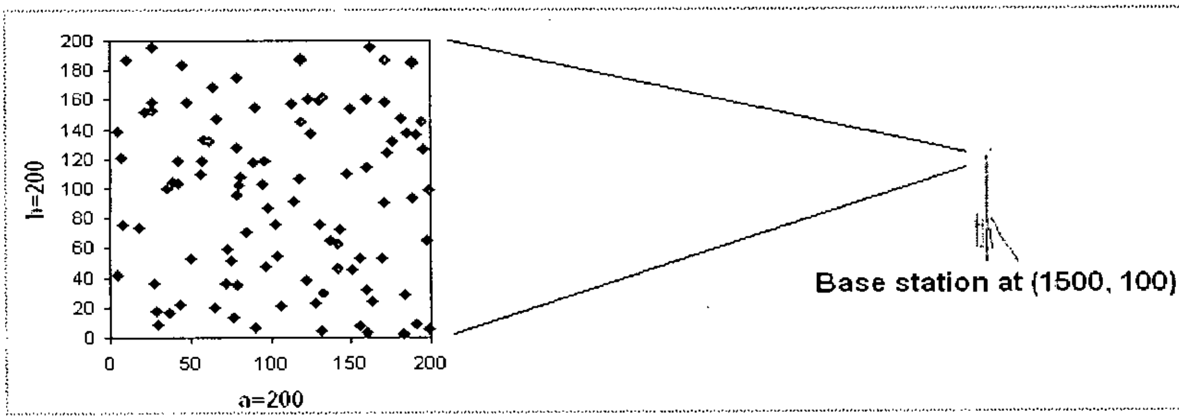


Figure 6-1 Network Setting: Uniformly distributed sensor nodes with a distant base station

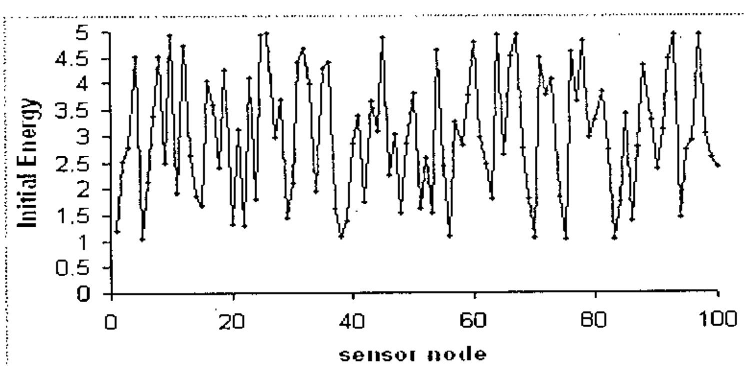


Figure 6-2 Initial Energy Distribution of the Sensor Nodes

6.2 Verification of the Mathematical Model

To verify the correctness of our mathematical model (Equation 27), we conducted simulation runs with the network settings and parameters described in Section 6.1. From the results of our simulation runs, we first plot the energy consumption rate versus the percentage of heads for a random LEACH node in Figure 6-3. According to the graph:

1. Energy consumption rate initially decreases very sharply with the increase of the percentage of cluster heads.

2. There is an optimal point for which energy consumption rate is the lowest. After this point the energy consumption rate increases with the increase of the percentage of cluster heads. In our simulation runs this optimal point is (0.057, 0.003433).

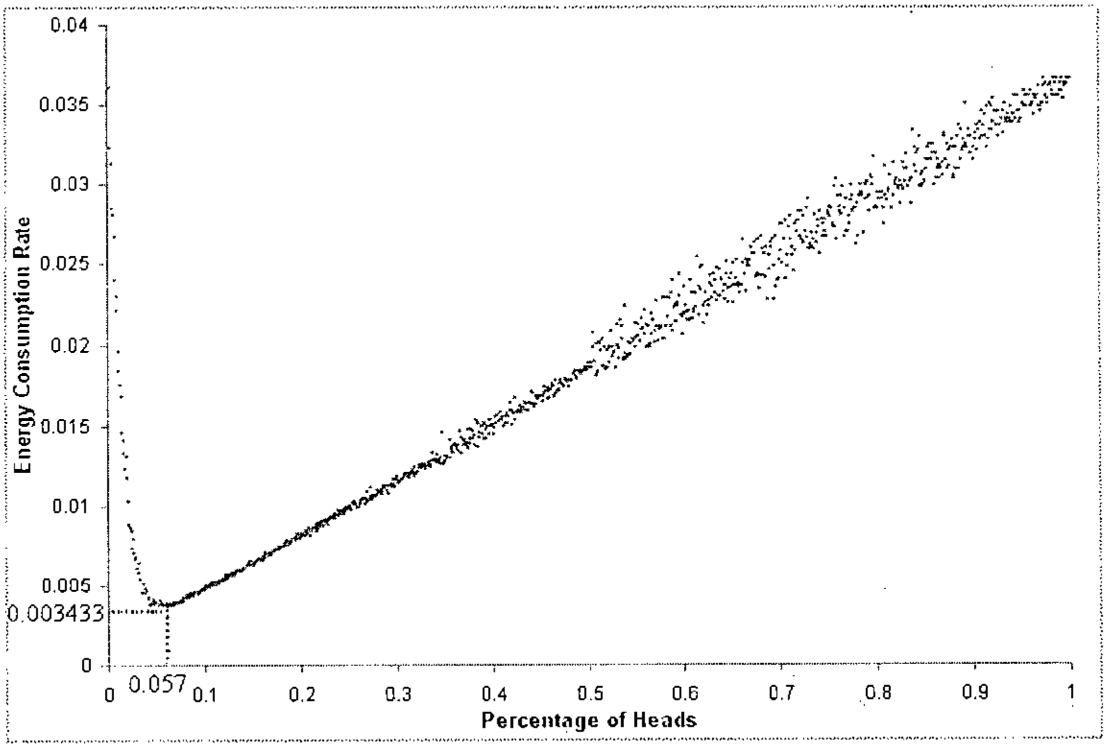


Figure 6-3 Energy Consumption Rate against Different Percentage of Heads for a Random LEACH node

Next, we plot the average energy consumption rate by all the sensor nodes in the network in Figure 6-4. This graph has the similar pattern that has been found in Figure 6-3. The behavior of energy consumption rate against the percentage of heads in Figure 6-3 and Figure 6-4 is due to following reasons:

1. The percentage of cluster heads P reflects the expected value of the threshold. When the value of P is close to 0, the probability of the random numbers picked by the sensor nodes in the network to be less than the threshold becomes very low. In this situation, most of the sensor nodes will frequently become one member cluster head. Therefore, the expected energy consumption rate becomes high. As the value of P increases, the probability of frequently becoming one member cluster head for the sensor nodes decreases, i.e., the probability of becoming follower increases and the expected energy consumption rate decreases. Though

the probability of becoming a one member cluster head decreases with the increase in P after a certain point this does not help that much to reduce the ultimate energy consumption rate.

2. After the said point the increase in the percentage of cluster heads increases the probability of the sensor nodes to become regular cluster head. This is superseding the energy savings by not becoming one member cluster head. For this reason, as the value of the percentage of cluster heads increases the ultimate energy consumption rate increases.

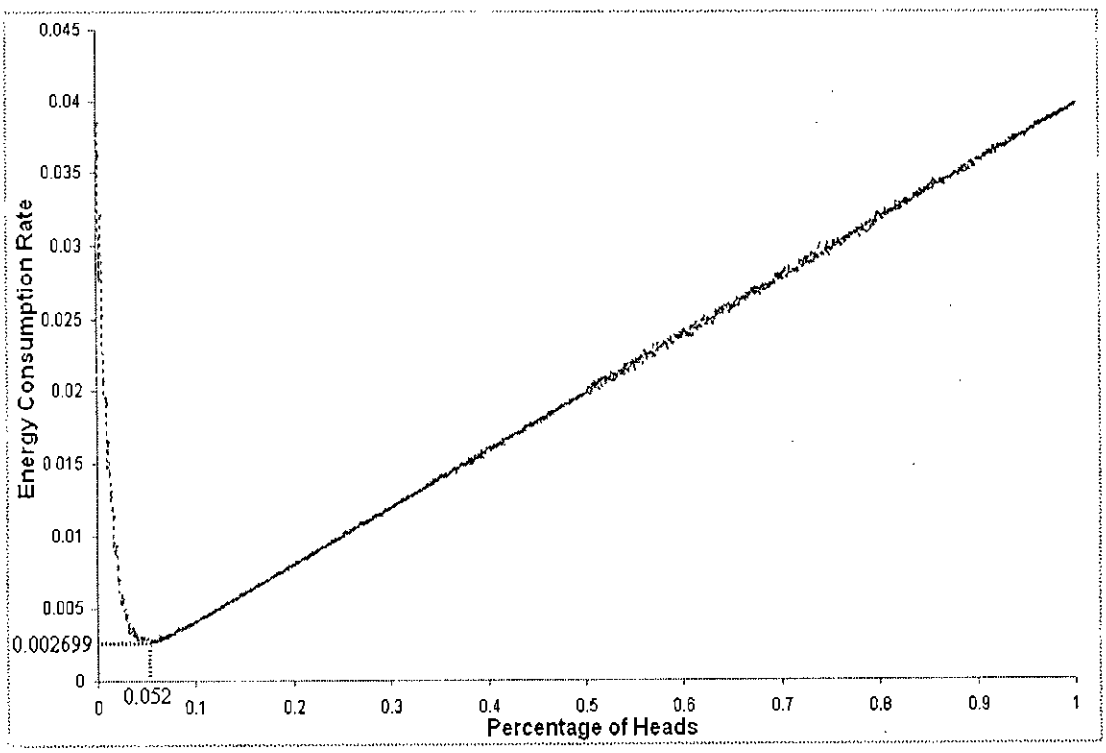


Figure 6-4 Average Energy Consumption Rate against Different Percentage of Heads over the Whole Network

We plot the energy consumption rate against different percentage of cluster heads P using Equation 27 of our mathematical model for the same network settings and parameters. The resultant graph is shown in Figure 6-5. The value of the probability of becoming cluster head of a sensor node at the start of any subinterval Φ_0 must not exceed 1 and the value of Φ_0 can be computed from by Equation 6. According to Equation 6, if the value of P exceeds 0.61 then the value of Φ_0 will exceed 1. In order to avoid this, we plot the graph against the percentage of cluster heads P up to 0.61.

The graph obtained from the mathematical model also shows the similar pattern that has been found in Figure 6-3 and Figure 6-4. Energy consumption rate also decreases very sharply at the beginning with the increase of the percentage of cluster heads. The optimal point (0.045, 0.003733) in Figure 6-5 is also almost equal to that of in Figure 6-4. And beyond this optimal point the energy consumption rate increases with the increase of the percentage of cluster heads in both graphs. This result is clearly validating the correctness of our mathematical model.

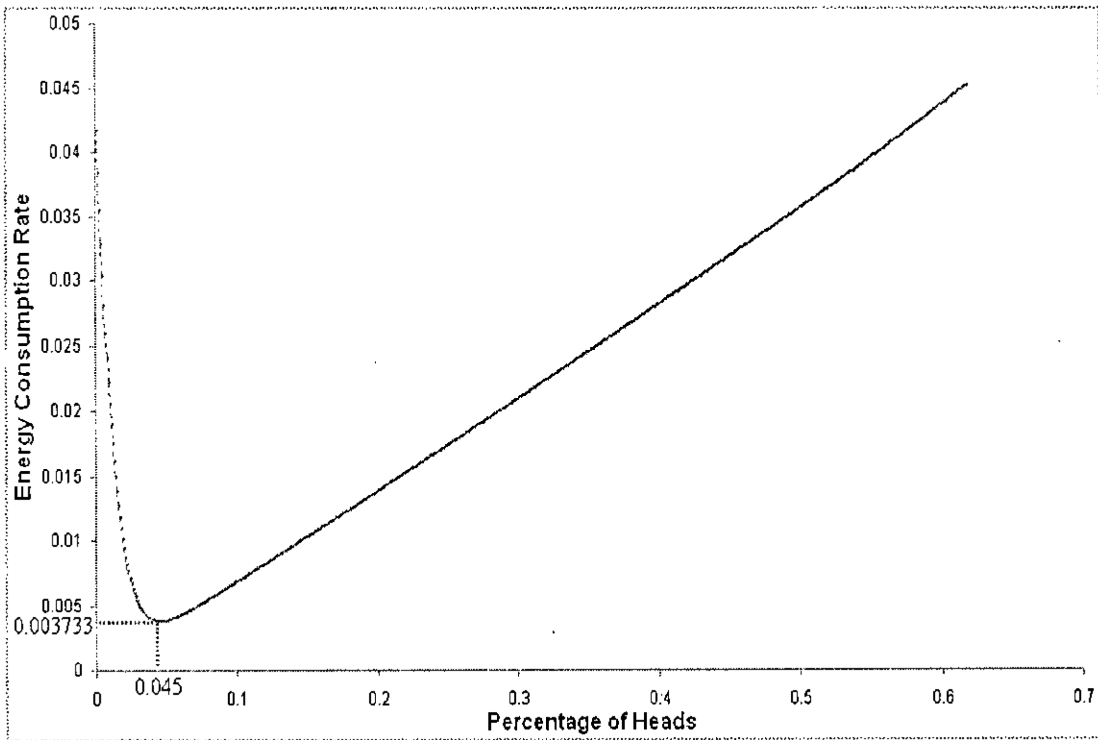


Figure 6-5 Energy Consumption Rate against Different Percentage of Heads according to the Mathematical Model of LEACH

6.3 Result from New Mathematical Model

We plot the graph of energy consumption rate against the percentage of cluster heads P using new mathematical model with the same network settings and parameters in Figure 6-6. This graph is also depicting the similar behavior that has been depicted in the graphs plotted from the simulation runs and from the previous mathematical model and shown in Figure 6-4 and Figure 6-5 respectively. The values of the optimal percentage of heads do not vary from Figure 6-5 to Figure 6-6. However, the energy consumption rate at this optimal point in the new algorithm is 0.003582 instead of 0.003733. Therefore, the percentage of improvement in the energy consumption rate at the optimal point in the new model is 4.05%.

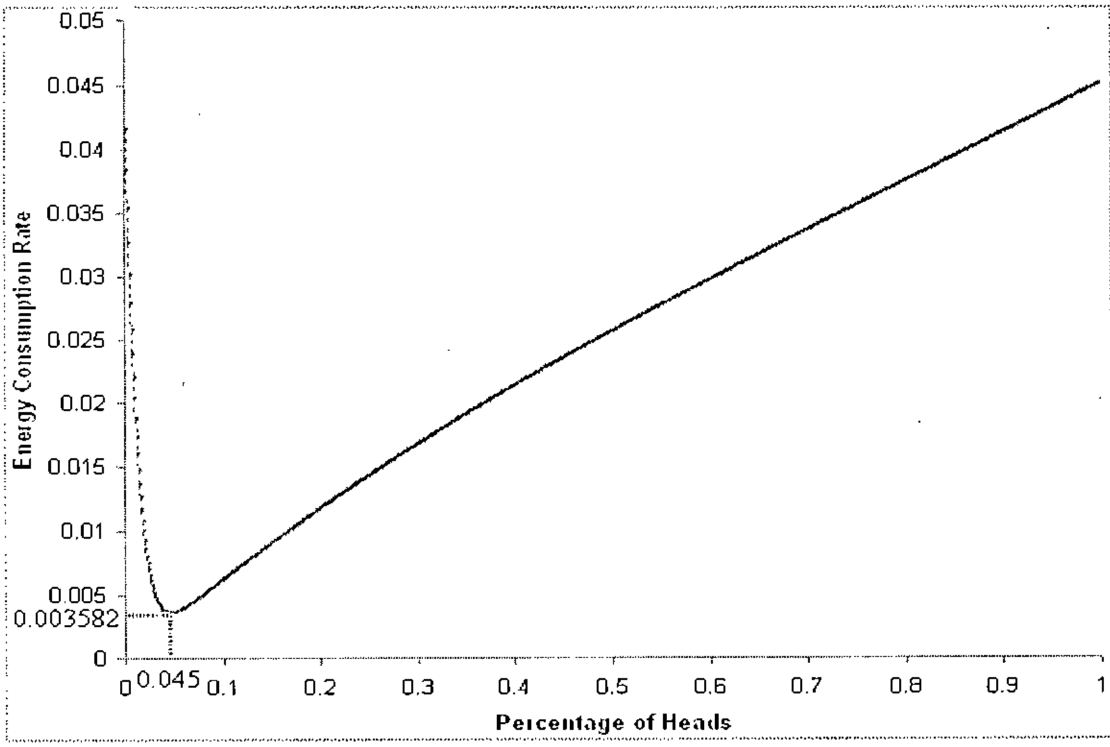


Figure 6-6 Energy Consumption Rate against Different Percentage of Heads according to the Mathematical Model incorporating Heuristic 1

6.4 Overall Comparison

We run our simulation for CHSN, LEACH, and a best LEACH variant with the network settings and parameters described in Section 6.1. The authors of Deterministic Cluster Head Selection [17] claimed that it improves the network stability period by 30% over LEACH whereas the authors of SEP [16] claimed that it does the improvement over LEACH by 26%. These two are the most improved LEACH variants claimed so far. For this reason, we take Deterministic Cluster Head Selection as the best LEACH variant instead of SEP in our performance comparison.

At first, we determine the optimal values of the percentage of heads *P* for each clustering technique using corresponding mathematical model. These values are 0.045, 0.063, 0.045 for LEACH, LEACH variant and CHSN respectively. Next, we run our simulations using corresponding optimal value of P for each clustering algorithm under comparison. We plot the number of live nodes against the number of interval passed in Figure 6-7. These values are taken from the average of ten simulation runs. According to the graph, CHSN performs better than LEACH and its variant.

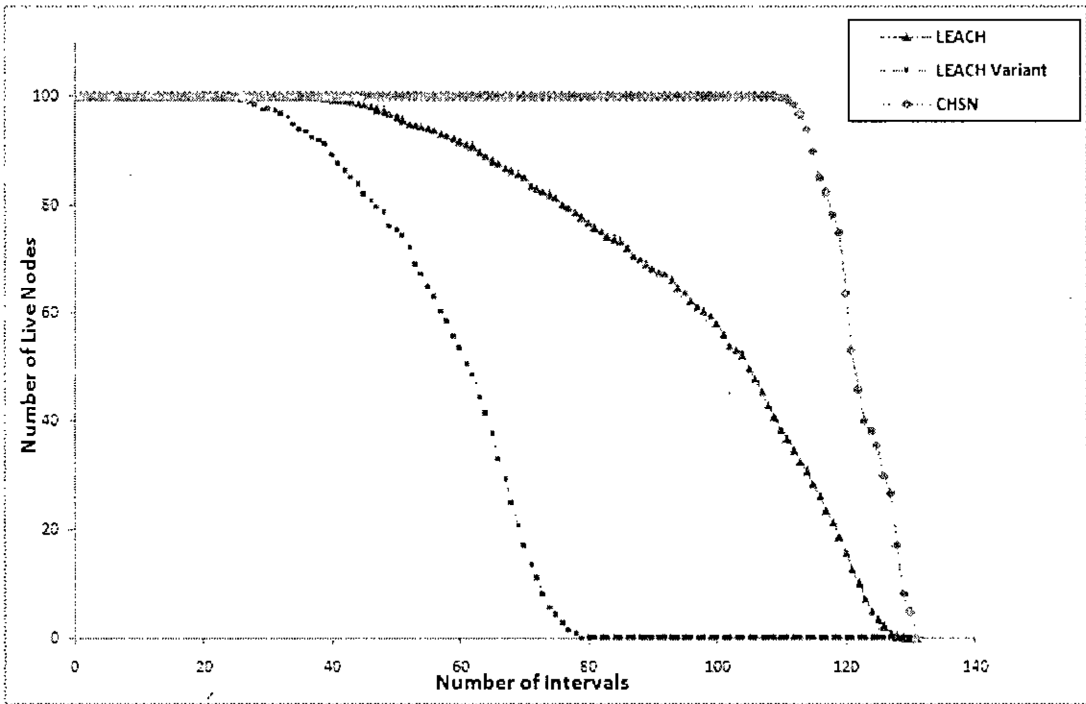


Figure 6-7 Number of Live sensor nodes in Different Intervals for LEACH, LEACH Variant, and CHSN

The probability of a sensor node to become a cluster head does not depend on its residual energy in LEACH. A sensor node with high residual energy may become a follower while a sensor node with low residual energy is becoming a cluster head. After being a cluster head with low residual energy a sensor node will die quicker. As a result, the first sensor node dies very quickly in LEACH. Due to the death of sensor nodes the number of live nodes in the network decreases as the time passes.

LEACH variant (Deterministic Cluster Head Selection) distributes the probability of becoming cluster head to the sensor nodes based on the ratio between the current residual energy and the initial maximum energy in the network. As the current residual energy of a sensor node decreases its probability of becoming regular cluster head decreases. It also reduces the value of the threshold. At the reduced threshold the probability of becoming one member cluster head increases. This in turn increases the energy consumption rate of the sensor nodes where energy is already scarce. As a result, the first sensor node dies in LEACH variant even quicker than that of LEACH.

CHSN distributes the probability of becoming cluster head to the sensor nodes based on their relative residual energies. In this approach, a sensor node with lower residual energy has lower probability of becoming cluster head and a sensor node with higher residual energy has higher probability of becoming cluster head. The energy consumption rate of a sensor node with higher residual energy is higher. However, the energy consumption rate of a sensor node with lower residual energy is lower. This proportionate distribution of energy consumption rate results in almost equal lifetime for all the sensor nodes in the network. Therefore, the first sensor node dies after longer duration in CHSN than that in LEACH. Most of the sensor nodes remain alive for a long time. However, once they start dying almost all of them die in a short time. For this reason, number of live nodes in the network falls vary sharply just after the first node dies in CHSN.

Now, we plot the standard deviations of the number of live nodes for LEACH, LEACH Variant, and CHSN. The resultant graph is shown in Figure 6-8. From the graph it is obvious that for a long duration of time from the deployment, the behavior of CHSN is more stable than that of both LEACH and LEACH variant. However, the behavior of CHSN becomes relatively unstable for a shorter period at the end of the network lifetime when nodes start dying, which is quite acceptable.

We also observed CHSN performance over LEACH with respect to three metrics as follows:

1. First Node Dies (FND): The time needed for the death of the first node. This metric is also called the network stability period.
2. Half of the Nodes Die (HND): The time needed for the death of the half of initially deployed live nodes.
3. Last Node Dies (LND): The time needed for the death of the last live node in the network. This metric is also called the network lifetime.

Figure 6-9(a), Figure 6-9 (b), Figure 6-9 (c) shows the average values for FND, HND, and LND over ten simulation runs for each clustering algorithm.

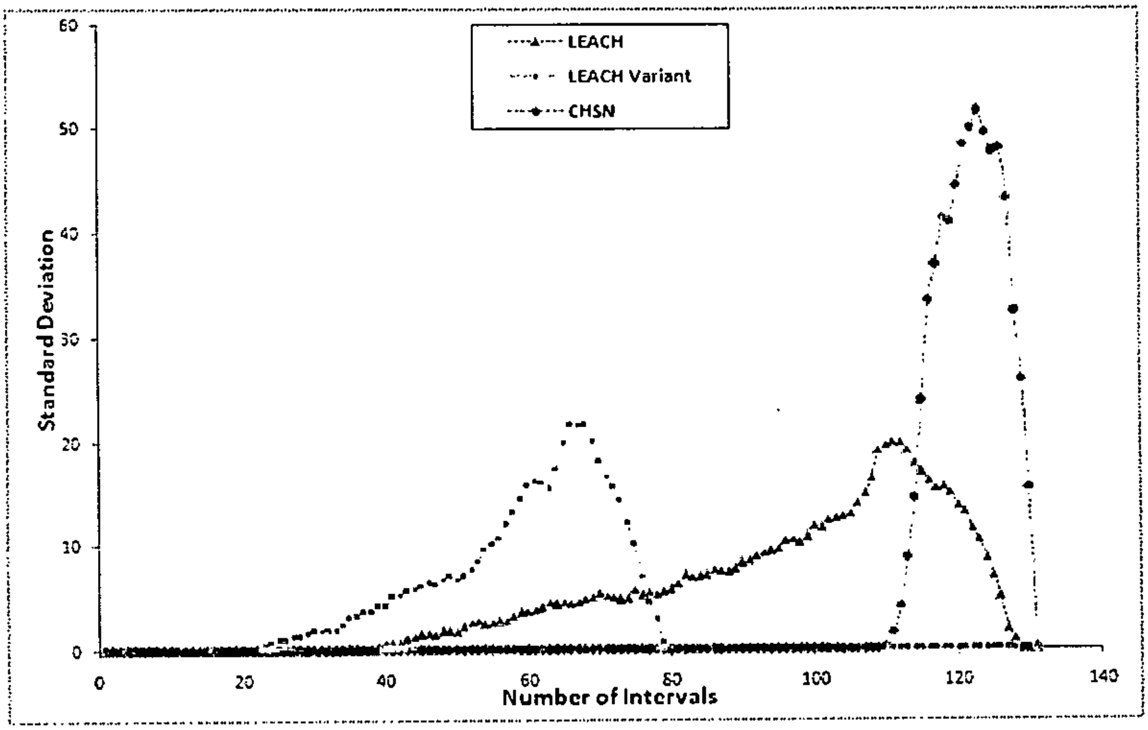
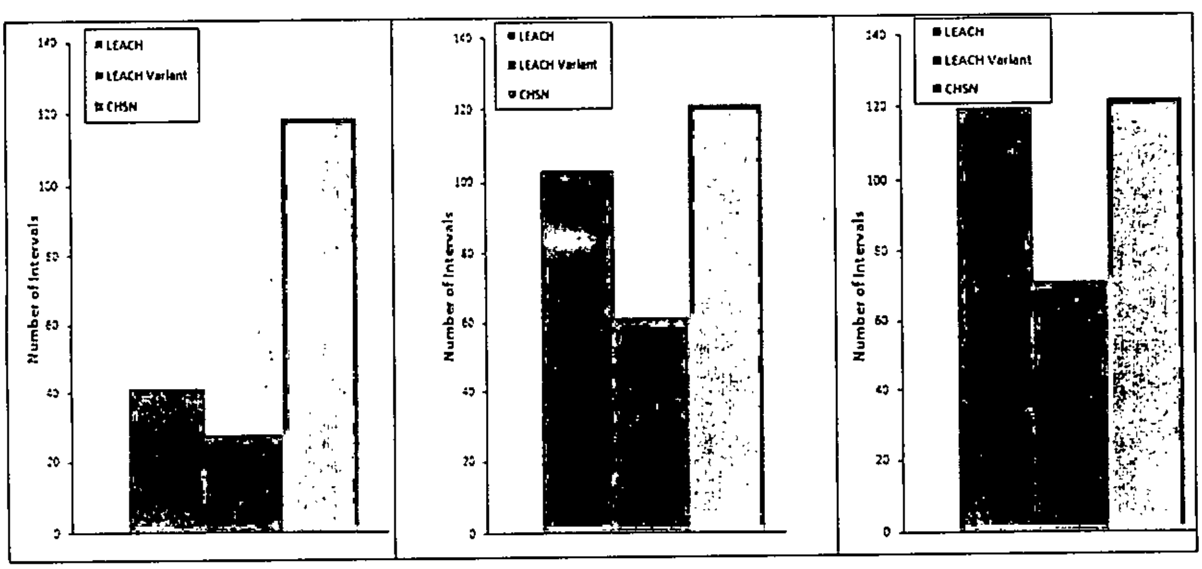


Figure 6-8 Standard deviations of the number of live sensor nodes in Different Intervals for LEACH, LEACH Variant, and CHSN



a) First Node Dies (FND) b) Half of the Nodes Die (HND) c) Last Node Dies (LND)

Figure 6-9 Comparison of First Node Dies (FND), Half of the Nodes Die (HND), and Last Node Dies (LND)

The performance improvements on these metrics by CHSN over LEACH and its variant are shown in Table 6-1.

Metric	Improvement with respect to LEACH	Improvement with respect to LEACH Variant
First Node Dies (FND)	188.56 %	326.62 %
Half of the Nodes Die (HND)	17.21 %	98.04 %
Last Node Dies (LND)	1.67 %	70.73 %

Table 6-1 Improvements of First Node Dies (FND), Half of the Nodes Die (HND), and Last Node Dies (LND) in CHSN

The standard deviations of FND, HND, and LND for LEACH, LEACH Variant, and CHSN found from 10 simulation runs are quite acceptable and shown in Table 6-2.

Metric	Standard Deviation		
	LEACH	LEACH Variant	CHSN
First Node Dies (FND)	3.414	2.348	5.103
Half of the Nodes Die (HND)	8.140	3.425	4.756
Last Node Dies (LND)	8.034	2.757	4.433

Table 6-2 Standard deviation of First Node Dies (FND), Half of the Nodes Die (HND), and Last Node Dies (LND) in CHSN

As CHSN ensures almost equal lifetime for each sensor node by distributing the energy consumption relative to the current residual energy, it takes a long time for the first sensor node to die. Therefore, the improvement on FND in CHSN is very significant over LEACH and its variant. For the same reason and for preserving the initial optimal number of cluster heads, CHSN has improvements on HND and LND over LEACH and its variant.

7 Conclusion and Future Works

In this thesis, we devised a mathematical model for LEACH protocol, a widely accepted clustering protocol for WSN. We have validated the correctness of our mathematical model by simulation results. We have proposed a new technique namely CHSN to cluster sensor network which considers the heterogeneity of sensor nodes in terms of residual energy levels to elongate both the network stability and the network lifetime. We have analyzed the performance of our clustering technique as well as the performance of other popular clustering techniques.

For the verification of the correctness of our mathematical model, we have simulated a WSN with a random setting. We have applied our mathematical model on the same network setting. We have plotted two graphs of energy consumption rate versus the percentage of heads, one for each of the above cases. The conformity between these two graphs ensures the correctness of the mathematical model.

We have also showed that we can find the optimal percentage of heads, for which the energy consumption rate would be the lowest from the graph of the mathematical model. Using this optimal value, we conducted the simulation runs to see the performance of our proposed clustering technique compared to other clustering technique. We found that our clustering technique performs better than others in terms of the number of live nodes, the first node dies, the half of the nodes die, and the last node dies.

In this thesis, we have considered heterogeneity only in terms of residual energy. However, the heterogeneity in a sensor network may arise from the difference in transmission and receiving range. We have not considered this type of heterogeneity in this thesis assuming all the sensor nodes can reach the base station directly. We preserved the initial optimal number of cluster heads from the beginning of the deployment to avoid the situation where it forces the sensor nodes to become one member cluster head. This situation might not arise from the beginning of the deployment. However, this situation may become severe after certain time. In this thesis, we have not considered that time. In our future works we will incorporate the heterogeneity of the sensor nodes in terms of receiving and transmission ranges. We will also consider the time from when the initial optimal number of cluster heads should be preserved in our future research work.

8 References

[1] *W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, Energy efficient communication protocol for wireless microsensor networks, In Proceedings of the 33rd Hawaii International Conference on System Science, vol. 8 , pp. 8020, January, 2000.*

[2] *J. Han and K. Micheline, Data Mining Concepts and Techniques, Morgan Kauffman, 2001.*

[3] *S. Basagni, Distributed clustering algorithm for ad-hoc networks, Proceedings of International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN), pp. 310, 1999.*

[4] *S. Banerjee and S. Khuller, A clustering scheme for hierarchical control in multi-hop wireless networks, Proceedings of IEEE INFOCOM, pp. 1028-1037, 2001.*

[5] *M. Gerla, T. J. Kwon, and G. Pei, On demand routing in large ad hoc wireless networks with passive clustering, Proceeding of WCNC, vol. 1, pp. 100-105, 2000.*

[6] *C. R. Lin and M. Gerla, Adaptive clustering for mobile wireless networks, Proceedings of IEEE J. Select. Areas Commun., vol. 15, pp. 1265-1275, 1997.*

[7] *J. Kamimura, N. Wakamiya, M. Murata, Energy-efficient clustering method for data gathering in sensor networks, Proceedings of Workshop on Broadband Advanced Sensor Networks, vol. 103, pp. 31-36, 2004.*

[8] *L. J. Jiunn, T. M-Hui, C. T-Chiang, H. Yueh-Min, Adaptive power aware clustering and multicasting protocol for mobile ad hoc networks, International Conference on Ubiquitous Intelligence and Computing, vol. 4159, pp. 331-340, September 6, 2006.*

[9] *S. Lindsey, C. S. Raghavendra, PEGASIS: Power-efficient gathering in sensor information systems, aerospace conference proceedings, vol. 3, pp. 3-1125 - 3-1130, 2002.*

[10] *L. Li, S. Dong, X. Wen, An energy efficient clustering routing algorithm for wireless sensor networks, The Journal of China Universities of Posts and Telecommunications, vol. 13, issue 3, pp. 71-75, September, 2006.*

[11] *Y. Sangho, H. Junyoung, C. Yookun, H. Jiman, PEACH: Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks, Computer communications , vol. 30, pp. 2842-2852, 2007.*

[12] *S. Ghiasi, A. Srivastava, X. Yang and M. Sarrafzadeh, Optimal energy aware clustering in sensor networks, SENSORS Journal, vol. 2, issue 7, pp. 258-269, 2002.*

[13] *H. Chan and A. Perrig, ACE: An emergent algorithm for highly uniform cluster formation, Proceedings of EWSN, pp. 154-171, 2004.*

[14] *O. Younis and S. Fahmy, Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach, Proceedings of IEEE INFOCOM, vol. 1, pp. 640, 2004.*

[15] *J. Y. Cheng, S. J. Ruan, R. G. Cheng, and T. T. Hsu, PADCP: power aware dynamic clustering protocol for wireless sensor network, International Conference on Wireless and Optical Communications Networks, April 13, pp. 6, 2006.*

[16] *G. Smaragdakis, I. Matta, A. Bestavros, SEP: A stable election protocol for clustered heterogeneous wireless sensor networks, Proceedings of 2nd Int. Workshop on Sensor and Actuator Network Protocols and Applications, 2004.*

[17] *M. Haase, D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head selection, IEEE MWCN, pp. 368-372, 2002.*

[18] *W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless micro sensor networks, IEEE Transactions on Wireless Communications, pp. 660–670, October 2002.*

[19] *C. Nam, H. Jeong, D. Shin, The adaptive cluster head selection in wireless sensor networks, IEEE International Workshop on Semantic Computing and Applications, pp.147-149, 2008.*

[20] *T. Murata and H. Ishibuchi, Performance evaluation of genetic algorithms for flowshop scheduling problems, Proc. 1st IEEE Conf. Evolutionary Computation, vol. 2, pp. 812–817, June 1994.*

[21] *W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, In Proceedings of the IEEE Transactions on Wireless Communications, vol. 1 , pp. 660-670, 2002.*

[22] *http://en.wikipedia.org/wiki/Box-Muller_transform*