AN EFFICIENT WEDGE MERGING SCHEME FOR ENERGY HOLE MITIGATION IN WIRELESS SENSOR NETWORKS

By

Nusrat Sharmin

MASTER OF SCIENCE

IN

INFORMATION AND COMMUNICATION TECHNOLOGY

INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

The thesis titled "**AN EFFICIENT WEDGE MERGING SCHEME FOR ENERGY HOLE MITIGATION IN WIRELESS SENSOR NETWORKS**" submitted by Nusrat Sharmin, Roll No. 0411312022, and Session April, 2011, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Information and Communication Technology on July 29, 2017.

BOARD OF EXAMINERS

1. **Dr. Mohammad Shah Alam** — **Chairman (Supervisor)**
   Associate Professor
   IICT, BUET, Dhaka

2. **Dr. Md. Saiful Islam** — **Member (Ex-Officio)**
   Professor and Director
   IICT, BUET, Dhaka

3. **Mohammad Imam Hasan Bin Asad** — **Member**
   Assistant Professor
   IICT, BUET, Dhaka

4. **Dr. Khawza Iftekhar Uddin Ahmed** — **Member (External)**
   Professor
   Department of EEE
   United International University, Dhaka

Declaration

It is hereby declared that this thesis and any part of it has not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidate

Nusrat Sharmin
0411312022
IICT, BUET

Dedication

THIS THESIS IS DEDICATED

TO

MY FAMILY

Contents

List of Tables

List of Figures

List of Abbreviations

WSN	Wireless Sensor Network
WEMER	WEdge MERging
BS	Base Station
HN	Head Node
CH	Cluster Head
FDMA	Frequency Division Multiple Access
TDMA	Time Division Multiple Access
RE	Residual Energy
DT	Distance Threshold
LL	Long Link
CT	Cooperative Transmission
PEGASIS	Power Efficient Gathering in Sensor Information Systems
LEACH	Low Energy Adaptive Clustering Hierarchy
CCS	Concentric Clustering Scheme
CBCCP	Chain Based Cluster Cooperative Protocol
MCDA	Multilayer Cluster Designing Algorithm
CCO	Cluster Coordinator
ADRP	Adaptive Decentralized Re-clustering Protocol
CDC	Cooperative Duty Cycle
EEPB	Energy Efficient PEGASIS-based Protocol
IEEPB	Improved Energy Efficient PEGASIS-based Protocol
EAPHRN	Energy-Aware PEGASIS-based Hierarchal Routing Protocol for Wireless Sensor N

List of symbols

E_{elec}	unit energy dissipation for transmitter electronics or receiver electronics
ϵ_{fs}	amplifier energy in the free space model
ϵ_{mp}	amplifier energy in the multipath model
d_0	threshold
$D_{ito}BS$	Distance with Base Station
$D_{ito}HN$	Distance with Head Node

Acknowledgment

All praises are for the almighty Allah for giving me the strength, without which I could not afford to attempt this research work.

I would like to express my sincere and heartiest gratitude to my honorable thesis supervisor Dr. Mohammad Shah Alam, Associate Professor, Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka for his continuous motivation, guidance and keen encouragement which helped me throughout the time of my research work. Nothing is comparable to his keen advice and the freedom he provided for me in research. I am grateful to him for his cooperation throughout my thesis work.

I would like to thank all the members of the board of examiners for their precious time in understanding my work and their insightful comments. I would like to thank to all of my friends and colleagues for their cooperation. Last but not least, I am grateful to my parents for their continuous supports and cooperation.

Abstract

In multi-hop Wireless Sensor Network (WSN), nodes closer to Base Station (BS) need to relay traffic from other nodes of the network which makes their energy depleted very fast and may cause an energy hole in the network. Moreover, energy consumption among nodes is not balanced due to non-uniform distribution of nodes and it causes some node to drain out their energy faster than other nodes in the network. In WSN, this energy hole problem plays a key factor to reduce the lifetime of the network as data cannot be sent from other sensor nodes to the BS although the residual energy in the network remains high. Some techniques to alleviate the energy hole problem include using a mobile sink, extending transmission range, and deploying redundant of the node near to the Sink. But the load of each node is still unbalanced and high energy consumption node die early. In order to reduce energy holes in the network, a new scheme WEMER based on WEdge MERging, is proposed here. In WEMER, the whole network is divided into many small equiangular wedges. When the residual energy of all the nodes within a wedge becomes less than 40% of their initial energy, thereby increasing the probability of energy holes creation, this wedge merges with the neighboring wedge to thwart the energy holes formation. Simulation results show that proposed scheme balances the energy consumption among nodes and achieves much longer lifetime than the contemporary schemes like Power Efficient Gathering in Sensor Information Systems (PEGASIS) , Concentric Clustering Scheme (CCS) and Multilayer Cluster Designing Algorithm (MCDA).

Chapter 1

Introduction

A wireless sensor network (WSN) is composed of a large number of tiny, low cost, low-powered sensor nodes [1]. These tiny sensor nodes have small internal memory and low computational ability with limited battery life time. Sensor nodes are capable of monitoring, sensing, aggregation and transmission of data to the Base Station (BS) or Sink node (central gathering point). WSNs are used in many communication applications including security, medical, surveillance, weather monitoring, traffic monitoring [2]. Sensor nodes are able to measure various parameters of the environment and transmit collected data to the Base Station directly or through multi-hop communication. The base station is connected to the wired world where the data can be collected in large databases for future use. Figure 1.1 shows a typical sensor network example. The WSN may comprise of hundreds or even more number of nodes, which provides reliable monitoring of any applications. The sensed data are transmitted to the base station directly or by multi hop fashion.

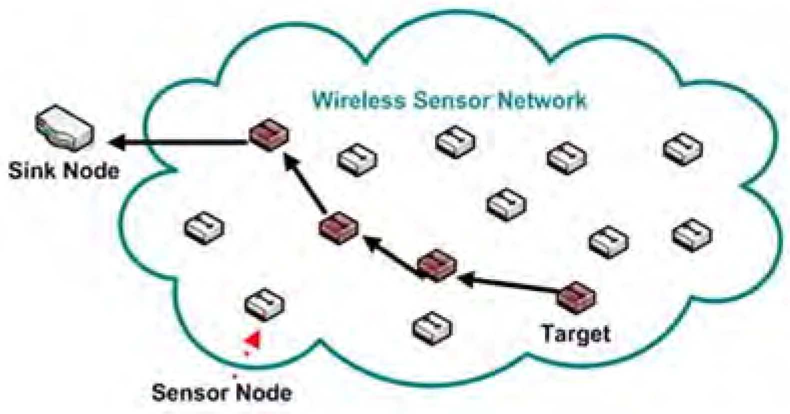


Figure 1.1: A typical wireless sensor network [3]

1.1 Challenges in Wireless Sensor Network

Before formation of the sensor network and deployment of sensor nodes, network is needed to be more scalable and efficient. Some important challenges that the wireless sensor networks should overcome are described below.

Energy Efficiency: The first is the energy efficiency. The sensor nodes are battery powered and it is often very difficult to change or recharge batteries for these sensor nodes; so energy consumption should be managed wisely in order to extend the network lifetime. Many researchers are focusing their work on energy efficient network. Hence, if the batteries are exhausted, the sensors may fail and might not function. So, efficient routing may overcome and extend the network lifetime.

Latency: The second is latency. Latency requirement basically depends on the application. In the sensor network applications, the detected events must be reported to the base station in real time so that the appropriate action could be taken immediately. The routing protocols and network topology will ensure the delivery of the data with minimum delay.

Fault Tolerance: Node failure and change of topology of network is very common in case of WSN. Researchers are working on how to make network robust and reliable even in case of node failures and topology changes.

Throughput: The required number of successful packet transmission of a given node per unit time is determined as throughput. Throughput requirement also varies with different applications.

Fairness: In many sensor network applications when bandwidth is limited, it is necessary to make sure that the base station receives information from all sensor nodes fairly.

Scalability and Adaptability: Some nodes may stop functioning due to battery drain or link error or any other environmental problems, since WSN protocol should be adaptable to changes in network size, density of node and topology.

Security: Data is sent to the end users by getting direct access to the messages present in the sensors through internet services. Hence, there is a need to prevent access to the data from unauthorized parties or from any malicious actions.

1.2 Energy Hole Problem

Nodes deployment is the first step in establishing sensor network. Sensor nodes are battery powered and randomly deployed in target area. The major challenges of the sensor nodes are battery power limitations, processing power constraints, duplicate data gathering and limited memory of the network. Optimizing the energy consumption is one of the major tasks in WSNs to prolong the network lifetime. To address this issue, much work has been done in this area during the last few years. If the sensor nodes are deployed uniformly, nodes near the sink send their own data as well as the date collected by other nodes away from the sink in multi-hop scenario. In this case, the sensor nodes near the sink consume more energy and die out quickly [5–7]. Figure 1.2 is showing this scenario, nodes near sink is deployed their energy than other nodes in the network. On the other hand if multi-hop is not used and all nodes transmit their data directly to the BS, node farthest from the BS die much faster than the nodes that are closer to the BS because they need more transmission power to transmit their data to the BS [8]. As a result, the sensor network will disconnect having sufficient energy left unused. This causes a significant decrease in the network lifetime. Energy holes create a partition in the network in such a way that it cannot make full connectivity in the network. Various techniques have been proposed to address the Energy Hole Problem (EHP) [9]. Some techniques to alleviate the energy hole problem includes Adjustable transmission range, sink or node mobility and non-uniform sensor distribution mainly deploying redundant

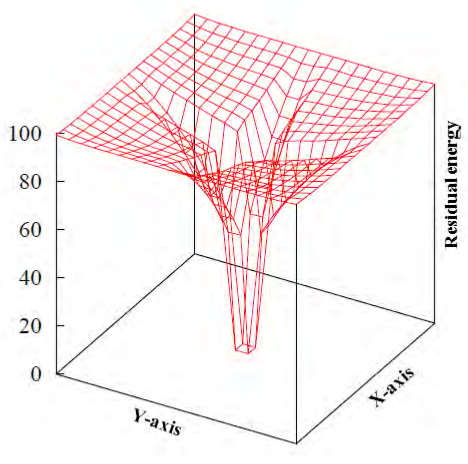


Figure 1.2: Node residual energy [4]

of the nodes near to the sink [10]. The aim of energy hole avoidance is to delay or bypass the formation of energy hole to maximize the network lifetime.

1.3 Techniques for solving energy hole problem

The energy-hole has the potential to drastically reduce the useful lifespan of sensor networks. So maximizing the effective network lifetime is equivalent to avoiding the energy-hole. Energy hole problem plays a key factor as data cannot be sent from other sensor nodes to the BS although the Residual Energy (RE) in the network remains high [6]. So the energy hole should be prevented to the largest extent possible. network lifetime can be increased by increasing initial energy, decreasing individual energy consumption, transmission and reception energy. Figure 1.3 shows method that are used for increasing network lifetime

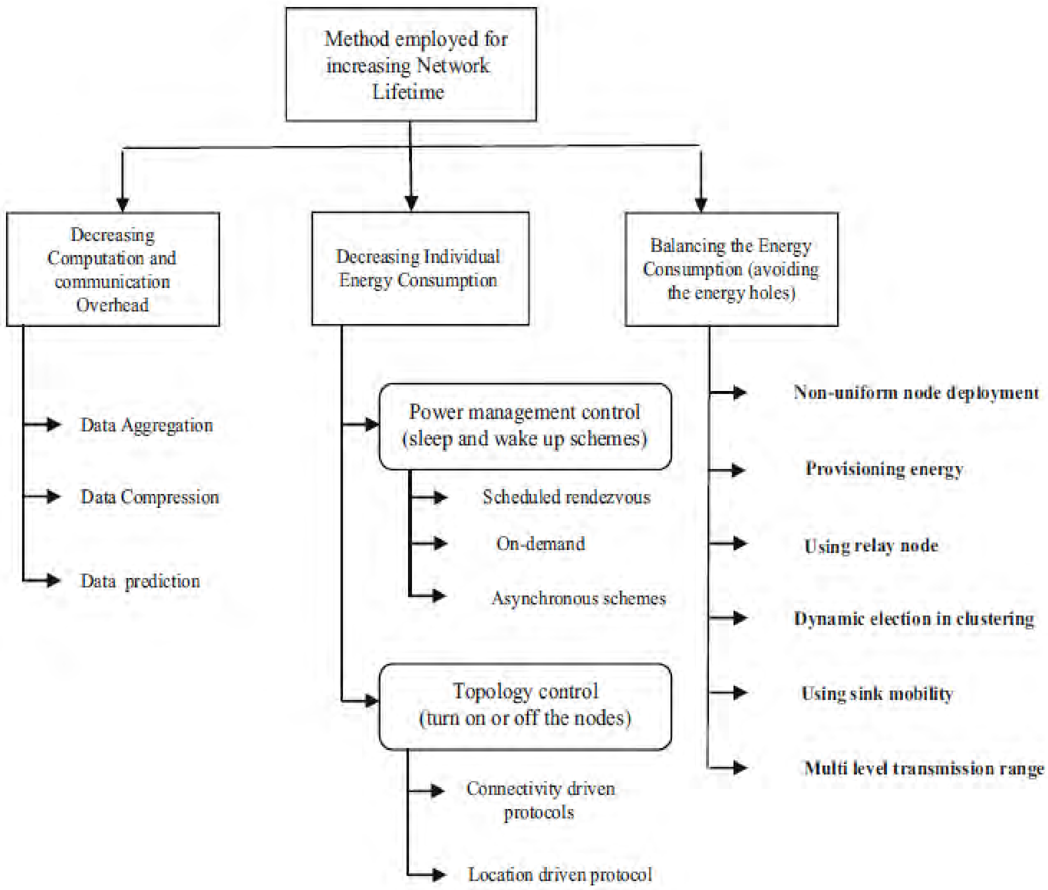


Figure 1.3: Method employed for increasing network lifetime [10]

1.3.1 Chain based approach

Chain based routing is a popular technique to reduce energy consumption of nodes in WSN [11]. Energy-Aware PEGASIS based Hierarchal Routing Protocol for Wireless Sensor Networks (EAPHRN) [12] uses Distance Threshold (DT) to form a chain. At first each sensor node compute Local DT (LDT) which is the average distance between the node and n closest node where n is a constant and determined by total nodes in the network. After computing LDT, every node sends LDT value to BS. Now BS calculates DT which is the average value of LDT and sends this DT value to every node of the network to form a chain from the farthest node of the BS. All nodes pick its next connected node within DT distance. Here defining n is complex because it depends on number of all nodes in the network, if n is low then DT value will be low and there could be a situation where no node will be in DT range. Also Delay is not considered here. The authors in [13] proposed IEEPB (an Improved Energy-Efficient PEGASIS-based protocol) as an improvement of EEPB (energy-efficient PEGASIS-based protocol) [14] protocol. In order to solve the Long Link (LL) problem of EEPB residual energy of node and distance of node from BS both are used as a weighting method to select the leader of the IEEPB. $W1$ and $W2$ two coefficients of weight factors are introduced and their values determined according to the system requirement. This protocol can solve the LL problem of EEPB but it is also time consuming as the data need to send through long chain.

1.3.2 Duty cycle based approach

Duty cycling [15–18] is most prominent technique to turnoff transceiver periodically to save energy. Authors in Cooperative Duty Cycle (CDC) – MAC [19] combine Cooperative Transmission (CT) with duty cycling to reduce energy hole but they spend more time on synchronization. In synchronous duty cycling approach clock synchronization is needed which increases energy consumption. To overcome this problem, asynchronous duty cycle approach is introduced. Asynchronous protocol works with two modes: sender initiated mode and receiver initiated mode. Sender initiated mac protocol has been introduced where sender takes initiative to seek attention of receiver by sending a long preamble [20–22]. It has advantages over synchronous MAC because in sender initiated protocol sensor nodes don't need to create any schedule by using clock synchronization. But long preamble causes extra overhead and high probability of collision. Then receiver initiated protocol came into focus [23–25]. Unlike sender initiated protocol, here receiver takes initiative by sending beacon to seek sender attention. Receiver initiated protocol reduces extra overhead over sender initiated protocol. Duo-MAC [26] usually runs in a low duty-cycling mode and behaves as an energy-efficient MAC but whereas when an event is detected, a node enters in a high duty-cycling mode and behaves as a delay-efficient MAC to forward real-time traffic.

1.3.3 Clustering approach

Clustering is a popular approach to save energy and prolong the network lifetime [27–30]. For large scale network, clustering is a popular approach. In this approach every member of cluster send their data to cluster head to transmit it to base station. Adaptive Decentralized Re-clustering Protocol (ADRP) selects Cluster Head (CH) and set of next heads for upcoming rounds based on residual energy of each node and average energy of cluster [31] but no new node can be added until next initial phase and set of cluster will not be same if any next head node gets died. In [32] authors restrict the number of CH advertisement with optimal number of CH count during CH selection time to save energy. Clustering is always an efficient technique to solve energy hole problem [33–38]. Chain Based Cluster Cooperative Protocol (CBCCP) [39] starts its processing by dividing the network into some subareas (clusters). Each subarea has one CH, varied number of Cluster Coordinator (CCO) and defined number of relay node to distribute the load. Number of CCO depends on the number of cluster in which all CCO are located. Every cluster need to maintain that it have one CCO for every cluster lying below to it. As they select the CH and CCO randomly and do not consider the RE or distance to choose them than there is a chance that node with low RE will be selected as CH or CCO repeatedly whereas other high RE node is not.

1.3.4 Adjustment of transmission range

By controlling transmission range energy hole problem can be solved. There are many studies that used transmission range control strategies to reduce energy consumption of nodes [40–45]. In [42] a sensor node determines its possible next hop nodes using a controlled region selection strategy and a node whose residual energy is maximum is chosen as the next hop. This avoids repeated and random selection of a node as next hop node which occurs in normal fixed transmission range scheme. In order to uniform energy usage a distinct set of hop sizes for sensors is introduced in [43].

1.4 Motivation

WSNs are usually deployed to monitor one or more phenomena and to collect useful data in an unattended environment such as battlefield monitoring, disaster management, industrial process control, habitat monitoring etc. In order to cover the whole area of interest, the size of the network may grow large and become a multi-hop network. Therefore, network energy must be conserved to fulfill network lifetime requirement of the application. Sensors' battery is the only source of power to keep the network alive. When sensor nodes use multi-hop transmission, the nodes that are near to the sink have more traffic to forward that makes their energy depleted very and get died early. So if more nodes which are close to the sink are involved to forward data than the burden of single node will decrease and lifetime of network will increase. During multi-hop routing process, how to choose the next hop node is a critical issue. Depending on the purpose of various applications, a node might choose its next hop node based on criteria like maximal residual energy, largest degree, shortest path or other routing strategies. In order to reduce delay and to save energy shortest path routing is preferred. In order to maximize network lifetime in WSN, exploiting an efficient strategies that will balance energy consumption among sensor nodes was the motivation behind this research.

1.5 Objective

The core objective of this thesis is to design a new scheme to reduce the energy hole problem for wireless sensor networks. During the process of designing this proposed protocol the following are the objectives of this thesis:

1. To develop an algorithm for merging wedges with the objective of eliminating or reducing energy holes as well as to maximize network lifetime.

2. To develop a simulation model in order to characterize the energy hole mitigation mechanism of the proposed wedge merging scheme.

3. To compare the performance of the proposed technique with the existing schemes in terms of network lifetime, energy cost and end to end delay.

1.6 Organization of Thesis

This thesis consists of five chapters. Brief description of its different chapter is as follows:

Chapter one introduces energy hole problem of wireless sensor network. Related researches regarding energy hole problem are presented in this chapter.

Chapter two presents literature review of some energy aware protocol. Limitations of those protocols are also discussed.

Chapter three presents proposed work which is wedge merging scheme with network model to eliminate energy hole problem in WSN. Theoretical analysis of technique is also describe here.

Chapter four illustrates simulation results and performance evaluation of the proposed method compared with three popular method.

Chapter five concludes the thesis and recommendations for future work is also available in this chapter.

1.7 Summary

This chapter introduce a very brief introduction of wireless sensor networks. A detail analysis of energy hole creation and some techniques that are used to solve energy hole are included here. Motivation and objective of the research work are also discussed to get a clear overview of the research work. Finally, organization of the thesis is mentioned in this chapter.

Chapter 2

Background and Related Work

2.1 Literature Review

The main task of sensor nodes is to monitor physical or environmental conditions to cooperatively pass their data through the network to a Base Station(BS) where the data can be observed and analyzed. Optimizing the energy consumption is one of the major tasks in WSNs to prolong the network lifetime. In recent years there are so many studies that had been proposed to reduce energy consumption of nodes. Some techniques are given below:

2.2 Power Efficient GAthering in Sensor Information Systems (PEGASIS)

Two most famous routing protocols that lead the improvement in energy efficiency nowadays are Low-Energy Adaptive Clustering Hierarchy(LEACH) [46] and PEGASIS [47]. LEACH is one of the first cluster-based routing protocols for WSNs. In order to reduce energy consumption the nodes organize themselves into clusters, where one node from every cluster acts as the cluster head to send data to base station. Sensor node is selected as a cluster heads by rotation, so that energy consumption in communicating with the base station is spread to all sensor nodes in the network. However, the drawback of LEACH is that selection of cluster head is random and it cannot be deployed in networks spread over large distances as LEACH uses single-hop communication. Each node delivers the sensing data to the nearest neighbor node through the chain. One of sensor nodes acts as the head node and delivers sensing data to the base station. The head node is selected by depending on round to consume the energy evenly in the wireless sensor networks The PEGASIS

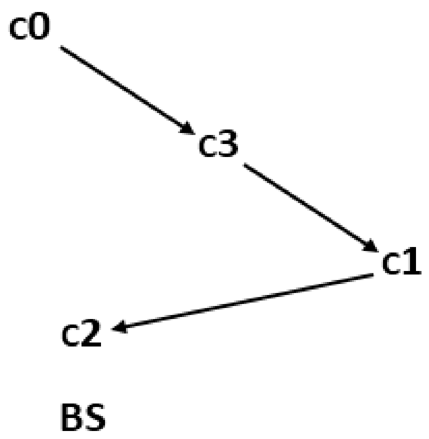


Figure 2.1: Chain construction using the greedy algorithm

protocol runs as follows:

2.2.1 Chain Construction

PEGASIS used the same radio model that was used in LEACH protocol which is first order radio model. Any node can receive data from only one node. Sensor nodes are distributed randomly in the network. The PEGASIS protocol performs two steps to construct a chain:

First step PEGASIS used greedy algorithm to construct a chain. Chain construction procedure will be started from the farthest node from the base station. Figure 2.1 shows the node C0 located farthest from the base station is starting point and node C0 connecting to node C3, node C3 connecting to node C1, and node C1 connecting to node C2 in that order. This step is continued until all nodes are connected with chain. Signal strength is used to measure the distance with neighbor nodes and each node can adjust its signal strength so that only one node that is connected with it can hear its message. The chain is reconstructed when a node dies to bypass the dead node.

Second step After chain construction procedure base station broadcast information of the chain to sensor nodes in second step.

2.2.2 Leader Node Selection

Only one node from the chain will be selected for transmit data to the base station, PEGASIS named this node as leader node. In every round a new leader node is selected based on round. To consume the energy evenly in the wireless sensor networks the leader node is selected by rounds. odes are randomly deployed, so to select leader node randomly PEGASIS used formula round i mod N (N represents node number) to transmit data to the base station in round i. The leader in each round will be at a random position on the chain. This random selection of leader node helps PEGASIS to consume energy evenly. This random selection make random position node die in the network.

2.2.3 Data transmission

After the chain construction procedure leader node start a token passing approach to start the data transmission. Figure 2.2 is showing token passing approach of PEGASIS. The PEGASIS protocol uses a token which contains a small message to start data transmission. In figure 2.2 c2 is the head node, that means c2 will send all nodes data to the base station. Node c2 will pass a token along the chain to node c0 to send its data. Node c0 will send its data to node c1 and after getting data from c0, c1 will fuse its data with c0 and transmit these data to node c2. After node c2 receives the data it will pass a token to node c4 and c4 will send the data toward node c2 in the same way.

In sum, PEGASIS protocol construct a chain using greedy algorithm and each node collect and send its data to its neighbor node. Each node selected as a leader node by turns in PEGASIS so die node will be throughout the network. Figure 2.3 shows the data gathering of PEGASIS protocol.

However PEGASIS has some limitations, these are given below:

1) One chain is form for whole network which creates long chain and long chain makes delay of communication.

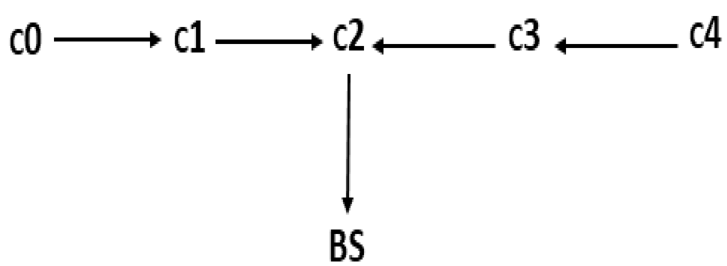


Figure 2.2: Token passing approach of PEGASIS [47]

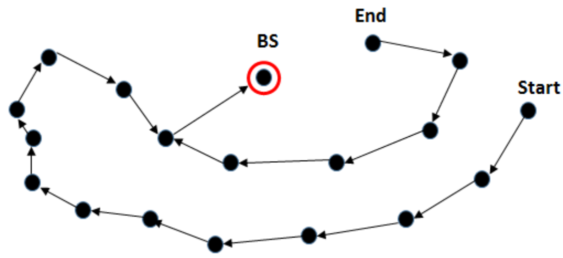


Figure 2.3: Data gathering of PEGASIS

2) Only leader node can transmit data to the base station. So leader node is heavily loaded.

3) When PEGASIS protocol selects leader node, energy or distance is not considered.

4) Redundant transmission of data as only one node is selected as leader node.

5) In PEGASIS, it is essential to have a complete view of the topology at each and every node for chain construction.

5) Greedy algorithm create long link problem in large network, that makes energy consumption of node is high.

2.3 Concentric Clustering Scheme (CCS)

CCS [48] is an enhanced of PEGASIS protocol. The main improvement of CCS is that it consider base station position when divide the network into concentric circles. The term concentric clustering means that the shape of a cluster is concentric circles when cluster is formed. CCS can improve the delay problem of PEGASIS. CCS perform in following way:

2.3.1 Level Assignment

Every sensor network get its level from base station. By adjusting transmission power, base station divides the network into level like concentric circle. Figure 2.4 is showing level assignment of CCS. Level size depend on base stations setting value. Number of level may vary according to the number of nodes, density of the sensor network and position of the base station.

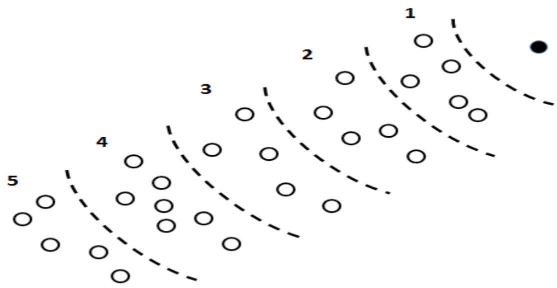


Figure 2.4: Level assignment of CCS

2.3.2 Chain construction for every level

In CCS for every level a chain is constructed by using greedy algorithm. Chain construction will be started from the node which farthest from the base station. Every level will identify a node which is farthest from the base station to start the chain construction. Figure 2.5 shows constructed chain of CCS.

2.3.3 Head node selection of every level

One of the nodes at each level is selected as a head node. If there are N nodes in a level then for ith round i mod N th node is selected as a head node. After the head node selection at each level it informs the head nodes of one lower level and one higher level.

2.3.4 Data transmission

Data transmission procedure of CCS is as like as PEGASIS. All node in each level transmit the data along the chain to its nearest neighbor. The node receives the data fuse it with its data and send this to its neighbor node. The head node of each level transmit the data to the lower head node. Head node of each level is

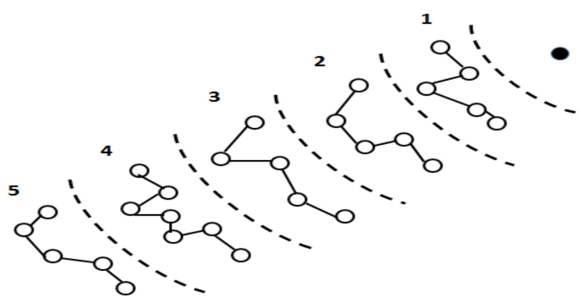


Figure 2.5: Chain construction of CCS

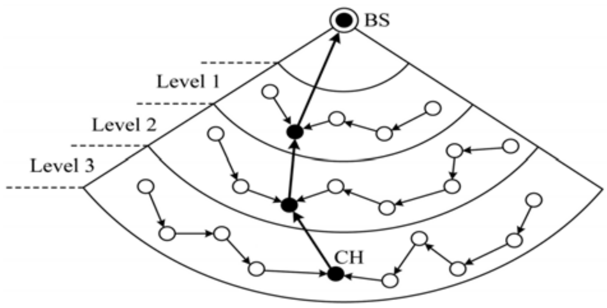


Figure 2.6: Data transmission [48]

responsible for receiving data from its level node and upper head node. Head node is also responsible for transmitting all data to the lower head node. When all data is received by level 1 head node then it will send its own and received data to the base station. Figure 2.6 shows data transmission of CCS.

Limitations of CCS are given below:

1) There is unbalanced node distribution at each level, which will cause the levels with a small number of nodes to deplete their energy first.

2) For every level a chain is form, thats make a long chain for level and long chain makes transmission delay.

3) During head node selection time energy and distance is not considered that may cause unbalanced energy consumption.

2.4 Multilayer Cluster Designing Algorithm (MCDA)

MCDA [49] is a combination of Distributed Cluster Designing (DCD) and Centralized Cluster Designing Approach (CCD). MCDA comprises of three steps: self organizing, flat layer design and clustered layer design.

2.4.1 Self-organizing

When a node switching on their active mode, they broadcast a beacon message to give their existence. The neighboring node receives the beacon message set up their neighbor table and calculate the link quality. MCDA sets up neighbor table at each deployed node and place the neighbor count in this table. Figure 2.7 shows self-organizing of MCDA.

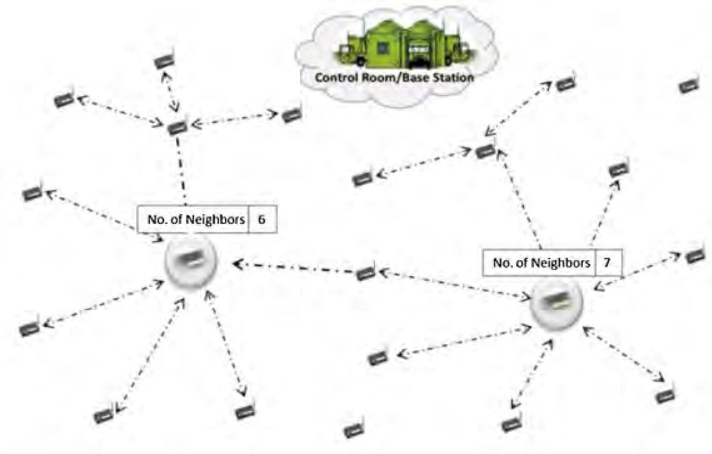


Figure 2.7: Self-organizing [49]

2.4.2 Flat Layer Design

Nodes which are close to the base station need to relay traffic that makes deplete their energy very fast. This close node is called as neighbor node of base station. The nodes up to one hop that directly transmit their data are remain unclustered. This is named as flat layer and depicted in figure 2.8.

2.4.3 Clustering layers design

Cluster is form in second layer with the help of first layer of decision maker(dm) node. Second layer node communicate their nodal density on their turn to their dm to take part in the competition for becoming cluster head. Node with high node degree will be selected as head node for layer 2. Figure 2.9 shows cluster layer design

Figure 2.8: Flat layer design [49]

Figure 2.9: Cluster layer design [49]

of MCDA.

2.4.4 Cluster member selection

The elected cluster heads broadcast ”join request” packets. The recipient node send ”join accept” message to the cluster head. If node receive more than one join request than cluster head with low nodal density will be selected as cluster head.

Limitations of MCDA are given below

1) Nodal density is the only metric to select a cluster head, so it will make same node to select as cluster head again and again.

2) Number of cluster member for a cluster is not defined. Thats why cluster is not equally loaded.

3) Layer 1 node direct communicate with base station, this makes boundery node of layer 1 to die early

2.5 Time Division Multiple Access (TDMA)

TDMA is probably the best choice for an energy efficient multiple access protocol for wireless sensor networks [50]. TDMA protocols create a schedule for network activity: each node is assigned at least one slot in a time frame, which is considered to be the number of slots required to get a packet from each source to the sink. Slots are assigned to nodes for sending or receiving data, and when there is no activity the node can shut down its radio interface, thus saving power. It is very important that the slots are assigned in a way that prevents transmission interference, or conflicts.

Two or more nodes that don't conflict can be active in the same slot. In a multi-hop topology, this provides spatial reuse.

Depending on how the schedule is created, TDMA protocols are centralized or distributed [51]. In centralized approach the base station or cluster head gathers information about the network topology and distributes individual schedules to nodes. Centralized algorithms are very fast and simple, but they require a lot of traffic to build full topology information at the base station. They are best suited for static star topologies with a low number of networks, but not for multi-hop WSNs with many nodes, especially if the topology changes often. In sensor networks distributed TDMA algorithms are preferred over centralized ones as they are more suited to large numbers of nodes and they do not need full topology data. The problem that arises in distributed schedules is their adaptability to topology changes caused by node mobility, failure, or adding more nodes.

Thus, the main and most important advantage of TDMA is low power consumption. Also, it's achieve better channel utilization as the number of nodes is higher, which matches the WSN profile. TDMA allows different nodes to access the shared medium without interfering with each other and thus effectively avoids collisions. Furthermore, they can avoid the hidden terminal problem by scheduling transmission of interfering nodes at different times.

Nevertheless, a conventional TDMA protocol has a number of drawbacks which make it undesirable for certain sensor applications and/or network configurations. Some of these drawbacks are:

1. Tasks of slot allocation and schedule maintenance is complex [52].

2. It is difficult to change frame size and slot assignment when introducing new nodes.

3. TDMA requires strict time synchronization for the time-slots. [53].

4. In lightly loaded WSN environment the principles of TDMA cause the unnecessary wastage of energy of the sensor nodes and an unreasonable amount of delay.

There are some protocols that works on modification of TDMA. Like CALCA (Collision Avoidance Level based coloring Algorithm) [54] instead of using global clock-synchronization used local time difference between sensor nodes and their parent. In order to use unused slot owner of the slot start slot sharing approach. If owner of the slot have no data to send than it will share its slot with node who have data to send. OCS (On-demand Convergecast Scheduling) [52] MAC protocol is a centralized and adaptive multihop scheduling-based TDMA protocol, time slots are only assigned to nodes that are sources or relays of traffic. In addition, when there is no traffic is received from these sources for a specific time period, the sink will remove those nodes from the last slot assignment. BS-MAC (Bitmap-assisted Shortest job first based MAC) [55] adaptively handles the varying amount of data traffic by allowing nodes to send data who send data request in its control slot time. In addition, it implements Shortest Job First algorithm to reduce node's job completion time that results in significant improvement in average packet delay of nodes. Slot splitting is a technique to give chance newly added node to assign slot.

2.6 Summary

A detail overview of literature is discussed here along with basic of TDMA transmission. The basic observation from different literature is analyzed. Moreover, It has been observed from the literature study that different protocols used different techniques to reduce energy consumption among node. However, the existing schemes are expensive from communication, energy, and time perspectives.

Chapter 3

AN EFFICIENT WEDGE MERGING SCHEME FOR ENERGY HOLE MITIGATION IN WIRELESS SENSOR NETWORKS

In Wireless Sensor Network (WSN), nodes are equipped with restricted battery power, the usage of energy is a very major concern in a WSN. Therefore, efficient energy utilization of nodes has become a hot research area in WSNs. In multi-hop Wireless Sensor Network (WSN), nodes closer to Base Station (BS) need to carry traffic from other nodes of the network which makes their energy depleted very fast and cause an energy hole in the network. Moreover, energy consumption among nodes is not balanced due to "Non uniform" distribution of nodes and it causes some node to drain out their energy faster than other nodes in the network. In WSN, this energy hole problem plays a key factor to reduce the lifetime of the network as data cannot be sent from other sensor nodes to the BS although the residual energy in the network remains high. This means that a considerable amount of energy gets wasted and the network lifetime ends too early.

In this chapter, a layered-based routing technique with wedge merging option is proposed to solve the energy holes problem in WSNs. In order to distribute the energy consumption uniformly across the network, a wedge merging technique is proposed At first, the whole network will be divided into several coronas with equiangular wedges which in turn will create many sectors, an area between the corona and the wedge. The sensor nodes at each sector will form a chain to transmit data packets toward the sink via Head Node (HN) and a hop-to-hop delivery. HNs at each sector are selected based on their remaining energy and distance with its relevant successor HN.

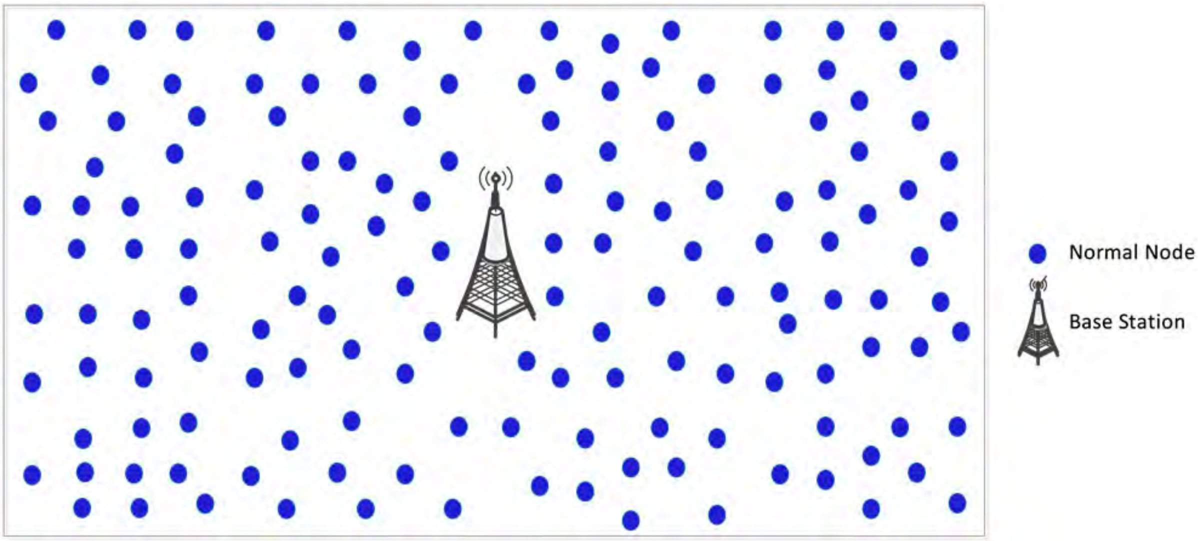


Figure 3.1: Deployment of nodes over the network with Base Station at the center

3.1 Network Model and assumptions

Assumptions about used sensor network model are as follows:

(1) A multihop based sensor network is assumed where a large number of sensor nodes are dispersed over a area and report their sensed data to a single BS. The only BS is located at the center of the network to increase the accessibility for all of the sensor node, as shown in Figure 3.1.

(2) Nodes and BS are static in nature, no mobile sensor nodes or BS is used.

(3) Sensor nodes are homogeneous which means they have similar sensing, processing and communication capability.

(4) After deployment all sensor nodes are left unattended. Therefore, energy cannot be recharged.

(5) The communication links are symmetric such that two nodes can communicate with each other by using the same transmission power.

(6) Sensors are required to send their sensed data constantly at a certain rate. For sake of simplicity, assume that each sensor node generates and sends l bits of data per unit time.

(7) Assume that there is a perfect MAC layer in the network. Collision and retransmission is not considered here.

(8) Each node can aggregate the data and the energy consumed in aggregation is EDA (nJ/bit). In data aggregation scheme, nodes only retransmit only the average

of the received data. It is also assumed that the sensed data is highly correlated so the nodes always aggregate the data gathered from its neighbor into a single length-fixed packet.

3.1.1 Energy Model

A typical sensor node comprises three basic units: sensing unit, processing unit, and transceivers. First order radio model is adopted for proposed scheme [46]. In this radio model, the electronics energy E_{elec} = 50nJ/bit to operate the transmitter or receiver circuit. The transmitter amplifier ϵ_{amp} = 5nJ/bit/message. It utilizes both channel models of the free space with d^2 power loss and multipath fading with d^4 power loss. The radios have power control and can expend the minimum required energy to reach the intended recipients. The radios can be turned off to avoid receiving unintended transmission Thus, to transmit a K-bit message a distance d using this radio model, the radio expends:

if $d < d_0$ than $E_{tr}(K, d)$ will be

$$E_{tr}(K, d) = K * E_{elec} + k * \epsilon_{fs} * d^2$$

if $d \geqslant d_0$ than $E_{tr}(K, d)$ will be

$$E_{tr}(K, d) = K * E_{elec} + k * \epsilon_{mp} * d^4$$

And to receive this message, the radio expends:

$$E_{rx}(K) = K * E_{elec}$$

where E_{elec} is the unit energy dissipation for transmitter electronics or receiver electronics. ϵ_{fs} the amplifier energy in the free space model while ϵ_{mp} is the one in the multipath model and d_0 is the threshold and defined as:

$$d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$$

3.1.2 Definition of the network

Following terms are defined for WEMER scheme.

Round: After completing chain construction phase, node send their data to chain head through the constructed chain in data transmission phase. This cycle is called round. In one round WEMER maintains the same chain configuration.

Network Lifetime: Network lifetime is the period until all sensor nodes of the network discharged their energy.

Merging Threshold: If any sectors average energy is below than 40% of its initial energy, the WEMER merges two wedge. For WEMER network merging threshold 40% of initial energy of a sector.

3.2 Proposed Wedge Merging Scheme

The lifetime of sensor node mainly depend on the number of alive nodes and connectivity of the network. Once a sensor node is out of energy it will die prematurely and will affect the performance of the network. The proposed scheme aims to eliminate the energy hole problem in WSN as well as to maximize the network lifetime. In this section, WEMER technique is presented in details. WEMER is divided into following two phases: i) Initial setup phase, and ii) Data aggregation and transmission phase. All above mentioned phases are discussed in the subsequent sub-sections.

3.2.1 Initial setup phase

At initial setup BS will divide the whole network into some sector. Sector is the small portion of the network, it is an area between the corona and the wedge. This initial setup phase divided into two phases: i) Corona creation procedure ii) Wedge Creation Procedure.

3.2.1.1 Corona creation procedure

This section describes how the network area will be divided into concentric coronas. Before starting data transmission, each node need to find the corona it belongs to. In the network initialization phase BS will divide the network into some circularly shaped coronas by adjusting its transmission power. The range of transmission for corona creation mainly set by the BS depending on the requirement of number of coronas and the node density in the network. BS adjust its transmission power equivalent to half of its transmission range and set its corona number as zero (CN=0). After increasing corona number by 1 BS broadcast a corona_creation packet. A node receiving a corona message with CN = i sets its corona to i, unless it has already joined an equal or a lower corona. By increasing corona number by

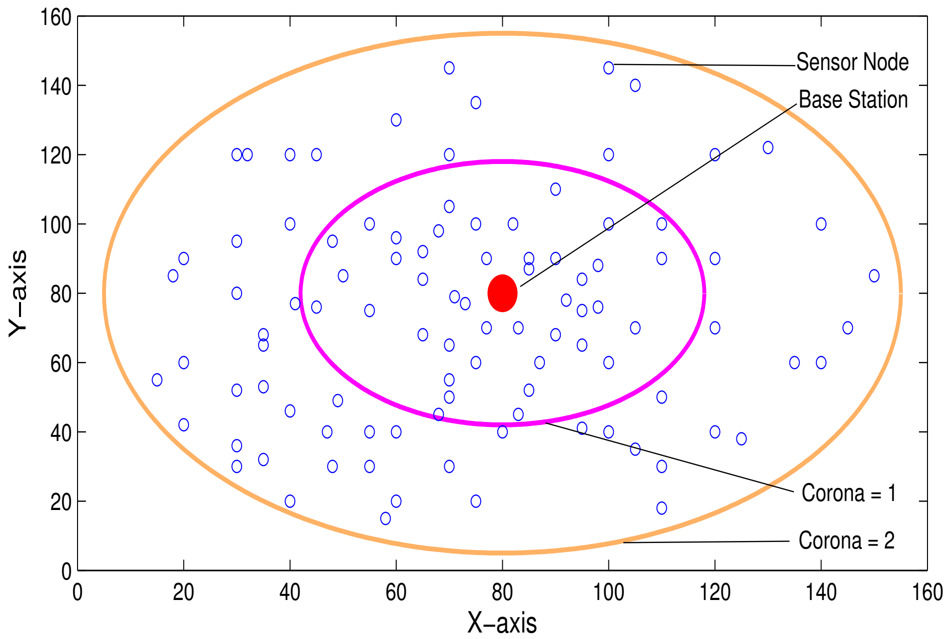


Figure 3.2: Network get divided into 2 coronas

1 and transmission range this corona creation procedure continues until all nodes get its corona number. After the corona creation procedure whole network will be divided into several coronas similar to concentric circles with center Sink. Figure 3.2 is showing network with 2 concentric coronas. The innermost circular region is corona 1 and outermost is corona 2 as with the help of BS the whole network is divided into two coronas. The pseudo-code of the corona creation procedure is shown in algorithm 1.

3.2.1.2 Wedge creation procedure

To divide the network into equiangular wedges the BS directs its antenna to one portion of the network and transmits wedge_creation packet with $sinkID$ and wedge number. This packet is transmitted with its maximum transmission power level to ensure coverage of every node in that direction. When any node received a wedge number then it can identify that they are in that wedge. With a change angle of directionality BS then again transmits and re-broadcasts a new wedge_creation packet and this process continues until the BS broadcast the wedge identifier beacon to the entire network. Figure 3.3 is showing network with wedges. Here sensor network gets divided into 12 wedges. The pseudo-code of the wedge creation procedure is

Algorithm 1 Concentric Coronas Creation

R = radius of circular monitoring area ▷ For the sink
r = maximum transmission range of sensor node
CN = corona number
while all node gets corona number **do**
  $i = 1$
  sink node create a corona_creation packet with CN field set to i
  adjust the transmission power equivalent to transmission range $i \times \frac{r}{2}$
  Broadacst corona_creation packet with (sinkID, CN)
  $i = i + 1$
end while

Initialize $cn = 0$ ▷ For any sensor node (i)
if (receive corona_creation packet and $cn = 0$) **then**
  set cn by CN field of received corona_creation packet
end if

shown in algorithm 2.

A unique sector is defined by the intersection of wedge and corona; every sector

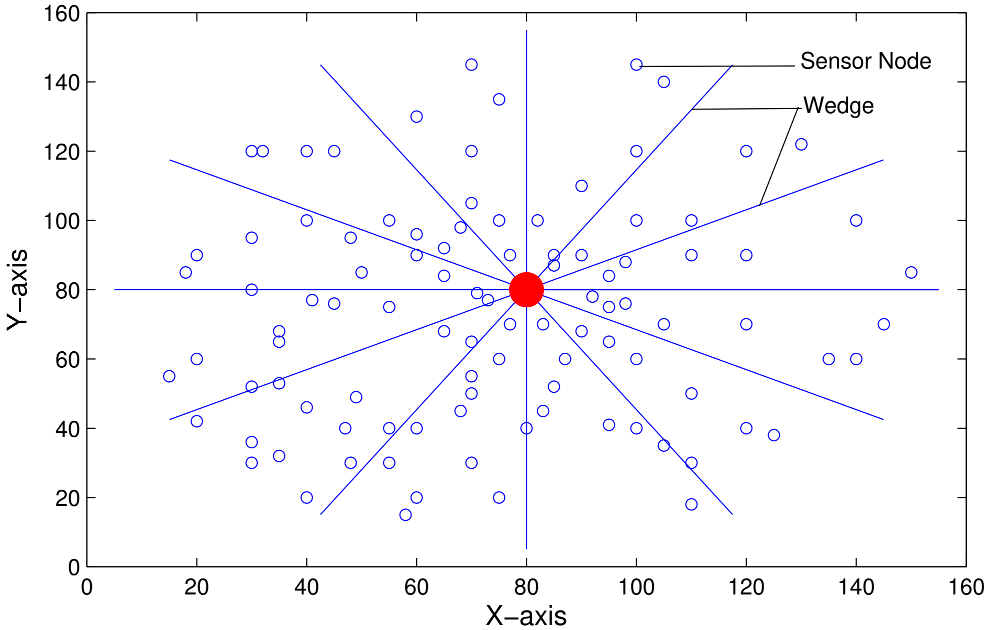


Figure 3.3: Network with 12 equiangular Wedges

is identifiable by the combination of unique wedge and unique corona identifier. The corona and wedge creation procedure guarantees that each node belongs to only one sector in the coordinate system and that node knows the identity of its sector.After the formation of corona and wedge each sensor node has acquired the identity of corona and wedge in which it belongs to. The positions of all the sensor nodes that have the same coordinates form a sector, like a hard disk. Figure 3.4 is showing network with corona, wedge and sector. In the figure number in the bracket means corona and wedge position for a sector. (1, 1) means position corona 1 and wedge 1, at the same way (2, 1) refer to position corona 2 and wedge 1. BS will check if every sector has more than one node, if not BS will merge that wedge which has empty sector with its neighbor wedge. Sector size varies according to distance from BS, the sector which is closer to BS is smaller in size and intra sector cost between that sector's node get reduced and they need to use small transmission energy.

Algorithm 2 Equiangular wedge creation

WN = wedge number ▷ **For the Sink**
$theta = 30$
adjust the transmission power equivalent to transmission range R
while $theta <= 360$ **do**
    sink create a wedge_creation packet with WN field set to j
    adjust angle of directional antenna by theta
    broadcast wedge_creation packet ($sinkID$, WN)
    theta = theta + 30
end while

Initialize $wn = 0$ ▷ **For any sensor node** (i)
if (receive wedge_creation packet and $wn = 0$) **then**
    set wn by WN field of received wedge_creation packet
end if

3.2.2 Data Transmission

If some of the nodes in the network only participate all of the processes while others are inactive or idle, then it produces more chances for that some node to die early

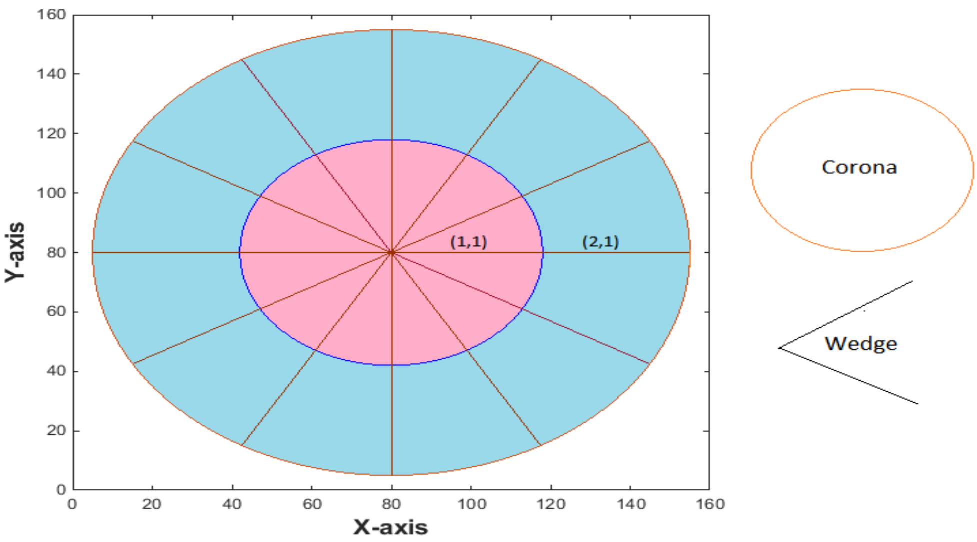


Figure 3.4: Network with corona, wedge and sector

and network partitioning which eventually decrease the network lifetime. WEMER works in round and according to figure 3.5 every round is further divided into three phases:

1) Head Node (HN) selection phase for a sector

2) Chain construction phase and

3) Data transmission phase.

As networks are multi-hop based routing idea is that every wedge will create a virtual path which will join the outermost sector to the BS to send the sensing information to the BS. Head node from a sector will send the cumulative traffic coming from its predecessors and its own sector's to its successors. As only one

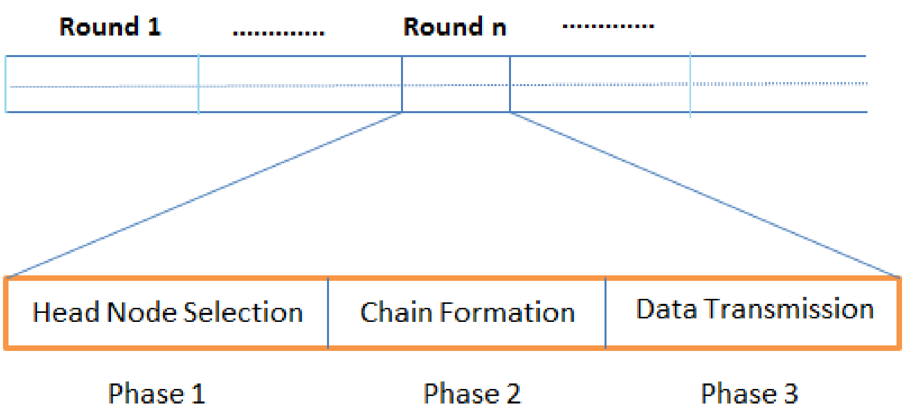


Figure 3.5: Data transmission Phase

node from a sector is used to forward the data to the BS so MAC level contention is very less.

3.2.2.1 Head node selection phase:

In WEMER, BS position is in the center of the network. So Head Node (HN) selection procedure will start from the inner most sector of the BS. For every sector there will be only one HN and HN selection depends on node's RE and distance with its relevant successor HN. The node whose RE is less than average energy of its sector will not participate in the HN selection competition. Every node for a sector will set a timer (Ti) according to the following equation to be a HN,

$$T_i = (Einit/Ei) + (DitoHN)(1)$$

Here E_{init} is the initial energy of the node, E_i is the node's RE, D_{itoHN} is the distance between node's and its successor head node (for nodes which are in corona 1 will consider their distance with BS, for them D_{itoHN} will be D_{itoBS}). The node whose timer first elapses to zero will broadcast a HN_Msg to its sector to inform other node to leave the head node selection competition. Sensor nodes which are in same sector, after receiving HN_Msg, will leave the competition. Every head node

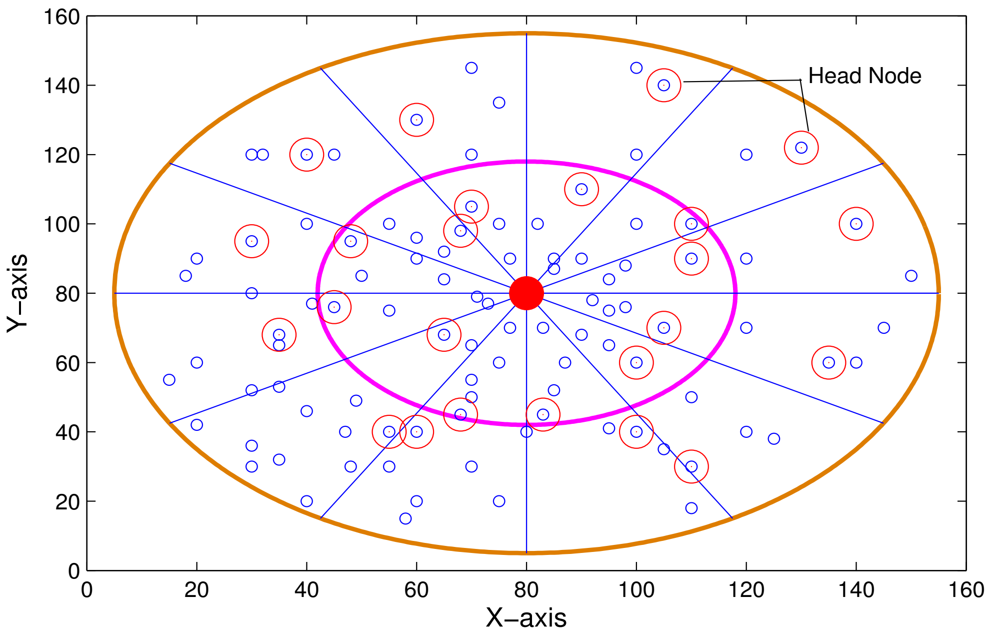


Figure 3.6: Selection of Head Node for every sector at round 1.

informs its own location information to its neighbor HN of a sector. Figure 3.6 is showing Head Node for every sector. This HN selection procedure will run at every round and same HN can operate for many continuous rounds until any node becomes more capable than the recent HN. If any node becomes better than the current HN of its sector again new HN selection phase will be called and the most suitable node will become new HN. The pseudo-code of the head node selection procedure is shown in algorithm 3.

Algorithm 3 Head node selection procedure for sector

E_{init} = initial energy of the node
E_i = node's Residual Energy
$D_{ito}BS$ = distance between node and BS
$D_{ito}HN$ = distance between node and its successor head node
for every node of corona calculate its timer T_i **do**
    if corona_number = 1 **then**
        $T_i = (Einit/Ei) + (DitoBS)$
    else
        $T_i = (Einit/Ei) + (DitoHN)$
    end if
end for
for every sector **do**
    node whose timer T_i first elapses to zero will broadcast HN_Msg
    if received HN_Msg node **then**
        received node will leave the head node selection competition.
    end if
end for

3.2.2.2 Chain Construction Phase:

The main concern of chain construction phase is to find a low cost chain that covers all nodes of the sector. PEGASIS and CCS used greedy algorithm to build a chain starting from the farthest node of the network. Greedy algorithm is always a good choice for chain construction but for large networks it causes a serious problem that

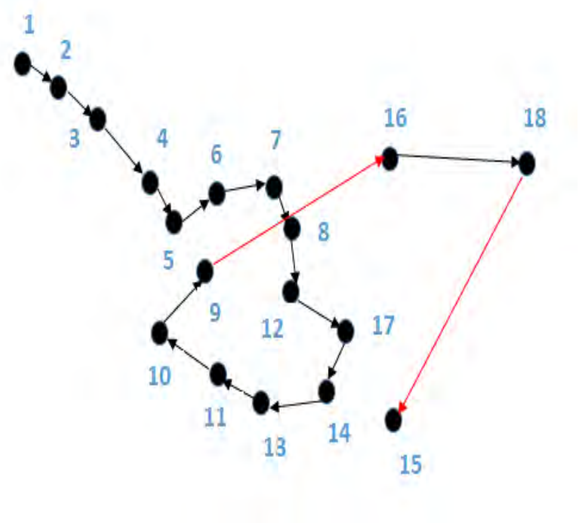


Figure 3.7: Long link problem

leads to long link(LL) problem [14, 56].

Figure 3.7 is showing the scenario of LL problem. In order to solve LL problem proposed scheme make an enhancement over the greedy method. Except connect with next minimum distance node that are not in the chain, WEMER connect with that node which is already in the chain and with whom distance is minimum. In each sector, the chain construction procedure will be started from the farthest node

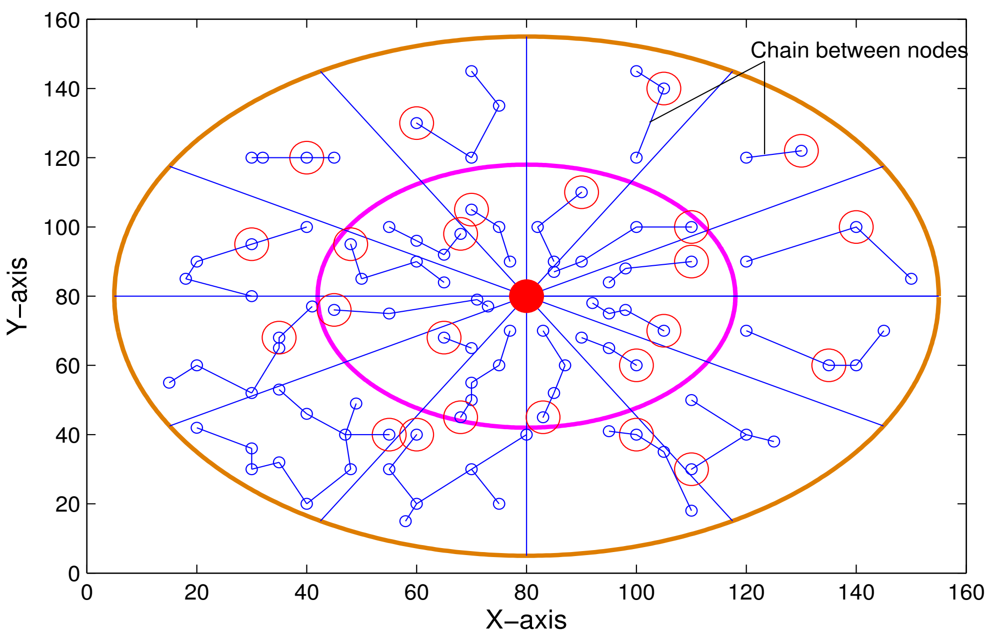


Figure 3.8: Chain construction for every sector with head node at round 1

from the head node and it will join the chain first. Than it will find the distance between itself and other nodes which have not joined the chain yet, finds the nearest node and sets it as node i waiting to join the chain, i represents the ith node joined. Node i gets the information of distance between itself and $i-1$ nodes which are on the chain, finds the nearest node j and directly connects with it to join the chain, at this point node i becomes the new end node of the chain. This procedure continues till all sensor nodes have joined the chain, so that there forms a branching chain finally. Figure 3.8 is showing construction of chain for every sector.

Algorithm 4 Chain construction procedure for sector

```
chain_head_point = hn
for all node (i) of sector that are not in the chain
if distance with i and chain_head_point is minimum then
    select i as next node to join the chain
else
    continue
end if
selected i node will join minimum distance chain node
now new end of the chain will be i, i.e, chain_head_point = i
continue chain construction procedure until all active node join the chain
```

In order to distribute energy load evenly among all nodes, LL should be avoided. This new chain creation procedure solves the problem of LL(Long Link) problem of greedy algorithm. So new chain construction method ensure less energy consumption of nodes. The pseudo-code of chain construction procedure is shown in algorithm 4.

3.2.2.3 Data Communication Phase

After WEMER performs first two phases, the Head Node(HN) initiates a token passing approach to start the data transmission from the maximum distance node from HN of the chain. The chain formed in WEMER has more than two end nodes. If for a node connected node count is zero; i.e no node is connected with it than it will be called the end node/tail node of that chain. As each chain may have more than two end nodes so before data transmission every end node need to inform if it is head node or not to its connected node. This information helps right flow of

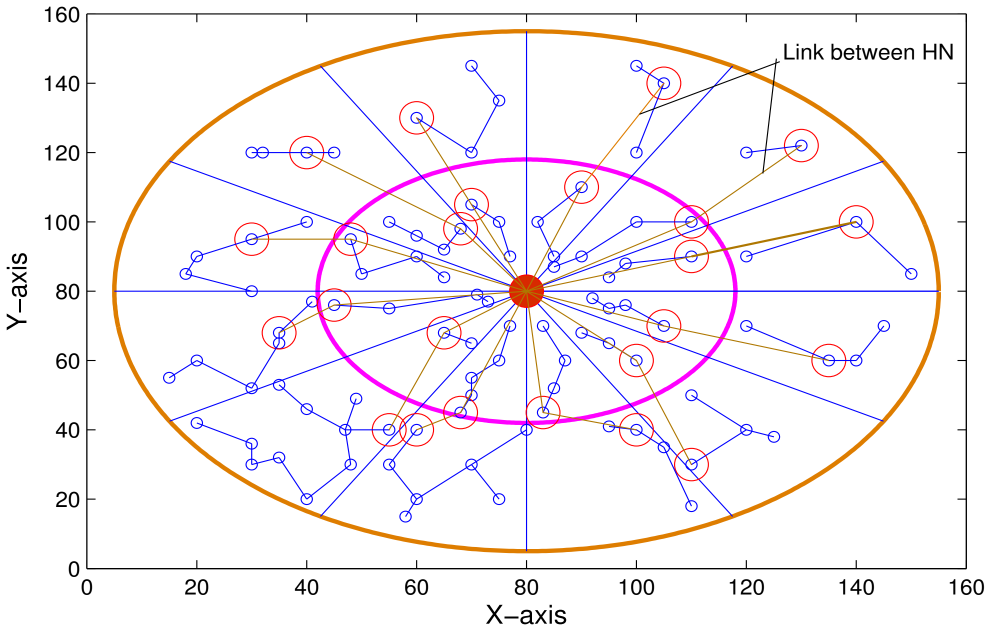


Figure 3.9: Data Communication Phase

data transmission. As chain has more than two end nodes so it is necessary to know the right path to reach HN. Thats why every end node inform its connected node if it is HN or not. The pseudo-code of data communication procedure is shown in algorithm 5. Each node delivers own sensing data to its neighbor node in their time slots assigned by TDMA mechanism, then the neighbor nodes fuse them with their own data and transmit these data to their other neighbor nodes. One round will end when Sink receives the data from the HN. Sensor node will get their TDMA schedule for data transmission from its HN. All nodes in each sector transmit their data along the chain to its nearest node in its slotted time. When HN collects all the data from its sector then it will send this to its successor HN or BS. This data transmission procedure starts from the outer most coronas. When HN within Corona 1 of a sector receives all data from its predecessor sector, after aggregation of all these data will send to the BS. As in this scheme HN for every sector is changed in every round according to distance and energy, so it reduce the burden of some node to elect repeatedly for HN. Figure 3.9 is showing connection between nodes along with link between HN.

Algorithm 5 Data Communication procedure

```
/*before data transmission except head node of the sector every node need to confirm that it is not the head node to its connected node. */
for (every node that connected node count is zero) do
    if node i is not the head node then
        inform connected node j
        for (every connected node of j) do
            if connected node is the head node then
                node j is in the path of head node
            end if
        end for
    else
        i is the head node
    end if
end for
```

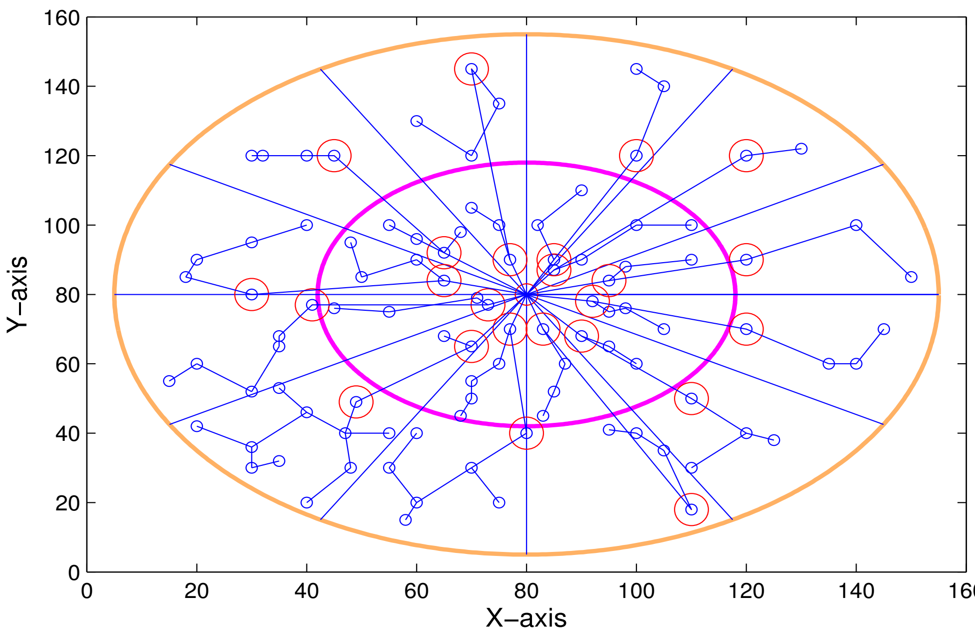


Figure 3.10: Every node in their sector connected by a chain at round 351

3.2.3 Merging Procedure of a wedge

After data transmission phase of every round, WEMER checks the average energy of every sector. If any sectors average energy become very low, it may create an

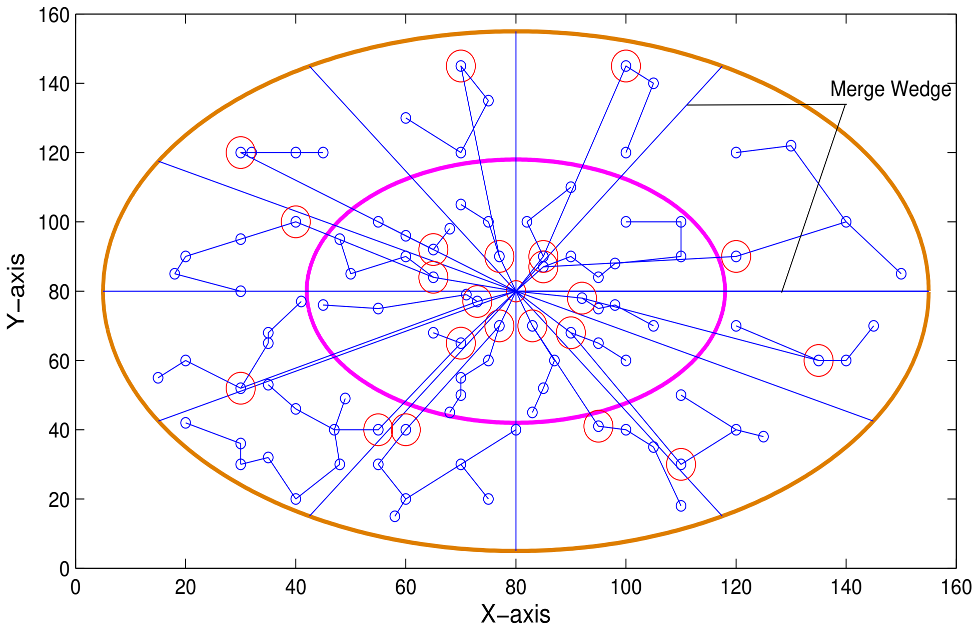


Figure 3.11: First wedge merge at round 352. New merge wedge with HN and chain

energy hole in that portion of the network. The result of energy hole is network partition along with sink isolation and network failure. Moreover it degrades network performance. For this reason energy hole problem is needed to be eliminated in order to maximize the network lifetime; and therefore wedge which has a sector with low energy needed to be merged with high energy wedge. The wedge merging phase is comprised of two steps: i) wedge merging step, ii) rearrange of nodes position.

Wedge merging step Most of the energy lost for sensor node in data transmission time. So every HN need to check average energy of its sector after every round. After creation of corona and wedge, BS needs to check if every sector has more than one node, if not than this sector need to merge with neighbor sector whose energy is higher. Even when any HN finds that its sector average energy is lower than 40% of its initial energy, than this HN needs to start merging of sector. Figure 3.11 is showing wedges after merging. Before starting of wedge merging, the sector which has lower average energy asked its two (left and right) neighbor HN to inform their sectors average energy by sending a beacon named avg_en. As every HN knows their energy information, so they will reply by sending average energy information to the requested HN. After getting reply from neighbor HN, the sector with higher energy will be selected to merge. Then merging procedure will be started from the innermost sector of that wedge which has low energy and ended at outermost sector of that wedge. Suppose that sector which is in corona-2 and wedge-2 position means (2, 2) position has low energy and decided to start merging, it will ask sector (2,3) and (2,1) for sending their average energy. Sector (2, 3) will be selected for merging as this sector has more average energy than sector (2, 1). At first, sector (1, 3) and (1, 2) will merge and all node from sector (1, 2) will update their sector as (1, 3). After that sector (2, 2) and (2, 3) will merge and this procedure will continue until the outermost sector of the low energy wedge sector get merge with.

Update of nodes position Every node which is merged with sector will update their position according to the new wedge and in this way every sector of low energy and high energy wedge will merge and become one wedge. Neighbor of merged wedge will update their neighbor information. The new wedge will select their HN and will create new chain for data communication.

In figure 3.11 the first merged wedge is showing. At round 351 every sector has enough energy to continue its data transmission. It is showing in figure 3.10. But after data transmission completes at round 351 HN of sector (2,2) found that its average energy is lower than 40% of its initial energy, so it will start to merge with other sector. From two neighbor sector (2,1) and (2,3) it found sector (2,1) has more

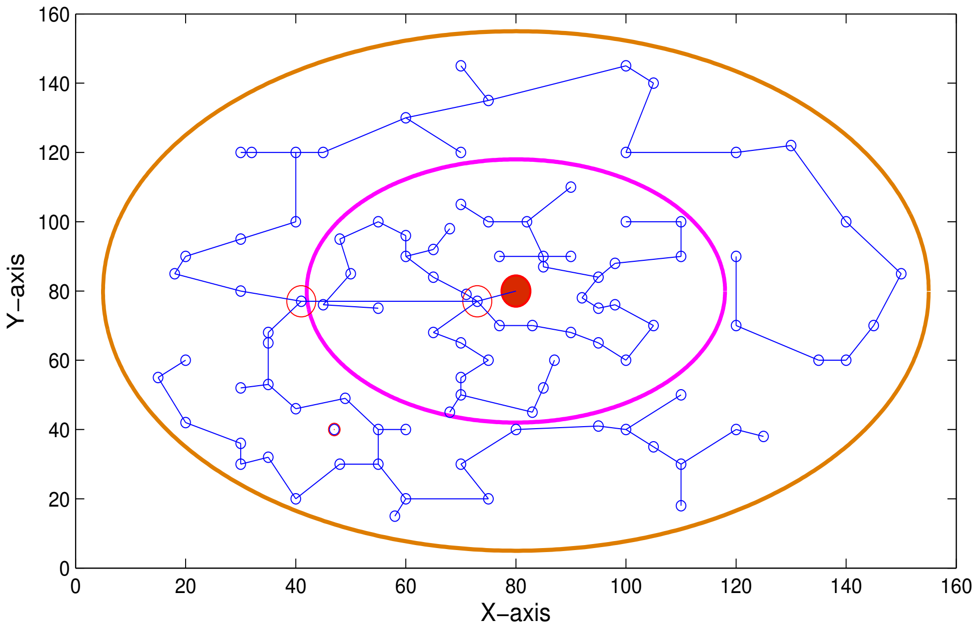


Figure 3.12: All wedge merge

energy than sector (2,3). So it will merge with sector (2,1) and a new sector will be construct and it is showing in figure 3.11. As the low energy sector merged with sector of high energy so it is again getting more high energy nodes to make a chain. Figure 3.12 is showing when all wedge merge in the network.

3.3 Delay efficient approach

In order to make WEMER delay efficient distance threshold concept is used. By using wedge merging technique when one or more wedge merge that time a node that are more distance from the Head Node(HN) needs to travel a long distance to send their data to the HN. This makes delay data transmission of the network. In order to make WEMER delay efficient a new idea is introduced, distance threshold (d_{th}) is used to reduce delay of the network. When in a sector maximum distance between two nodes become greater than d_{th} value than more than one HN will be needed for that sector.

After wedge merging every node will calculate their distance with other node of the sector. If the maximum distance between two nodes greater than d_{th}, than according to the distance more HN will be selected for that sector. As width of corona one and corona two is not same so two different distance threshold is used, for corona one distance threshold is 30 whereas for corona two distance threshold is

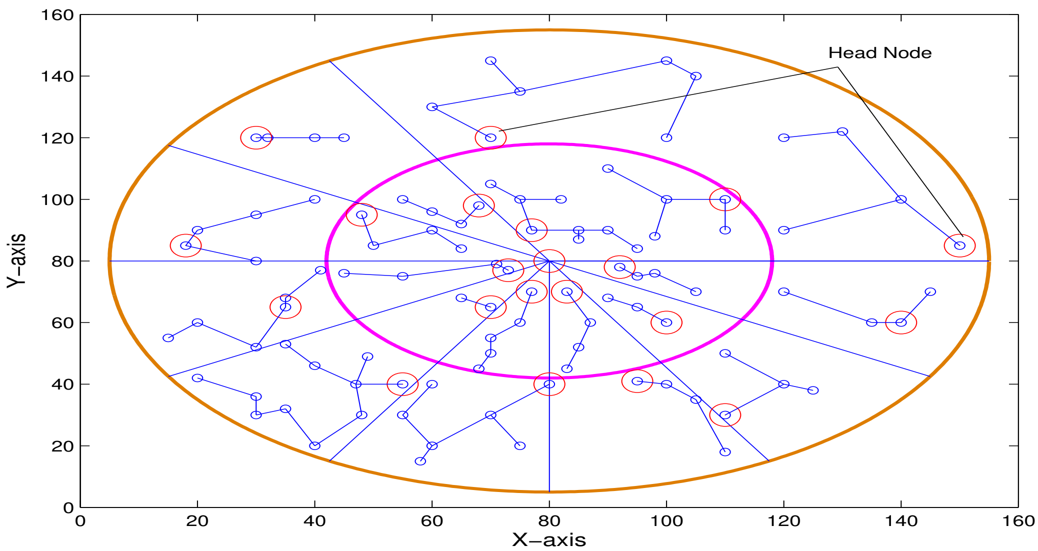


Figure 3.13: More than one HN for a sector is selected at round 535

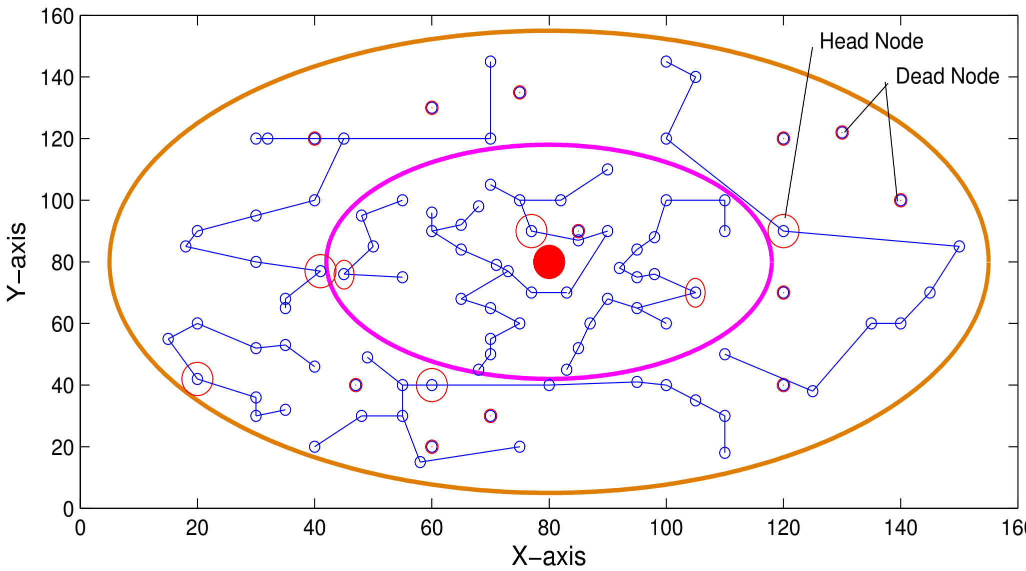


Figure 3.14: When all wedge merge

45. Figure 3.13 is showing two HN in a sector at round 535.

Multiple HN selection procedure is as like as single HN selection type. But the difference is that in order to make certain distance between HN when first HN select, it will send HN_Msg to its sector within transmission range d_{th} and the node who receive this massage will not participate for any other HN selection competition for this round. And the node who did not receive any message will continue head node selection procedure to be the second HN. This procedure continues until all nodes receive HN_Msg. Figure 3.14 is showing when all wedge merge in the network. After the HN selection procedure in order to construct a chain node will join to that HN which is close to it. Chain construction and data transmission technique is same as single HN procedure type.

3.4 Some Scenarios

If all node of corona 1 died and node of corona 2 is still active, than corona 2 node will direct communicate with base station. After nodes die in the network how the nodes connect are given below:

3.4.1 Network After First Node Died

Figure 3.15 shows the network condition when first node is died for WEMER and all three compared protocol. Figure 3.15 (a) is showing nodes data transmission condition for PEGASIS. For PEGASIS at round 115 first node died and when a node died a new chain is constructed bypassing that dead node. At the same way for CCS, MCDA and WEMER's first node died at round 311, 388, 582 and showing in figure 3.15 (b), 3.15 (c), 3.15 (d). When node died in the network that node cannot take part in data transmission so it is necessary to make energy consumption of every node very low so that it get died after a long time. It is very clear from the figure than WEMER performs better than all three protocols. The more the nodes involve in data transmission process it will help BS closer node to save their energy or protect high energy consumption. At every round chain construction procedure along with HN selection took place so it make sure that no node select HN repeatedly. After the wedge merging procedure merged sectors node update their position according to new wedge.

3.4.2 Network After Ten Percent of Node Died

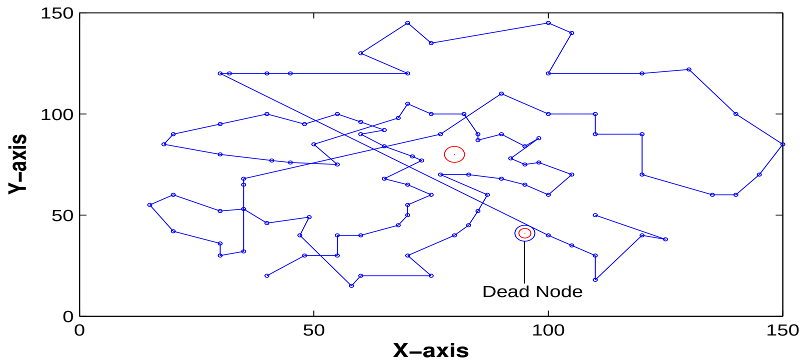
Figure 3.16 shows the network condition when ten percent of died for WEMER and all three compared protocol. Figure 3.16 (a) is showing nodes data transmission condition for PEGASIS. For PEGASIS at round 645 ten percent of its node died and when a node died a new chain is constructed bypassing that dead node. At the same way for CCS, MCDA and WEMER's ten percent of node died at round 751, 601, 933 and showing in figure 3.16 (b), 3.16 (c), 3.16 (d).

3.4.3 Network After Fifty Percent of Node Died

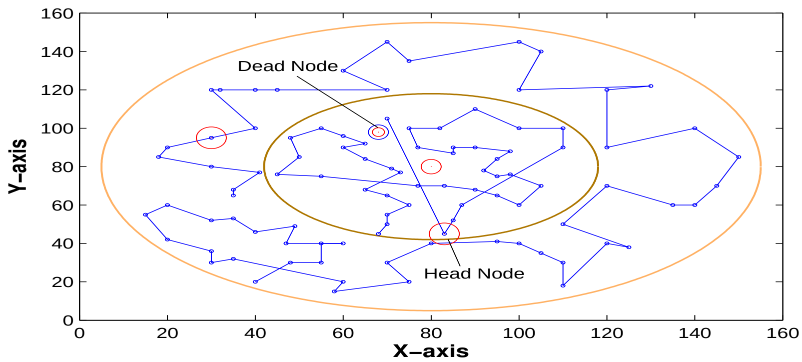
Figure 3.17 shows the network condition when fifty percent of node died for WEMER and all three compared protocol. Figure 3.17 (a) is showing nodes data transmission condition for PEGASIS. For PEGASIS at round 945 fifty percent node died and when a node died a new chain is constructed bypassing that dead node. At the same way for CCS, MCDA and WEMER's fifty percent of node died at round 951, 987, 1128 and showing in figure 3.17 (b), 3.17 (c), 3.17 (d).

3.5 Summary

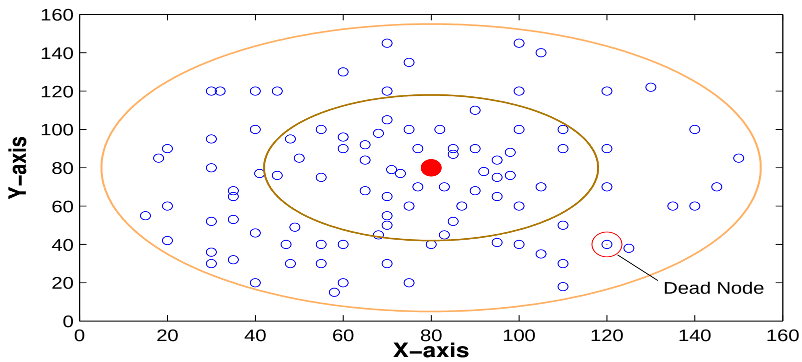
The main purpose for proposed work is to eliminate the formation of energy hole by distributing the traffic uniformly across the network. A combination of efficient chain construction technique along with wedge merging option is proposed to overcome the uneven energy consumption in the network. In last section a delay efficient procedure is proposed to reduce transmission delay. Through the wedge merging procedure, WEMER prolongs the network lifetime.



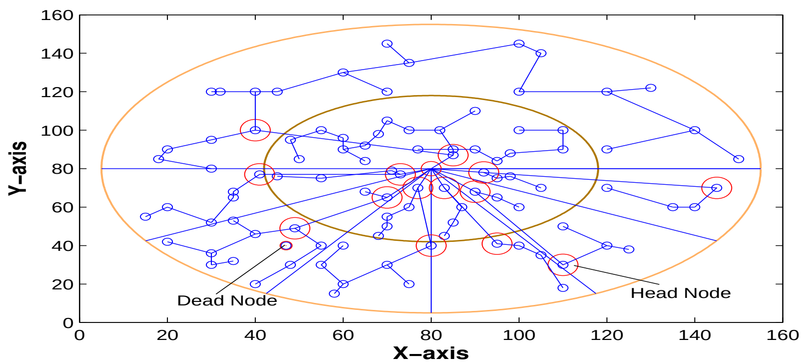
(a) PEGASIS



(b) CCS

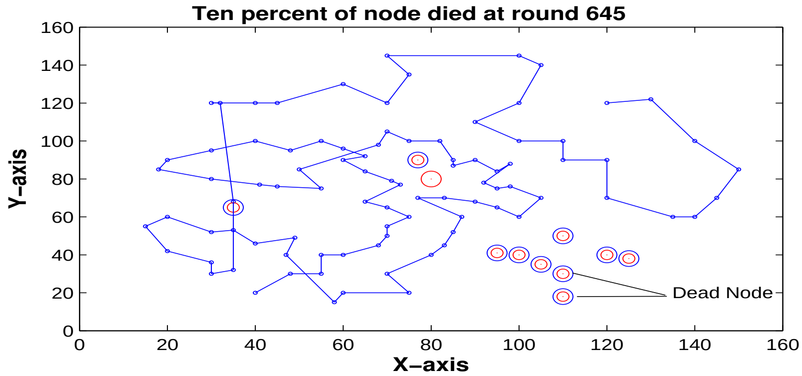


(c) MCDA

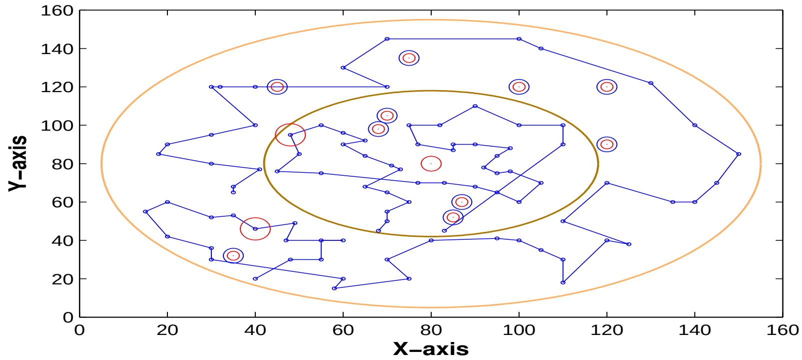


(d) Proposed WEMER Scheme

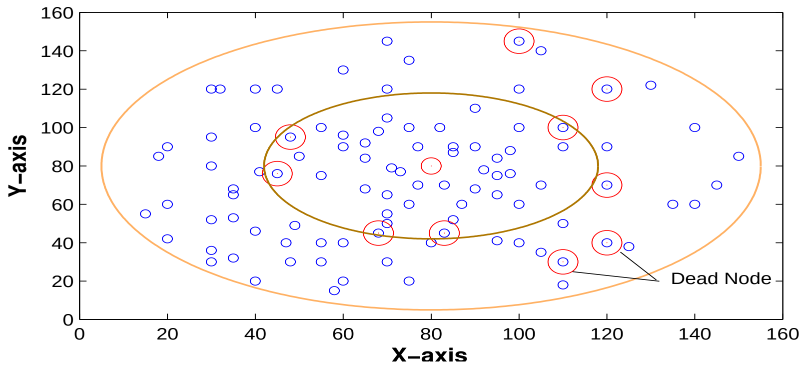
Figure 3.15: First node die time (a) at 115th round for PEGASIS (b) at 311th round for CCS (c) at 388th round for MCDA (d) at 582th round for WEMER



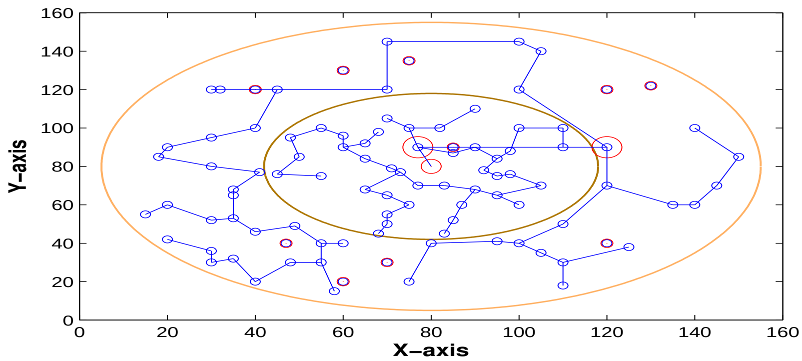
(a) PEGASIS



(b) CCS

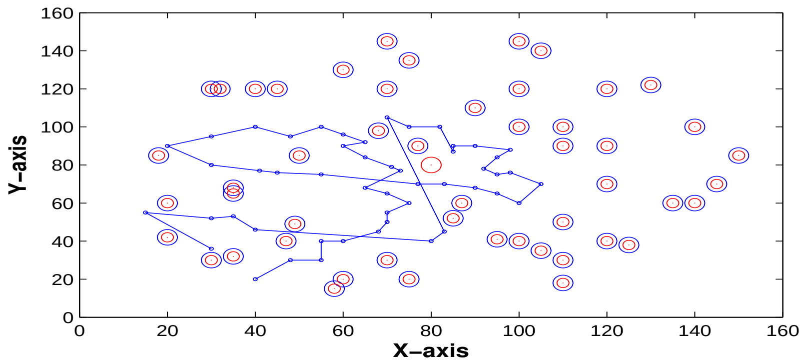


(c) MCDA

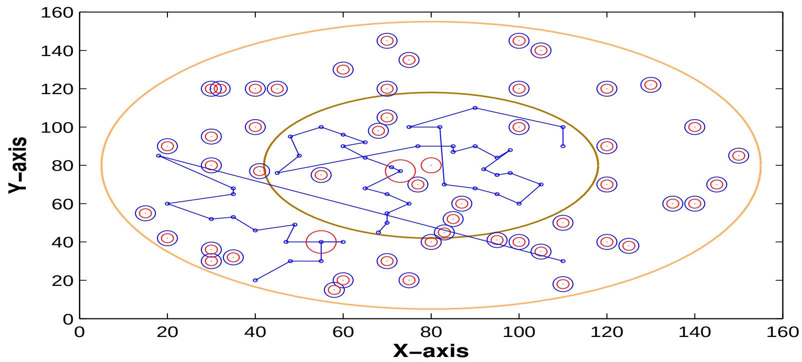


(d) Proposed WEMER Scheme

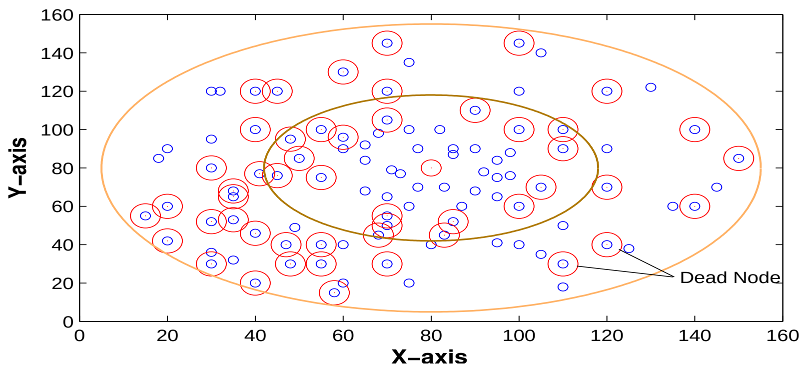
Figure 3.16: Ten percent of node die time (a) at 645th round for PEGASIS (b) at 751th round for CCS (c) at 601th round for MCDA (d) at 933th round for WEMER



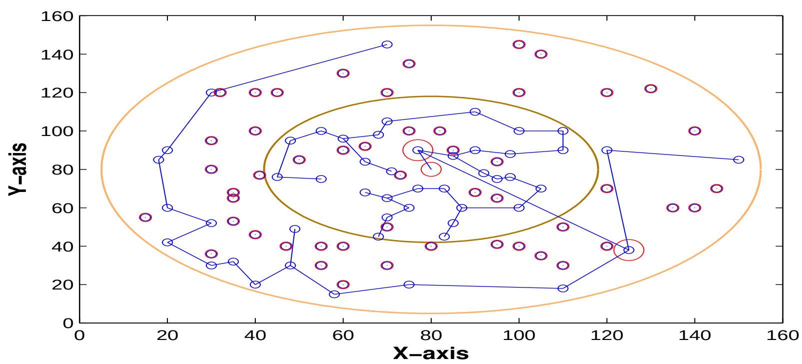
(a) PEGASIS



(b) CCS



(c) MCDA



(d) Proposed WEMER Scheme

Figure 3.17: Fifty percent of node die time (a) at 945th round for PEGASIS (b) at 951th round for CCS (c) at 987th round for MCDA (d) at 1128th round for WEMER

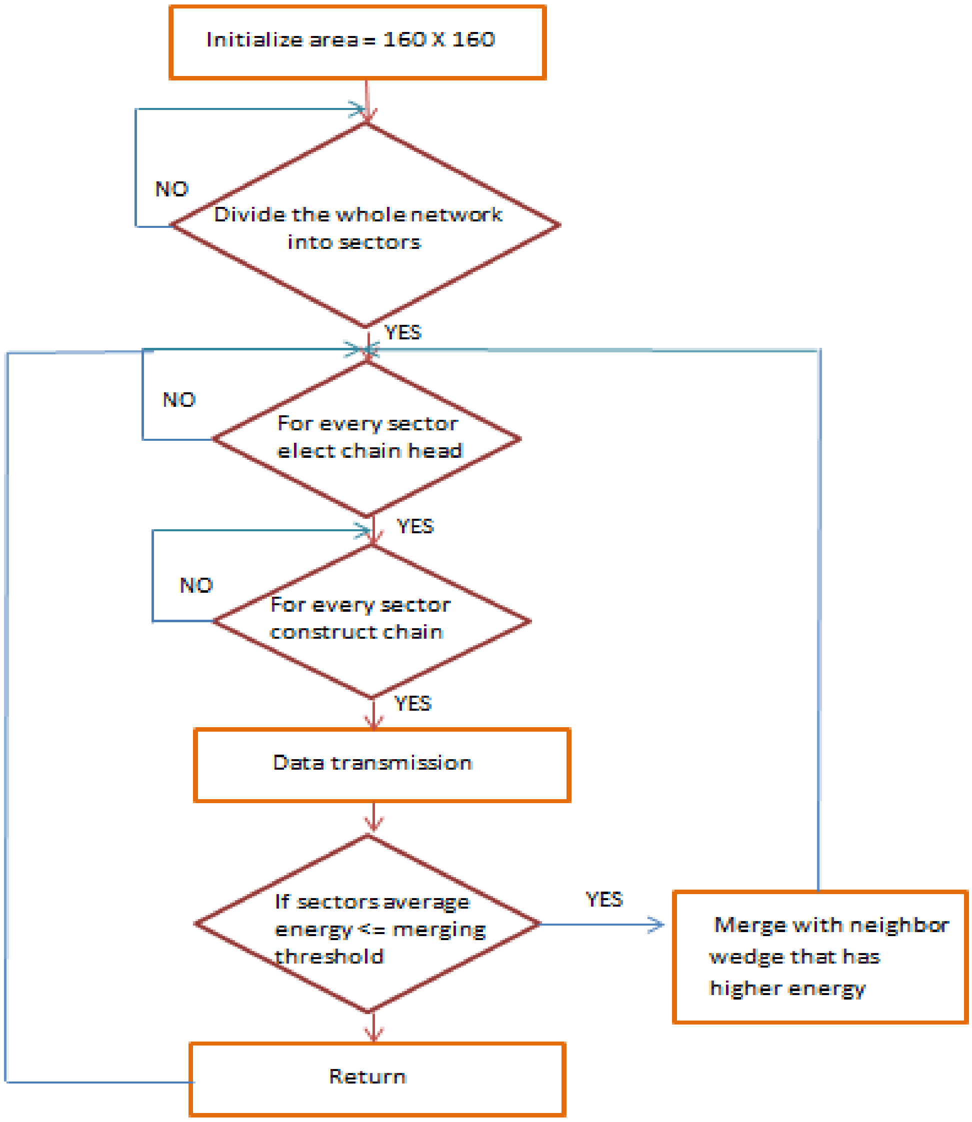


Figure 3.18: Flow chart of WEMER

Chapter 4

RESULTS AND PERFORMANCE EVALUATION

In this chapter, extensive simulation results are provide with comparisons of the performance between proposed WEMER technique and three other popular methods like Power Efficient GAthering in Sensor Information Systems (PEGASIS), Concentric Clustering Scheme (CCS) and Multilayer Cluster Designing Algorithm (MCDA). Study the performance of energy consumption, network lifetime as well as alive and dead nodes over round in this chapter.

Simulation is done in MATLAB [57] environment to evaluate the performance of proposed WEMER scheme with three popular protocols PEGASIS, CCS and MCDA. By using the simulation result with different simulation scenarios WEMER is compared with previous protocol. In this simulation environment BS is static and located at the center of network which is indicated by the red circle in figure 4.1 and it does not have any energy limitation. Network contains 2 circular shaped coronas and 12 wedges. BS is located at (80,80) coordinate position and all homogeneous and stationary 100 sensor nodes are deployed over the network in 160 X 160 regions.

4.1 Experiment Setup

System parameter is showing in table 4.1. Homogeneous sensor network is assumed in which all sensor nodes are having equal sensing and processing capabilities initially. Each node has initial energy of 0.5 Joule and length of data message is 4000 bits. Each sensor node has location information and no mobility is taken in account means nodes are static in nature. In proposed scheme data is passed through multi-hop approach.

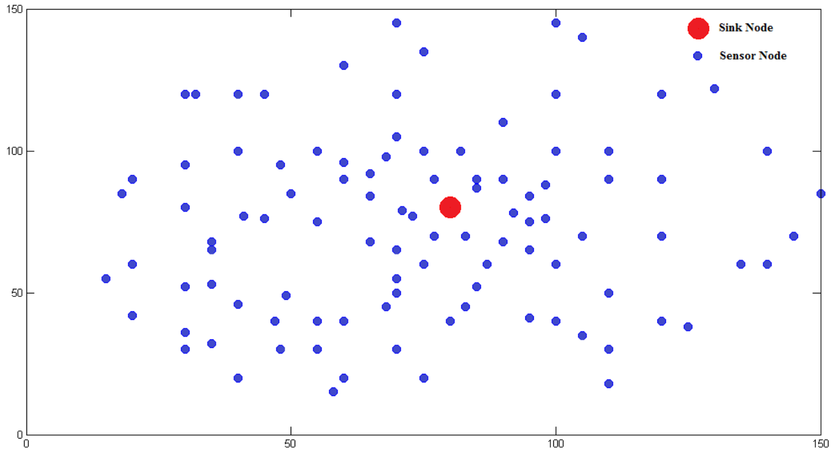


Figure 4.1: The network topology

Table 4.1: System Parameters

Parameters	**Value**
BS location	(80m, 80m)
Area	(160m X 160m)
Number of Nodes	100
Initial Energy	0.5 Joule
Packet Size	2000 bits, 4000 bits
ETx	50 nJ
ERx	50 nJ

4.2 Results and Discussion

In this section performance analysis of the WEMER scheme will be carried out through simulation results. Performance improvement in terms of network lifetime, energy cost and balanced energy consumption will be investigated and compared with existing protocols PEGASIS, CCS and MCDA.

4.2.1 Residual energy of network per round

Remaining energy of the sensor nodes after round effects on the network lifetime directly. If remaining energy of the nodes is high, the network lifetime will be high; and vice versa. Figure 4.2 demonstrates the performance comparison in terms of residual energy of the network with respect to round. Performance is studying by changing the value of data packet length. Fig 4.3 shows the result of same metrics for data packet length 2000. The x-axis shows the round whereas y-axis shows the total residual energy of nodes in the network. From the figure 4.2 and 4.3 it can say that proposed protocol is more energy efficient than PEGASIS, CCS and MCDA. As selection of Head Node(HN) for every sector is depending on distance with relevant HN and residual energy so the node with low residual energy and isolated from other node of the sector will not select as Head Node.

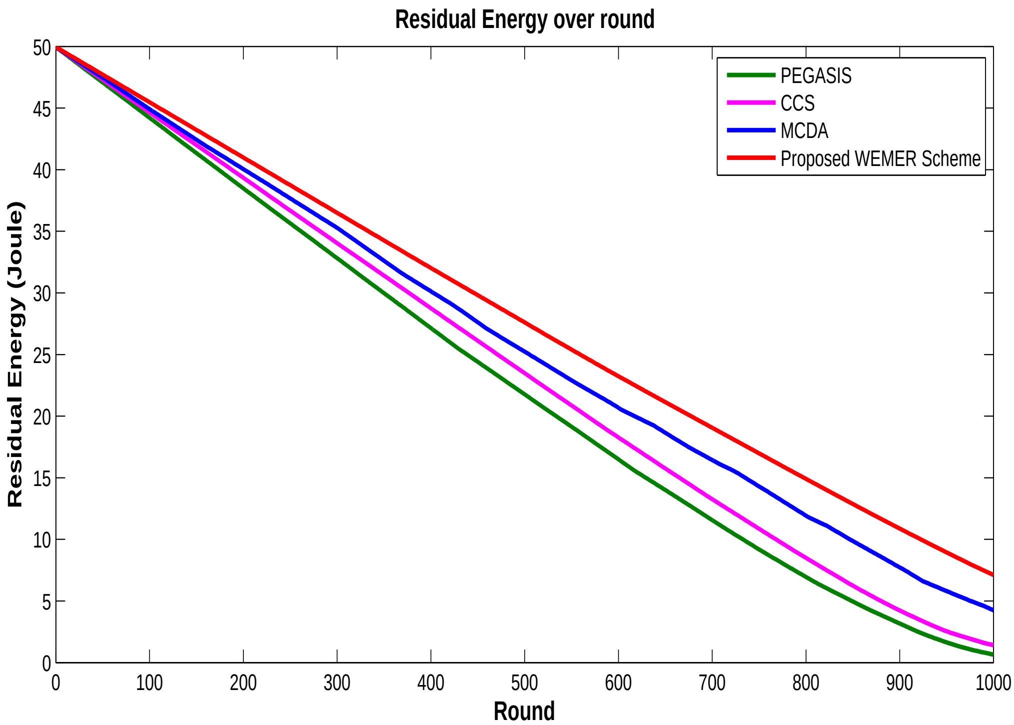


Figure 4.2: Residual energy of network with respect to round when data packet length is 4000

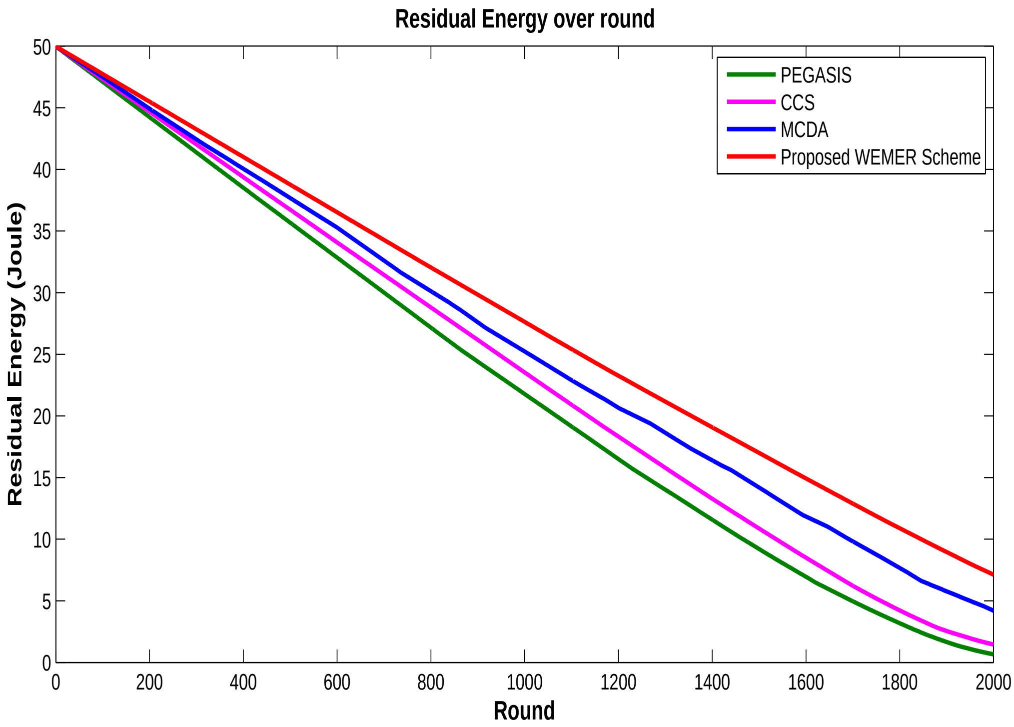


Figure 4.3: Residual energy of network with respect to round when data packet length is 2000

4.2.2 Number of alive nodes over round

The number of alive nodes specifies the level of the network lifetime. If it is high in most time of the simulation process, the network lifetime is high; and vice versa. Study the performance by changing the value of data packet length. Figure 4.4 and 4.5 are showing number of alive nodes of the network over round. The x-axis shows the round whereas y-axis shows the number of alive nodes in the network. Thirty percent of node die for PEGASIS, CCS and MCDA at round 844, 892 and 831, whereas at round 1075 thirty percent of node died for WEMER. PEGASIS protocol construct chain by using greedy method so it creates Long Link (LL) between nodes during chain construction, so that nodes deplete energy very fast and die early. CCS also used greedy method to construct chain and it also suffer from LL, whereas MCDA is free from LL but as its layer 1 node direct communicate with the BS so the nodes which are in the boundary deplete their energy very fast than other node. Chain construction procedure of WEMER is so efficient that reduces the distance and LL between sensor nodes. Also Head Node (HN) selection depend on distance

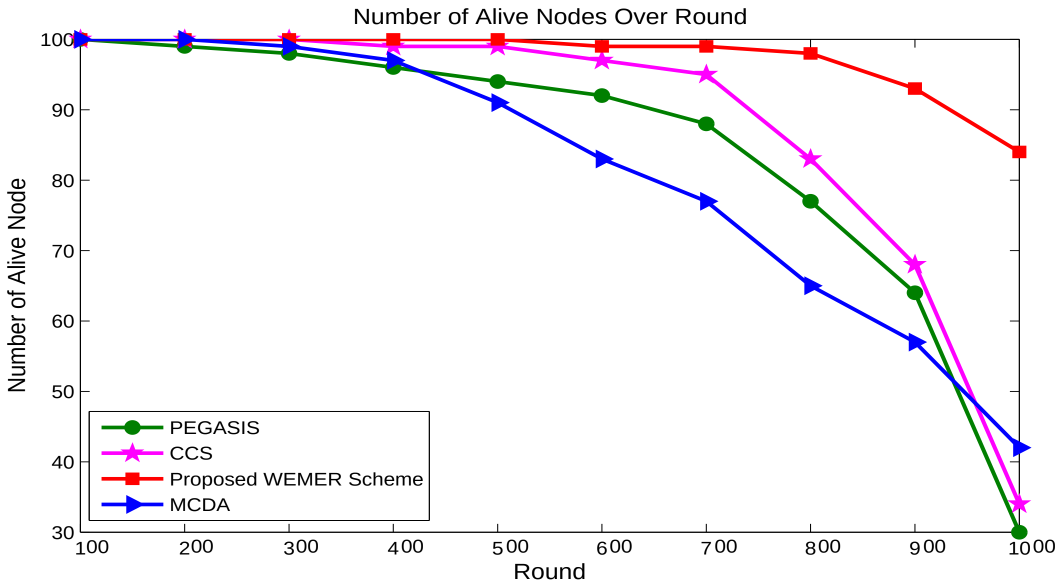


Figure 4.4: Number of alive nodes over round when data packet length is 4000

between successor HN or BS and energy, so all nodes get the chance to be HN. All these considered parameter of WEMER reduces energy consumption of nodes and increases lifetime of the network.

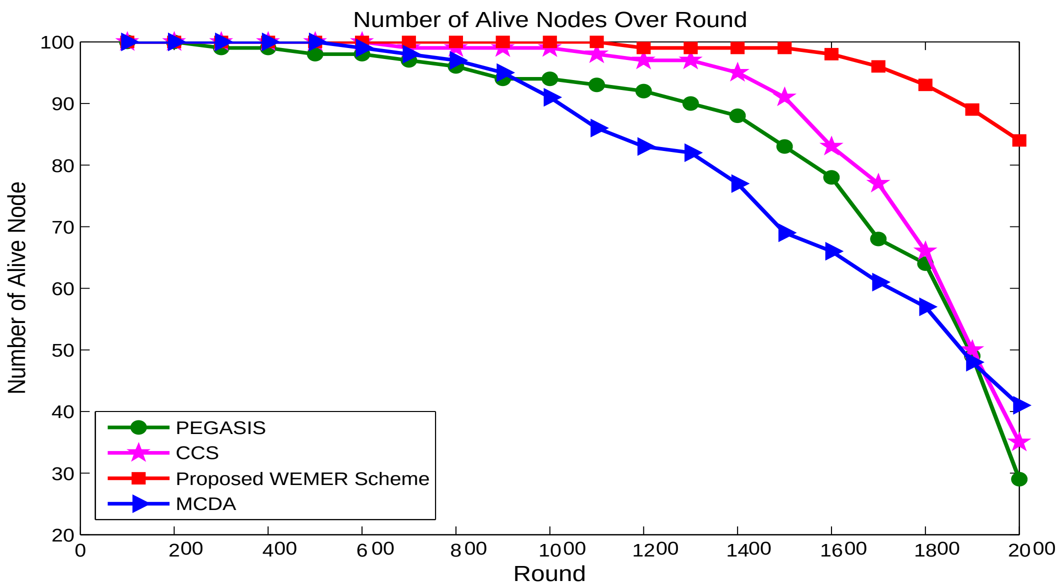


Figure 4.5: Number of alive nodes over round when data packet length is 2000

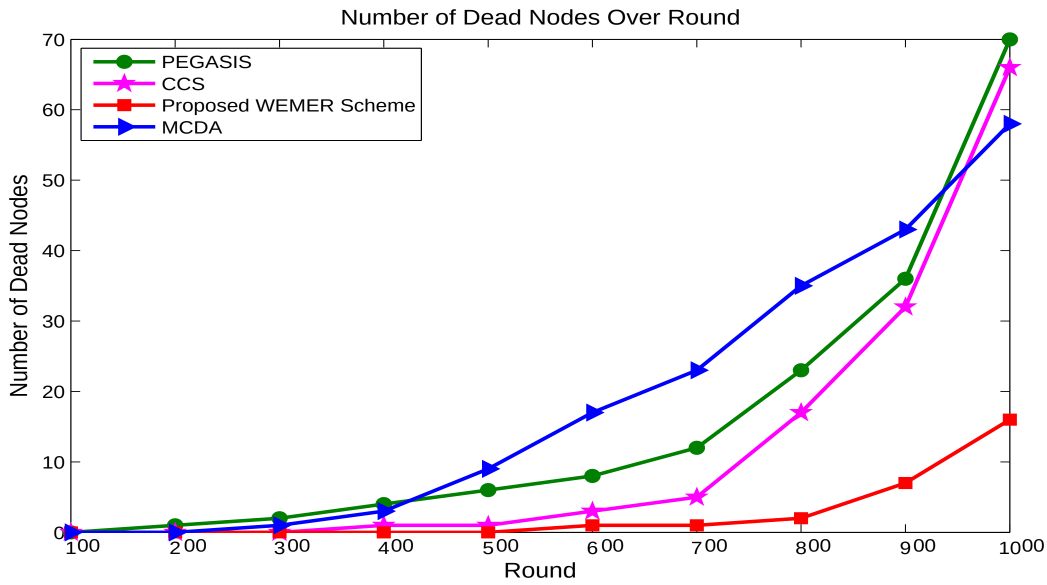


Figure 4.6: Number of dead nodes over round when data packet length is 4000

4.2.3 Number of dead nodes over round

The number of dead nodes specifies the level of the network lifetime. If it is high in most time of the simulation process, the network lifetime is low; and vice versa. Figure 4.6 and 4.7 are showing number of dead nodes of the network over round. Study the performance by changing the value of data packet length. The x-axis shows the round whereas y-axis shows the number of dead nodes in the network.

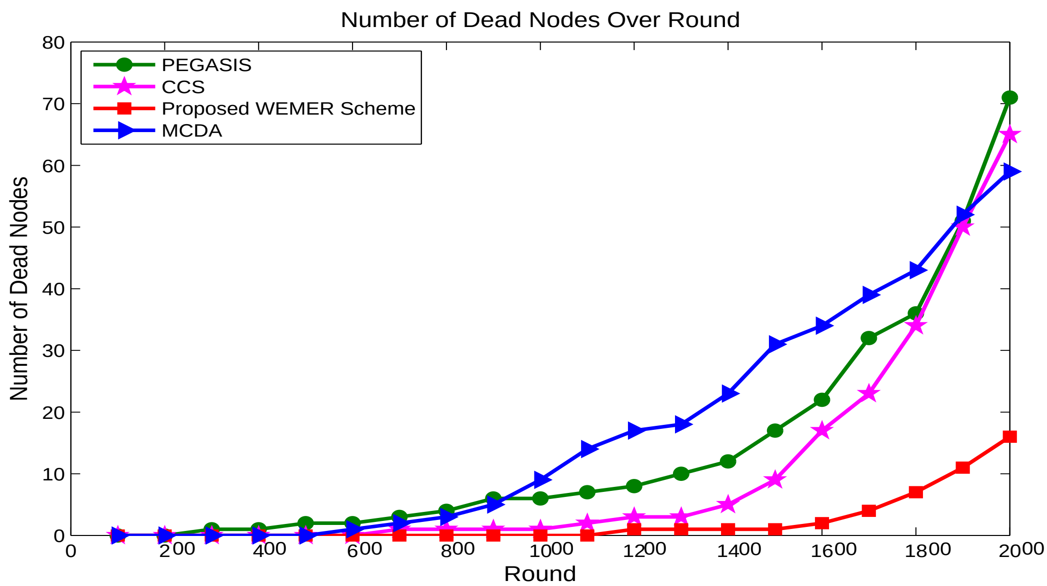


Figure 4.7: Number of dead nodes over round when data packet length is 2000

PEGASIS and CCS is a chain based protocol that transmit data by using chain. But they used greedy algorithm to construct a chain that suffer from LL problem. Also leader node selection of PEGASIS and CCS is random that make low energy node to select as HN. MCDA is free from LL but as its layer 1 node direct communicate with the BS so the nodes which are in the boundary deplete their energy very fast than other node. Chain construction procedure of WEMER is an enhanced of greedy algorithm that reduces the distance and LL among sensor nodes. Also Head Node (HN) selection depend on distance between successor HN or BS and energy, so all nodes get the chance to be HN.Chain construction procedure of WEMER is so efficient that reduces the distance and long link among sensor nodes.

4.2.4 Percentage of dead node

The number of dead nodes specifies the level of the network lifetime. If it is high in most time of the simulation process, the network lifetime is low; and vice versa. Figure 4.8 and 4.9 are showing percentage of dead nodes of the network over round by changing data packet length. The x-axis shows the percentage whereas y-axis shows the number of rounds when first, ten percent, thirty percent, fifty percent and all node died in the network. PEGASIS and CCS protocol construct chain by using greedy method so it creates Long Link (LL) between nodes during chain

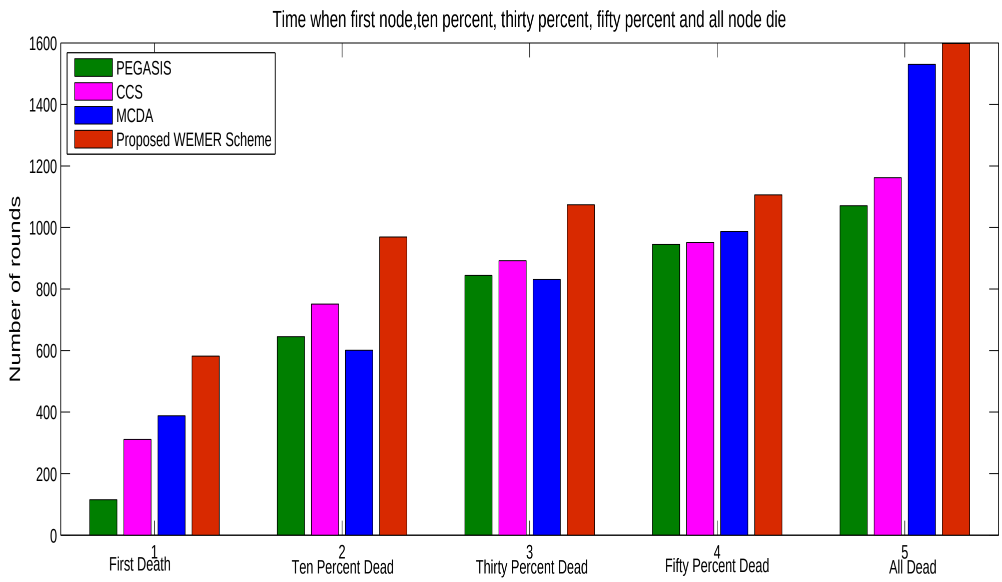


Figure 4.8: Percentage of dead node when data packet length is 2000

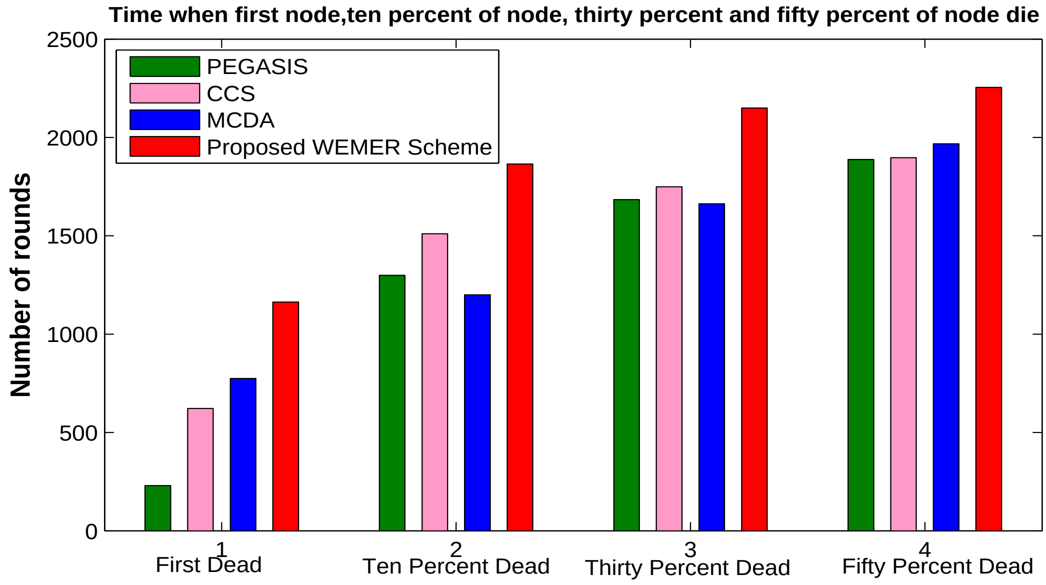


Figure 4.9: Percentage of dead node when data packet length is 4000

construction, so that nodes deplete energy very fast and die early. Though MCDA is free from LL but as its layer 1 node direct communicate with the BS so the nodes which are in the boundary deplete their energy very fast than other node. Chain construction procedure of WEMER is so efficient that reduces the distance and long link among sensor nodes. For PEGASIS, CCS, MCDA first node die at round 115, 311, 388 but for WEMER first node die at round 582. Fifty percent and all node die for WEMER at round 1128 and 1598, whereas for PEGASIS, CCS, MCDA fifty percent and all node die at round 945, 951, 987 and 1071, 1162, 1530.

4.2.5 Average energy cost of a node for a round

When a node energy cost for a round for its transmission become low than lifetime of the network will be maximize. Figure 4.10 and 4.11 are showing average energy cost for a node over round. Study the performance by changing the value of data packet length.The x-axis shows round whereas y-axis shows average energy cost of a node. If average energy cost of a node is minimum than it will help to eliminate or delay the energy hole for a network.In case of average energy cost WEMER perform better than any other protocol. Chain construction procedure of WEMER is so efficient that reduces the distance and long link among sensor nodes.

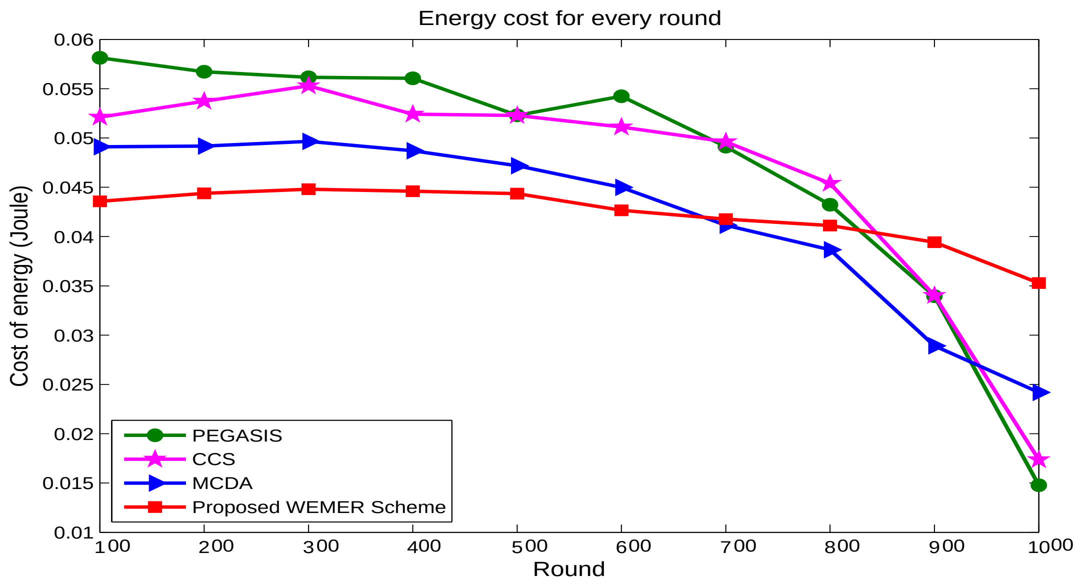


Figure 4.10: Average energy cost over round when data packet length is 4000

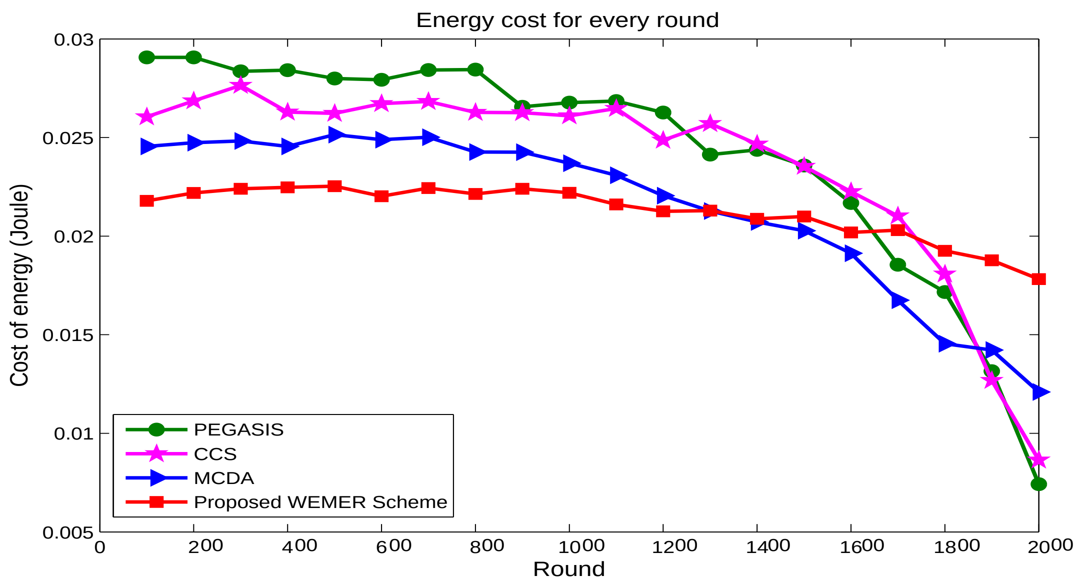


Figure 4.11: Average energy cost over round when data packet length is 2000

4.3 Result after delay efficient approach

In this section WEMER with delay efficient approach will be compared with existing protocols PEGASIS, CCS and MCDA.

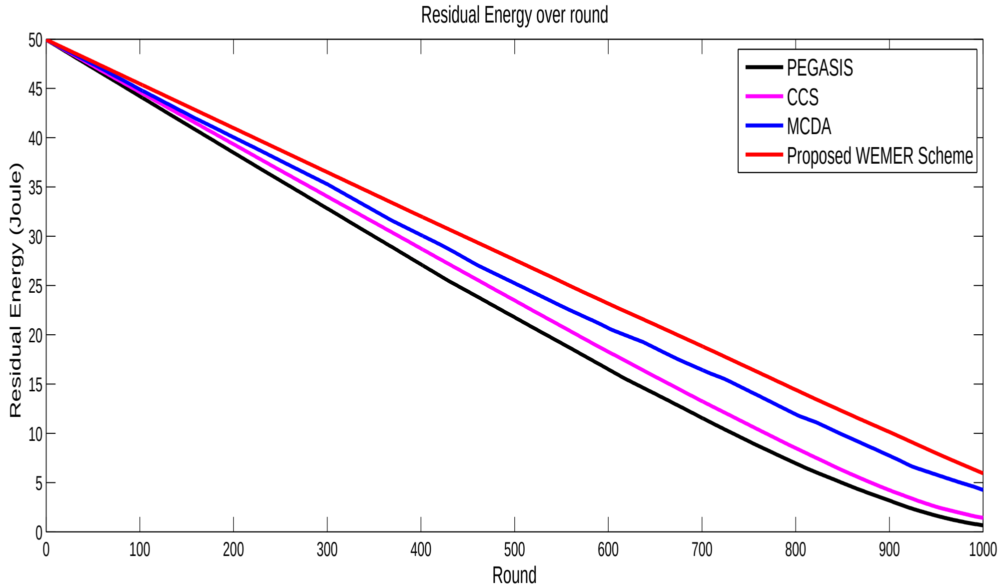


Figure 4.12: Residual energy of network with respect to round

4.3.1 Residual energy of network per round

Remaining energy of the sensor nodes after round effects on the network lifetime directly. If remaining energy of the nodes is high, the network lifetime will be high; and vice versa. Figure 4.12 demonstrates the performance comparison in terms of residual energy of the network with respect to round. After round 1000 remaining energy of nodes for WEMER were 7.1101J whereas after adding new delay efficient technique it is 5.9376J. As for some situation more than one HN is selected for a sector and know that HN is responsible for forward nodes data to lower level sector, so its consume more energy.

4.3.2 Number of alive nodes over round

The number of alive nodes specifies the level of the network lifetime. If it is high in most time of the simulation process, the network lifetime is high; and vice versa. Figure 4.13 is showing number of alive nodes of the network over round. The x-axis shows the round whereas y-axis shows the number of alive nodes in the network. After round 1000 alive nodes for WEMER were 84 whereas after adding new delay efficient technique it is 81.

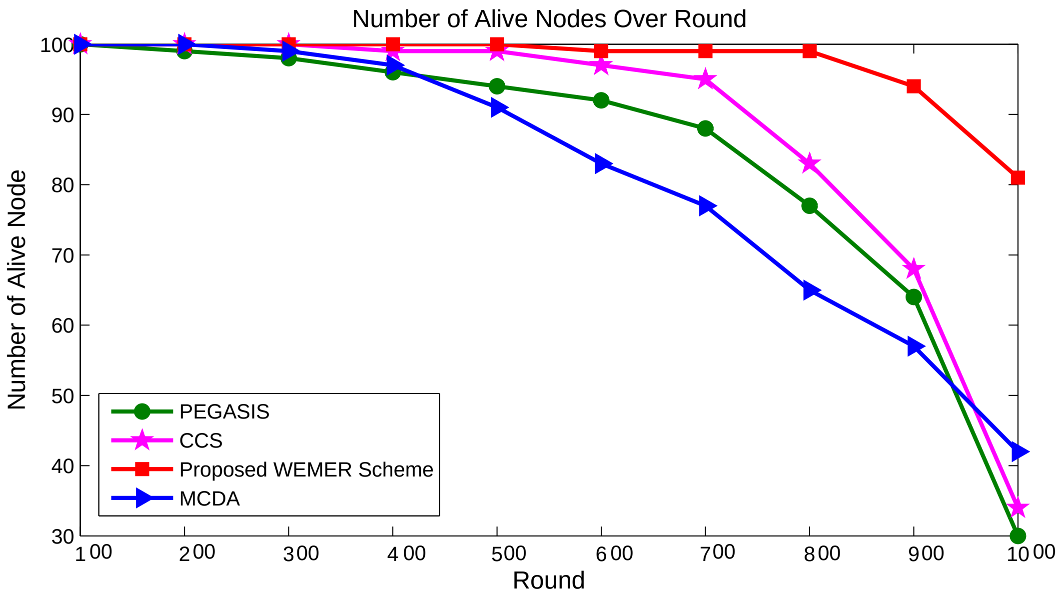


Figure 4.13: Number of alive nodes over round

4.3.3 Number of dead nodes over round

The number of dead nodes specifies the level of the network lifetime. If it is high in most time of the simulation process, the network lifetime is low; and vice versa. Figure 4.14 is showing number of dead nodes of the network over round. The x-axis shows the round whereas y-axis shows the number of dead nodes in the network. After round 1000 dead nodes for WEMER were 16 whereas after adding new delay efficient technique it is 19.

4.3.4 Percentage of dead node

The number of dead nodes specifies the level of the network lifetime. If it is high in most time of the simulation process, the network lifetime is low; and vice versa. Figure 4.15 is showing percentage of dead nodes of the network over round. The x-axis shows the percentage whereas y-axis shows the rounds when first, ten percent, thirty percent and fifty percent of node died in the network. First node died at round 582 and ten percent, thirty percent, fifty percent node died at round 969, 1074, 1106 and 1556.

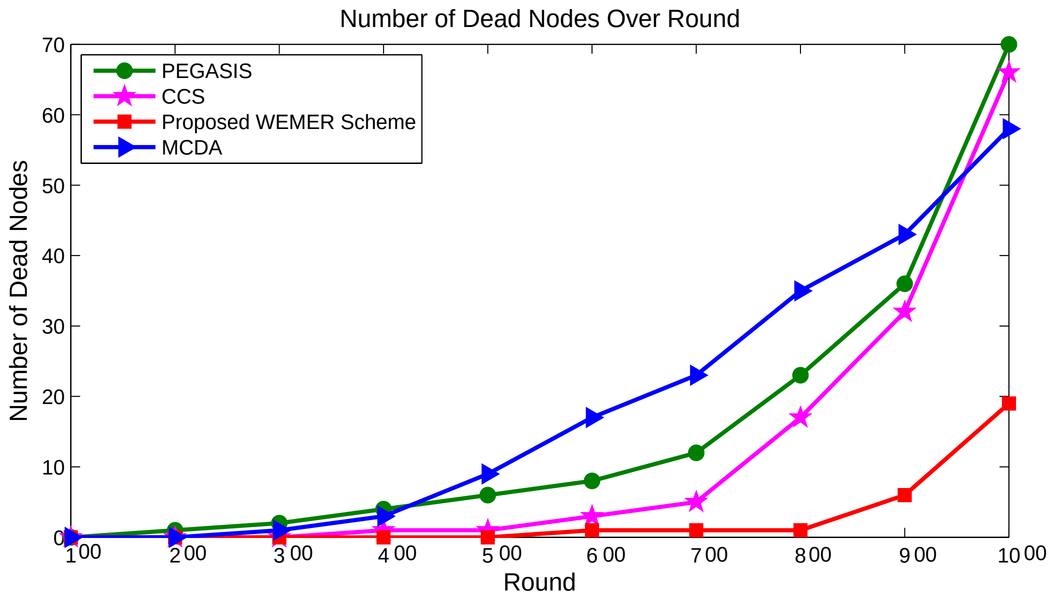


Figure 4.14: Number of dead nodes over round

4.3.5 Average energy cost of a node for a round

When a node energy cost for a round for its transmission become low than lifetime of the network will be maximize. Figure 4.16 is showing average energy cost for a node over round. The x-axis shows round whereas y-axis shows average energy cost of a node. If average energy cost of a node is minimum than it will help to eliminate or delay the energy hole for a network.

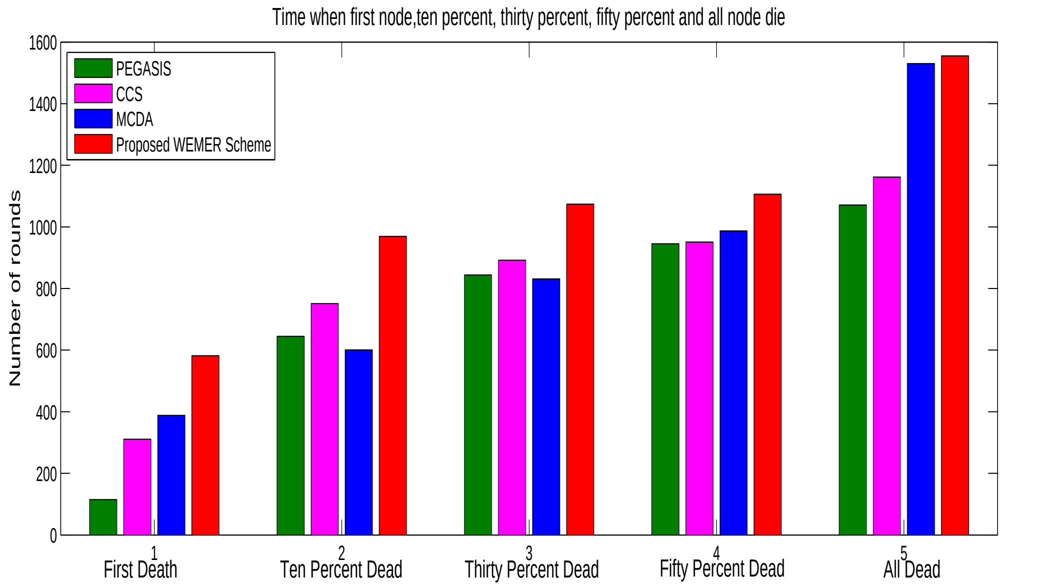


Figure 4.15: Percentage of dead nodes over round

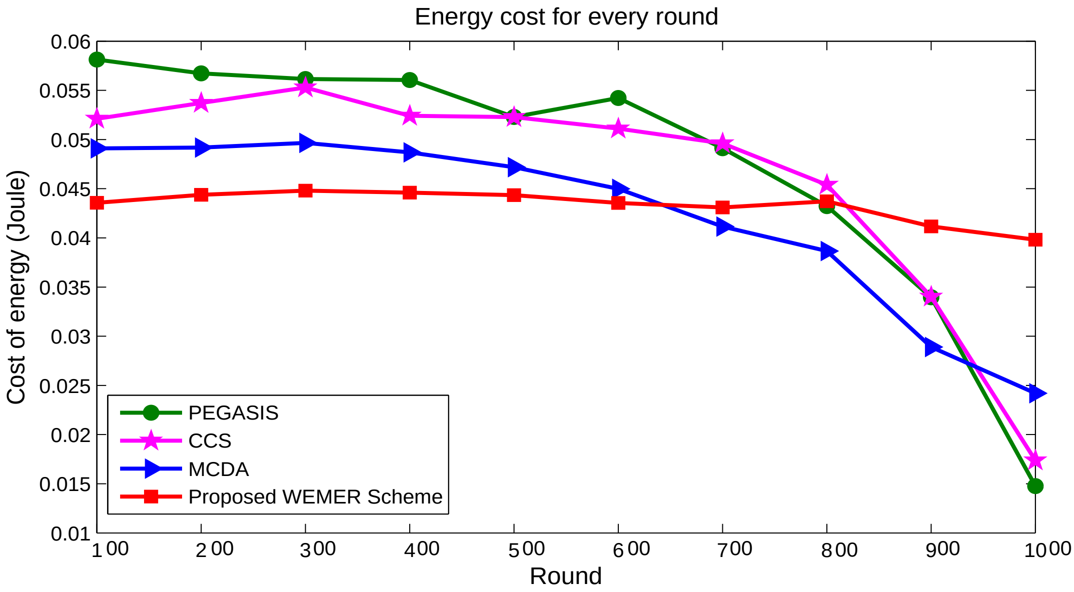


Figure 4.16: Average energy cost over round

4.4 Summary

Proposed WEMER technique is compared with existing PEGASIS, CCS and MCDA protocol in terms of network lifetime, energy cost and energy consumption throughout this chapter. The protocol has been tested on an energy consumption model that can be viewed as a realistic network. The analysis shows that the proposed WEMER scheme performs better than existing PEGASIS, CCS and MCDA protocol by increasing network lifetime.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Though definition of the network lifetime differs according to the protocol, but it is the most important issues in the WSN. Therefore, network lifetime mainly depends on number of active nodes and their connectivity. A sensor node will be disconnected from the network if its energy depleted and died. In order to ensure the network operation, energy must be used in an efficient way. If any node of a sensor network is died the network is no longer fully operational and it make degradation of network lifetime. Battery is the only power source for nodes, however, batteries cannot be replaced easily as they are deployed in an unattended areas.

Extending the lifetime of networks is one of the most significant challenges in designing WSNs. Network lifetime mainly depends on several factors like how network protocols, energy model and lifetime is defined. Unbalanced energy consumption among sensor nodes caused energy hole in the sensor network. When an energy hole appears, even though most of the sensors still have energy data cannot be sent from other sensors to the sink; makes sink isolation. This means that the network lifetime ends prematurely, and a considerable amount of energy is wasted. In order to maximize network lifetime it is necessary to prevent energy hole in the network.

In WEMER balanced energy consumption is achieved by dividing the network into corona and wedges which makes the size of sector small specially the sector which is closer to the BS. In head node selection phase, WEMER considers both the residual energy of nodes and the distance between node and BS as parameters to select more suitable head node and keep energy consumption balanced.. Moreover, for routing chain based communication is used and WEMER used an enhanced version of greedy algorithm to construct chain between nodes that also makes low energy consumption of nodes. Proposed wedge merging technique prevent the network from energy hole problem. In MATLAB simulation proposed WEMER scheme is compared with existing PEGASIS, CCS and MCDA. Simulation results show that WEMER performs better than the existing works, in terms of network lifetime, energy cost, balanced energy consumption and number of node died. For PEGASIS, CCS, MCDA first node die at round 115, 311, 388 but for WEMER first node die at round 582. Fifty percent of node die for WEMER at round 1128 whereas for PEGASIS, CCS, MCDA fifty percent of node die at round 945, 951, 987.

5.2 Future Work

Future work includes the performance evaluation through real world hardware experiment, investigation of more suitable chain construction strategy and optimal number of corona and wedge according to network status and node distribution strategy. An analytical model will be developed to evaluate performance of the protocol. .

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and systems magazine*, vol. 5, no. 3, pp. 19–31, 2005.

[3] "A typical Wireless Sensor Network." http://elb105.com/pfc-design-and-implementation-of-a-dynamic-wireless-sensor-network/.

[4] "Mobility in Wireless Sensor Networks." https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwjBt7zI343VAhUKOrwKHTOoA-YQFggwMAI&url=http%3A%2F%2Fwww.ece.neu.edu%2Ffac-ece%2Fbasagni%2Fpapers%2FBasagniCP08.pdf&usg=AFQjCNEqyLzn46UMQ13Tav5Sd_pNe8f7Gg.

[5] M. Perillo, Z. Cheng, and W. Heinzelman, "An analysis of strategies for mitigating the sensor network hot spot problem," in *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*, pp. 474–478, IEEE, 2005.

[6] J. W. Jung and M. A. Ingram, "Residual-energy-activated cooperative transmission (react) to avoid the energy hole," in *Communications Workshops (ICC), 2010 IEEE International Conference on*, pp. 1–5, IEEE, 2010.

[7] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pp. 1–12, IEEE, 2006.

[8] R. Sharma, "Energy holes avoiding techniques in sensor networks: A survey," *International Journal of Engineering Trends and Technology*, vol. 20, no. 4, pp. 204–208, 2015.

[9] X. Wu, G. Chen, and S. K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE Transactions on parallel and distributed systems*, vol. 19, no. 5, pp. 710–720, 2008.

[10] H. Asharioun, H. Asadollahi, T.-C. Wan, and N. Gharaei, "A survey on analytical modeling and mitigation techniques for the energy hole problem in corona-based wireless sensor network," *Wireless Personal Communications*, vol. 81, no. 1, pp. 161–187, 2015.

[11] H. A. Marhoon, M. Mahmuddin, and S. A. Nor, "Chain-based routing protocols in wireless sensor networks: a survey," *ARPN J Eng Appl Sci*, vol. 10, no. 3, pp. 1389–1398, 2015.

[12] H. Al-Hasan, M. Qatawneh, A. Sleit, and W. Almobaideen, "Eaphrn: energy-aware pegasis-based hierarchal routing protocol for wireless sensor networks," *Journal of American Science*, vol. 7, no. 8, pp. 753–758, 2011.

[13] F. Sen, Q. Bing, and T. Liangrui, "An improved energy-efficient pegasis-based protocol in wireless sensor networks," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference On*, vol. 4, pp. 2230–2233, IEEE, 2011.

[14] Y. C. Yu and G. Wei, "An improved pegasis algorithm in wireless sensor network," *Acta Electronica Sinica*, vol. 36, no. 7, pp. 1309–1313, 2008.

[15] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1567–1576, IEEE, 2002.

[16] T. Van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, pp. 171–180, ACM, 2003.

[17] L. Guntupalli, J. Lin, M. A. Weitnauer, and F. Y. Li, "ACT-MAC: an asynchronous cooperative transmission mac protocol for wsns," in *Communications Workshops (ICC), 2014 IEEE International Conference on*, pp. 848–853, IEEE, 2014.

[18] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The evolution of mac protocols in wireless sensor networks: A survey," *IEEE communications surveys & tutorials*, vol. 15, no. 1, pp. 101–120, 2013.

[19] H. Jiao, M. A. Ingram, and F. Y. Li, "A cooperative lifetime extension mac protocol in duty cycle enabled wireless sensor networks," in *Military Communications Conference, 2011-MILCOM 2011*, pp. 896–901, IEEE, 2011.

[20] A. El-Hoiydi and J.-D. Decotignie, "Wisemac: An ultra low power mac protocol for multi-hop wireless sensor networks," in *ALGOSENSORS*, vol. 4, pp. 18–31, Springer, 2004.

[21] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 307–320, ACM, 2006.

[22] B. Jang, J. B. Lim, and M. L. Sichitiu, "AS-MAC: An asynchronous scheduled MAC protocol for wireless sensor networks," in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pp. 434–441, IEEE, 2008.

[23] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 1–14, ACM, 2008.

[24] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "EM-MAC: a dynamic multichannel energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, p. 23, ACM, 2011.

[25] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *INFOCOM, 2011 Proceedings IEEE*, pp. 1305–1313, IEEE, 2011.

[26] M. Doudou, M. Alaei, D. Djenouri, J. M. Barcelo-Ordinas, and N. Badache, "Duo-mac: energy and time constrained data delivery mac protocol in wireless sensor networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pp. 424–430, IEEE, 2013.

[27] H. Lu, J. Li, and G. Wang, "A novel energy efficient routing algorithm for hierarchically clustered wireless sensor networks," in *Frontier of Computer Science and Technology, 2009. FCST'09. Fourth International Conference on*, pp. 565–570, IEEE, 2009.

[28] Y. Chen, Q. Li, L. Fei, and Q. Gao, "Mitigating energy holes in wireless sensor networks using cooperative communication," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pp. 857–862, IEEE, 2012.

[29] J. Yu, Y. Qi, G. Wang, and X. Gu, "A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution," *AEU-International Journal of Electronics and Communications*, vol. 66, no. 1, pp. 54–61, 2012.

[30] A. Jain and B. Reddy, "Sink as cluster head: An energy efficient clustering method for wireless sensor networks," in *Data Mining and Intelligent Computing (ICDMIC), 2014 International Conference on*, pp. 1–6, IEEE, 2014.

[31] F. Bajaber and I. Awan, "Adaptive decentralized re-clustering protocol for wireless sensor networks," *Journal of Computer and System Sciences*, vol. 77, no. 2, pp. 282–292, 2011.

[32] P. K. Batra and K. Kant, "Leach-mac: a new cluster head selection algorithm for wireless sensor networks," *Wireless Networks*, vol. 22, no. 1, p. 49, 2016.

[33] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks, mobile adhoc and sensor systems conference, 2005," in *IEEE International Conference on*, pp. 7–7, 2005.

[34] M. S. Gharajeh, "Avoidance of the energy hole in wireless sensor networks using a layered-based routing tree," *International Journal of Systems, Control and Communications*, vol. 7, no. 2, pp. 116–131, 2016.

[35] R. Banerjee and C. K. Bhattacharyya, "Energy efficient routing and bypassing energy-hole through mobile sink in wsn," in *Computer Communication and Informatics (ICCCI), 2014 International Conference on*, pp. 1–6, IEEE, 2014.

[36] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 828–854, 2017.

[37] N. Wang, Y. Zhou, and W. Xiang, "An energy efficient clustering protocol for lifetime maximization in wireless sensor networks," in *Global Communications Conference (GLOBECOM), 2016 IEEE*, pp. 1–6, IEEE, 2016.

[38] M. M. Zanjireh and H. Larijani, "A survey on centralised and distributed clustering routing algorithms for wsns," in *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, pp. 1–6, IEEE, 2015.

[39] S. Rani, J. Malhotra, and R. Talwar, "Energy efficient chain based cooperative routing protocol for wsn," *Applied Soft Computing*, vol. 35, pp. 386–397, 2015.

[40] C. Song, M. Liu, J. Cao, Y. Zheng, H. Gong, and G. Chen, "Maximizing network lifetime based on transmission range adjustment in wireless sensor networks," *Computer Communications*, vol. 32, no. 11, pp. 1316–1325, 2009.

[41] R. Logambigai, K. Thanigaivelu, and K. Murugan, "Extending network lifetime using optimal transmission range in wireless sensor networks," in *International Conference on Wireless Communications, Networking and Mobile Computing*, 2012.

[42] K. Thanigaivelu and K. Murugan, "K-level based transmission range scheme to alleviate energy hole problem in wsn," in *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, pp. 476–483, ACM, 2012.

[43] A. Azad and J. Kamruzzaman, "Energy-balanced transmission policies for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 7, pp. 927–940, 2011.

[44] S. Jabbar, A. A. Minhas, A. Paul, and S. Rho, "Multilayer cluster designing algorithm for lifetime improvement of wireless sensor networks," *The journal of Supercomputing*, vol. 70, no. 1, pp. 104–132, 2014.

[45] T. Liu, J. Peng, J. Yang, G. Chen, and W. Xu, "Avoidance of energy hole problem based on feedback mechanism for heterogeneous sensor networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 6, p. 1550147717713625, 2017.

[46] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*, pp. 10–pp, IEEE, 2000.

[47] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," in *Aerospace conference proceedings, 2002. IEEE*, vol. 3, pp. 3–3, IEEE, 2002.

[48] S.-M. Jung, Y.-J. Han, and T.-M. Chung, "The concentric clustering scheme for efficient energy consumption in the pegasis," in *Advanced Communication Technology, The 9th International Conference on*, vol. 1, pp. 260–265, IEEE, 2007.

[49] S. Jabbar, A. A. Minhas, A. Paul, and S. Rho, "Multilayer cluster designing algorithm for lifetime improvement of wireless sensor networks," *The journal of Supercomputing*, vol. 70, no. 1, pp. 104–132, 2014.

[50] V. Cionca, T. Newe, and V. Dadârlat, "Tdma protocol requirements for wireless sensor networks," in *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on*, pp. 30–35, IEEE, 2008.

[51] P. K. Pal and P. Chatterjee, "A survey on tdma-based mac protocols for wireless sensor network," *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 6, pp. 219–230, 2014.

[52] A. Y. Barnawi, "Adaptive tdma slot assignment using request aggregation in wireless sensor networks," *Procedia Computer Science*, vol. 10, pp. 78–85, 2012.

[53] J. Zheng and A. Jamalipour, *Wireless sensor networks: a networking perspective*. John Wiley & Sons, 2009.

[54] S. Das, I. Banerjee, M. Chatterjee, and T. Samanta, "Performance analysis of tdma based data transmission in wsn," in *Intelligent Systems Design and Applications (ISDA), 2014 14th International Conference on*, pp. 107–112, IEEE, 2014.

[55] A. N. Alvi, S. H. Bouk, S. H. Ahmed, M. A. Yaqub, N. Javaid, and D. Kim, "Enhanced tdma based mac protocol for adaptive data control in wireless sensor networks," *Journal of communications and networks*, vol. 17, no. 3, pp. 247–255, 2015.

[56] K.-H. Chen, J.-M. Huang, and C.-C. Hsiao, "Chiron: an energy-efficient chain-based hierarchical routing protocol in wireless sensor networks," in *Wireless Telecommunications Symposium, 2009. WTS 2009*, pp. 1–5, IEEE, 2009.

[57] "MathWorks - Makers of MATLAB and Simulink." `https://www.mathworks.com`.

Appendix A

Appendix A

Partial Source Code of Simulation Model

main.m

```
[General]
IniEng=0.5; % Initial Energy of Every Node
NoOfNode=100; % Number of Node
NoOfRound=1000; % Number of Round
total_circle = (NoOfRound/100);
data_pkt_length=4000;
ETX=50*0.000000001;
ERX=50*0.000000001;
Efs=100*0.000000000001; %Transmit Amplifier types
Emp=0.0013*0.000000000001;
EDA=5*0.000000001; %Data Aggregation Energy
```

WEMER.m

```
%for finding nodes wedge and corona
cor_x = 160; %3rd point for find angle with sink
cor_y = 80;  %3rd point for find angle with sink
pos_corona = 0;
pos_wedge = 0;

for i=1:1:NoOfNode
    To_sink_dist(i) = sqrt((Node.x(i)-Bx)^2+(Node.y(i)-By)^2);
    pos_corona = ceil(To_sink_dist(i)/38); %change according to corona circle
    Node.corona_pos(i) = pos_corona;

    node_angle=(atan2( abs((cor_x-Bx)*(Node.y(i)-By)-(cor_y-By)*(Node.x(i)-Bx)),
                (cor_x-Bx)*(Node.x(i)-Bx)+(cor_y-By)*(Node.y(i)-By) ))* 180/pi;;

    if(Node.y(i)< By)  %if nodes angle with BS is greater than 180
        node_angle = 180 + (180- node_angle);
    end
```

```
    pos_wedge = ceil(node_angle/30);
    if(pos_wedge == 0)
        pos_wedge = 1;
    end
    Node.wedge_pos(i) = pos_wedge;
    index_val = Node.corona_wedge_index(pos_corona,pos_wedge);
    Node.corona_wedge_index(pos_corona,pos_wedge) = index_val + 1;
    Node.corona_wedge_nodes_id(pos_corona,pos_wedge,index_val + 1)= i;
end
%end of finding nodes wedge and corona

% Getting Head Node for a sector
head_id= linspace(0,0,wedge_number);
for i=1:1:corona_number
for j=1:1:wedge_number
if(Node.corona_wedge_index(i,j)>1)
min_timer = 1000;
index_no = Node.corona_wedge_index(i,j);
avg_energy = Node.corona_wedge_avg_energy(i,j);
for k=1:1:index_no
node_id = Node.corona_wedge_nodes_id(i,j,k);

if Node.E(node_id)>= avg_energy     %.4*En
if(i==1)
    timer = To_sink_dist(node_id);
else
    timer = sqrt((Node.x(node_id)-Node.x(head_id(i-1,j)))^2+
                 (Node.y(node_id)-Node.y(head_id(i-1,j)))^2);
end
    timer = timer + 1/Node.E(node_id);
if(min_timer>timer)
head_id(i,j)= node_id;
Node.min_dis_node(i,j) = node_id;
min_timer = timer;
end
end
end
end
end
end
% End of getting Head Node for a sector

% For getting maximum distance node from a Head Node for a sector
Node.max_distance_node_id=zeros(corona_number,wedge_number);
Node.already=linspace(0,0,NoOfNode);

for i=1:1:corona_number
for j=1:1:wedge_number
```

```
if(Node.corona_wedge_index(i,j)>1)
max_distance = 1;
head_index_no = head_id(i,j);
index_no = Node.corona_wedge_index(i,j);
for k=1:1:index_no
node_id = Node.corona_wedge_nodes_id(i,j,k);
if(Node.E(node_id)>0)
if(head_index_no~=node_id)
    Node.to_head_node_distance(node_id)=
                        sqrt((Node.x(node_id)-Node.x(head_index_no))^2+
                        (Node.y(node_id)-Node.y(head_index_no))^2);
if(max_distance < Node.to_head_node_distance(node_id))
    Node.max_distance_node_id(i,j)= node_id;
    max_distance = Node.to_head_node_distance(node_id);
end
end
end
end
    Node.already(Node.max_distance_node_id(i,j)) = 1;
end
end
end
% end of getting maximum distance node from a Head Node for a sector

% For creating the chain in a sector
for i=1:1:corona_number
for j=1:1:wedge_number
first_cor_wed = 0;
chain_node_no = 1;
if(Node.corona_wedge_index(i,j)>1)
index_no = Node.corona_wedge_index(i,j);
tracker = Node.corona_wedge_index(i,j);
max_distance_node_id = Node.max_distance_node_id(i,j);
max_distance_node = max_distance_node_id;
Node.chain_form_node(chain_node_no) = max_distance_node;

while(tracker~=0)

min_distance = 1000;
dis_with_min = 1000;

for k=1:1:index_no
node_id = Node.corona_wedge_nodes_id(i,j,k);
if(Node.E(node_id)>0)
if(max_distance_node ~= node_id && Node.already(node_id)==0)
    Node.distance_frm_max_distance_node(node_id)=
                    sqrt((Node.x(node_id)-Node.x(max_distance_node))^2+
                    (Node.y(node_id)-Node.y(max_distance_node))^2);
```

```
if(min_distance > Node.distance_frm_max_distance_node(node_id)&&
                            Node.distance_frm_max_distance_node(node_id)~=0)
      min_distance = Node.distance_frm_max_distance_node(node_id);
      min_node = node_id;
end
end
end
end
%new portion of chain construction
for k=1:1:chain_node_no
if(first_cor_wed == 0)
    chain_node = Node.chain_form_node(k);
    dis_min = sqrt((Node.x(chain_node)-Node.x(min_node))^2+
                   (Node.y(chain_node)-Node.y(min_node))^2);
if(dis_min < dis_with_min)
dis_with_min = dis_min;
min_chain_node = chain_node;
end
end
end
%end of new portion of chain construction

if(max_distance_node ~= min_node)
    min_dis_node_id = min_node;
    Node.pri(max_distance_node) = min_dis_node_id;
    Node.chain_pri(min_node) = min_chain_node;
    Node.to_pri_dis(max_distance_node)=min_distance;
    Node.to_chain_pri_dis(min_node)=dis_with_min;
    Node.already(min_node)=1;
    chain_node_no = chain_node_no + 1;
    Node.chain_form_node(chain_node_no) = min_node;
    max_distance_node = min_dis_node_id;

    connect_node_count = Node.chain_connected_node_no(min_chain_node);
    Node.chain_connected_node_no(min_chain_node) = connect_node_count+1;
    Node.connected_node_ids(min_chain_node,connect_node_count+1) = min_node;

end
tracker = tracker -1;
end %end of while track
Node.tail_node_id(i,j)= min_node; % for getting tail node id for a sector
end
end
end
% end of creating the chain in a sector

%energy consumption for every round
Node.send_dis=linspace(0,0,NoOfNode);
```

```
Node.hn_route=linspace(1,1,NoOfNode);
Node.not_hn_route_node_no=linspace(0,0,NoOfNode);
for i=1:1:corona_number
for j=1:1:wedge_number
if(Node.corona_wedge_index(i,j)>1)
index_no = Node.corona_wedge_index(i,j);
head_node = Node.min_dis_node(i,j);
max_distance_node = Node.max_distance_node_id(i,j);

for k=1:1:index_no
node_id = Node.corona_wedge_nodes_id(i,j,k);
if(Node.E(node_id)>0)
if(node_id~=max_distance_node && node_id~=head_node)
    connect_node_count = Node.chain_connected_node_no(node_id);
if(connect_node_count == 0)
    hn = Node.chain_pri(node_id);
if(hn==head_node)
    Node.hn_route(node_id) = 1;
else
    total_not_hn = Node.not_hn_route_node_no(hn) + 1;
    Node.not_hn_route_node_no(hn) = total_not_hn;
    total_connected_node = Node.chain_connected_node_no(hn);
    no_node_not_chck = total_connected_node - total_not_hn;
    Node.hn_route(node_id) = 0;
while(no_node_not_chck ==0)
    h_node = Node.chain_pri(hn);
if(h_node==head_node)
    no_node_not_chck = 1;
else
    Node.hn_route(hn) = 0;
    total_not_hn = Node.not_hn_route_node_no(h_node) + 1;
    Node.not_hn_route_node_no(h_node) = total_not_hn ;
    total_connected_node = Node.chain_connected_node_no(h_node);
    no_node_not_chck = total_connected_node - total_not_hn;
end
end
end
end %end of if(connect_node_count == 0)
end
end
end
end
end
end

Node.send_dis = linspace(0,0,NoOfNode);
Node.chain_path = linspace(0,0,NoOfNode);
for i=1:1:corona_number
```

```
for j=1:1:wedge_number

if(Node.corona_wedge_index(i,j)>1)
index_no = Node.corona_wedge_index(i,j);
head_node = Node.min_dis_node(i,j);
max_distance_node = Node.max_distance_node_id(i,j);
tail_node =  Node.tail_node_id(i,j);
node_to_send = max_distance_node;
node_stage = 0;  % 1 = head node found but not tail node,
                 % 2 = node which are after head node, 3 = tail node found

while node_stage~=3
if(node_stage==0)
node_to_send = Node.pri(node_to_send);% node which gets selected by maximum
                                      % distance node to construct a chain
start_node = Node.chain_pri(node_to_send);
if(Node.hn_route(node_to_send) == 0)
Node.send_dis(node_to_send)=Node.to_chain_pri_dis(node_to_send);
Node.chain_path(node_to_send)= start_node;
else
Node.send_dis(start_node)=Node.to_chain_pri_dis(node_to_send);
Node.chain_path(start_node)= node_to_send;
end
if(node_to_send == head_node)
if(tail_node == head_node)
if(i==1)
Node.send_dis(node_to_send)=To_sink_dist(node_to_send);
else
Node.send_dis(node_to_send)=sqrt((Node.x(node_to_send)-Node.x(head_id(i-1,j)))^2
                              +(Node.y(node_to_send)-Node.y(head_id(i-1,j)))^2);
end
node_stage = 3;
else
node_stage = 1;
end
end
elseif(node_stage==1)
if(i==1)
Node.send_dis(node_to_send)=To_sink_dist(node_to_send);
else
Node.send_dis(node_to_send)=sqrt((Node.x(node_to_send)-Node.x(head_id(i-1,j)))^2
                            +   (Node.y(node_to_send)-Node.y(head_id(i-1,j)))^2);
end
node_stage = 2;
else
node_to_send = Node.pri(node_to_send);
start_node = Node.chain_pri(node_to_send);
Node.send_dis(node_to_send)=Node.to_chain_pri_dis(node_to_send);
```

```
Node.chain_path(node_to_send)= start_node;
if(tail_node == node_to_send)
node_stage = 3;
end
end
end
end
end
end

for i=1:1:corona_number
for j=1:1:wedge_number
if(Node.corona_wedge_index(i,j)>0)

index_no = Node.corona_wedge_index(i,j);
head_node = Node.min_dis_node(i,j);
for k=1:1:index_no
    node_id = Node.corona_wedge_nodes_id(i,j,k);
    total_travel_distance = total_travel_distance + Node.send_dis(node_id);
if(Node.E(node_id)>0)
if(node_id == head_node)
    dis_with_hn  = Node.send_dis(node_id);
if dis_with_hn > do
    Node.E(node_id) = Node.E(node_id)-(ETX*data_pkt_length+
                      Emp*(dis_with_hn.^4)*data_pkt_length);
else
Node.E(node_id) = Node.E(node_id)-(ETX*data_pkt_length+Efs*(dis_with_hn.^2)*
                                   data_pkt_length);
end
else
dis_with_forwarder = Node.send_dis(node_id);
forwarding_node = Node.chain_path(node_id);
if dis_with_forwarder>do
Node.E(node_id)=Node.E(node_id)-(ETX*data_pkt_length+Emp*(dis_with_forwarder.^4)
                                   *data_pkt_length);
else
Node.E(node_id) = Node.E(node_id)-(ETX*data_pkt_length+Emp*(dis_with_forwarder.^4)
                                   *data_pkt_length);
end
if(forwarding_node > 0)
Node.E(forwarding_node)=Node.E(forwarding_node)-(ERX + EDA)*data_pkt_length;
end
end %end of if(node_id == head_node)
end
end

end
end
```

```
end

%end of energy consumption for every round

%Check if merging is needed or not
Node.corona_wedge_avg_energy = zeros(corona_number,wedge_number);
Node.corona_wedge_low_energy_node = zeros(corona_number,wedge_number);
for i=1:1:corona_number
for j=1:1:wedge_number
if(Node.corona_wedge_index(i,j)>1)
TotalNetworkEnergy=0;
AvgEnergy = 0;
low_energy = 1;
low_energy_node = 0;
index_no = Node.corona_wedge_index(i,j);
for k=1:1:index_no
node_id = Node.corona_wedge_nodes_id(i,j,k);
if Node.E(node_id)>0
TotalNetworkEnergy=TotalNetworkEnergy+Node.E(node_id);
if (Node.E(node_id)<low_energy)
low_energy = Node.E(node_id);
low_energy_node = node_id;
end
end
end
AvgEnergy = TotalNetworkEnergy/index_no;
Node.corona_wedge_avg_energy(i,j) = AvgEnergy;
Node.corona_wedge_low_energy_node(i,j) = low_energy_node;

if(AvgEnergy <=(.4*IniEng))

left_wedge = left_neighbor_wedge(j);
right_wedge = right_neighbor_wedge(j);
left_index_no = Node.corona_wedge_index(i,left_wedge);
right_index_no = Node.corona_wedge_index(i,right_wedge);

if(left_index_no >0 || right_index_no>0)
if(left_index_no >0)
for k=1:1:left_index_no
node_id = Node.corona_wedge_nodes_id(i,left_wedge,k);
if Node.E(node_id)>0
left_wedge_TotalNetworkEnergy=left_wedge_TotalNetworkEnergy+Node.E(node_id);
end
end
left_wedge_avg_Energy = left_wedge_TotalNetworkEnergy/left_index_no;
end
if(left_index_no >0)
for k=1:1:right_index_no
```

```
node_id = Node.corona_wedge_nodes_id(i,right_wedge,k);
if Node.E(node_id)>0
right_wedge_TotalNetworkEnergy=right_wedge_TotalNetworkEnergy+Node.E(node_id);
end
end
right_wedge_avg_Energy = right_wedge_TotalNetworkEnergy/right_index_no;
end

if(left_wedge_avg_Energy > right_wedge_avg_Energy)
for cor=1:1:corona_number
low_energy_wedge_node_no = Node.corona_wedge_index(cor,j);
left_wedge_node_no = Node.corona_wedge_index(cor,left_wedge);

for k=1:1:low_energy_wedge_node_no
    node_id = Node.corona_wedge_nodes_id(cor,j,k);
    Node.corona_wedge_nodes_id(cor,j,k) = -1;
    left_wedge_node_no = left_wedge_node_no + 1;
    Node.corona_wedge_index(cor,left_wedge) = left_wedge_node_no;
    Node.corona_wedge_nodes_id(cor,left_wedge,left_wedge_node_no)= node_id;
end

Node.corona_wedge_index(cor,j) = 0;
end
left_neighbor_wedge(right_wedge) = left_wedge;
else
for cor=1:1:corona_number
low_energy_wedge_node_no = Node.corona_wedge_index(cor,j);
right_wedge_node_no = Node.corona_wedge_index(cor,right_wedge);
  for k=1:1:low_energy_wedge_node_no
      node_id = Node.corona_wedge_nodes_id(cor,j,k);
      Node.corona_wedge_nodes_id(cor,j,k) = -1;
      right_wedge_node_no = right_wedge_node_no + 1;
      Node.corona_wedge_index(cor,right_wedge) = right_wedge_node_no;
      Node.corona_wedge_nodes_id(cor,right_wedge,right_wedge_node_no)= node_id;
  end
Node.corona_wedge_index(cor,j) = 0;
end
if(left_wedge~=right_wedge)
right_neighbor_wedge(left_wedge) = right_wedge;
end
end
end %end of check right wedge and left wedge has more than one node
end %end of if(AvgEnergy <=(.4*En))
end
end
end
```