AN ENERGY EFFICIENT WSN WITH EQDC PROTOCOL AND IMPROVED NODE PARTICIPATION

Main Project Report
submitted in partial fulfillment of
the requirements for the award of the degree of
BACHELOR OF TECHNOLOGY

in

Computer Science and Engineering

from

APJ Abdul Kalam Kerala Technological University

Submitted By

ABOOBACKER SIDHEEQUE K (MEA19CS001)

MEA Engineering College

Department of Computer Science and Engineering

Vengoor P.O, Perinthalmanna, Malappuram, Kerala-679325

JUNE 2023

Department of Computer Science and Engineering
MEA ENGINEERING COLLEGE
PERINTHALMANNA-679325

Certificate

*This is to certify that the Project report entitled "***AN ENERGY EFFICIENT WSN WITH EQDC PROTOCOL AND IMPROVED NODE PARTICIPATION***" is a bonafide record of the work done by* **ABOOBACKER SIDHEEQUE K (MEA19 CS001)** *under my supervision and guidance. The report has been submitted in partial fulfillment of the requirement for the award of the Degree of* **Bachelor of Technology** *in* **Computer Science and Engineering** *from the APJ Abdul Kalam Kerala Technological University for the year 2022-2023.*

Prof. SREERAM S
Associate Professor & Dean Academics
Dept.of Computer Science and Engineering
MEA Engineering College

Dr. K. NAJEEB
Head Of Department
Dept. of Computer Science and Engineering
MEA Engineering College

Acknowledgements

An endeavor over a long period may be successful only with advice and guidance of many well wishers. I take this opportunity to express my gratitude to all who encouraged me to complete this project. I would like to express my deep sense of gratitude to our respected Principal **Dr. G. Ramesh** for his inspiration and for creating an atmosphere in the college to do the project.

I would like to thank **Dr. K. Najeeb**, Head of the department, Computer Science and Engineering for providing permission and facilities to conduct the project in a systematic way. I highly indebted to **Prof. Sreeram S**, Associate Professor & Dean Academics in Computer Science and Engineering for guiding me and giving timely advices, suggestions and whole hearted moral support in the successful completion of this project.

My sincere thanks to project co-ordinates **Dr. Shanid Malayil**, Asst. Professor in Computer Science and Engineering, **Prof. Aasha Verghese**, Asst. Professor in Computer Science and Engineering for their wholehearted moral support in completion of this project.

Last but not least, I would like to thank all the teaching and non-teaching staff and my friends who have helped me in every possible way in the completion of this project.

ABOOBACKER SIDHEEQUE K (MEA19CS001)

DATE :

Abstract

In Wireless Sensor Network (WSN), Wireless sensors are backed by their battery life. The battery depletion creates a communication hole in the network. The replacement of certain energy-depleted sensor nodes necessitates their detection. The work in this article presents a novel solution for the detection of coverage hole in the WSN and to provide the major issues, node deployment, and various routing protocols in wireless sensor networks. The major issue in wireless sensor networks is the faster dissipation of battery power, which leads to dead nodes.Full coverage is critical for the applications of surveillance and monitoring in wireless sensor networks(WSNs). However, the emergence of coverage holes is unavoidable generally due to various reasons.By proposed system detection of coverage hole and mitigate issue by improved node participation.Also discusses various node deployment techniques in which nodes are deployed at random. Routing protocols are used to determine an efficient path from nodes to the base station. These protocols are analyzed, and by means of routing method for the WSN that is the combination of energy routing of clustering and optimising leach protocol by distance routing of the Gaussian network model. Thereby increasing the efficiency of the whole network, it becomes possible to increase the lifetime of nodes in the network, network throughput, average residual energy and resulting improved network efficiency.

List of Abbreviations

WSN	Wireless Sensor Network
CDS	Connected dominating set
GAF	Geographic Adaptation Fidelity
ROI	Region Of Interest
DT	Delaunay Triangulation
CGA	Computational Geomrtry Approach
DVHD	Distance Vector Hole Determination
AOG	Angle Of Arrival
GPR	Gaussian Process Regression
GPS	Global Positioning System
CH	Clustering Head
RPC	Remote Procedure Call
DORA	Destination Oriented Routing Algorithm
LIDAR	Light Detection And Ranging
AN	Anchor Node
UN	Unknown Node
QOS	Quality of Service
DOS	Denial of Service
GMM	Gaussian Mixture Model
WUSN	Wireless underground sensor network
CRN	Cognitive Radio Networks
EQDC	Eqi-Quadrant Division
OSPR	Optimized Shortest Path Routing
BS	Base Station
LEACH	Low-Energy Adaptive Clustering Hierarchy

List of Algorithms

List of Tables

List of Figures

Contents

CHAPTER 1

INTRODUCTION

A wireless sensor network (WSN) is a wireless network that contains distributed independent sensor devices that are meant to monitor physical or environmental conditions. A WSN consists of a set of connected tiny sensor nodes, which communicate with each other and exchange information and data. These nodes obtain information on the environment such as temperature, pressure, humidity or pollutant, and send this information to a base station.The latter sends the information to a wired network or activates an alarm or an action, depending on the type and magnitude of data monitored.

Random distribution of nodes in Wireless Sensor Networks (WSNs) means that the sensor nodes are deployed randomly within the network area without following any specific pattern or placement strategy. In other words, the nodes are placed at random locations, and their positions are not predetermined, but here the network area for distribution is predefined.Random deployment can be used to scale the network to cover a larger area, as it does not require precise node placement or pre-existing infrastructure, can help reduce the cost and time of deploying the network, as it does not require extensive planning or surveying of the deployment area, also helps to make the network more robust and resilient to node failures, as the random distribution can help distribute the workload and ensure redundancy.On the otherhand, disadvantages includes not guarantee full coverage of the deployment area, and there may be regions that are underserved or overserved, may result in poor connectivity or gaps in the network, which can affect data collection and transmission, can result in inefficient energy consumption, as some nodes may be placed in locations with high traffic or low relevance, leading to unnecessary energy consumption.These can be solved by the proposed system of having coverage hole detection and rectifying issue by means on improved node participation of the movable nodes to the coverage hole area.Resulting of solving all the issues that arises due to random distribution of nodes,ie, guarantee the full coverage of the deployment, no connectivity gaps and lower energy consumption.

Delaunay triangulation can be used to optimize the coverage of a WSN by creating a mesh of triangles that covers the deployment area. This ensures that every point in the network is covered by at least one sensor node, and helps minimize the number of nodes required for full coverage.Can be used to optimize the energy consumption of sensor nodes in WSNs. By ensuring that sensor nodes are not redundantly deployed, Delaunay triangulation can help reduce the number of active nodes and the amount of energy consumed by the network.To optimize the routing efficiency in WSNs by providing a natural framework for the design of routing protocols. The triangles in the mesh can be used to define communication paths between nodes, and can help reduce the distance and number of hops required for data transmission. Scale WSNs to cover larger areas, as it provides a framework for deploying a large number of nodes and optimizing their placement. This makes it possible to monitor larger areas with greater precision and accuracy.It can help improve the fault tolerance of WSNs by providing redundancy and alternate paths for data transmission. If a node fails or becomes unavailable, the mesh can be reconfigured to ensure that data can still be transmitted through alternate routes.Used to find the largest free space inside a network for next deployment target. And determine the optimal sensing coverage radius for each sensor,hence reducing the energy usage.

Clustering protocols are an important class of protocols used in Wireless Sensor Networks (WSNs) to organize sensor nodes into groups, or clusters, based on their spatial proximity, energy levels, or other relevant factors. The goal of clustering is to increase the energy efficiency of the network by reducing the number of nodes that need to transmit data to the base station, which in turn helps to prolong the lifetime of the network.In a clustering protocol, nodes are divided into groups, with each group being led by a designated cluster head (CH). The CH is responsible for aggregating data from the nodes in its group and transmitting it to the base station or to another CH in a higher-level cluster. This helps to reduce the number of nodes that need to transmit data directly to the base station, which in turn reduces the energy consumption of the network.Also by the LEACH protocol is designed to reduce energy consumption in WSNs by organizing the nodes into clusters and rotating the role of cluster head (CH) among the nodes to balance the energy consumption across the network.By organizing the nodes into clusters and rotating the CH role, the LEACH protocol helps to balance the energy consumption across the network, which in turn helps to prolong the lifetime of the network,used in large-scale networks, as it does not require a centralized infrastructure for cluster formation.It is adaptive to changes in the network topology, and can adjust the CH rotation rate based on the energy levels of the nodes.

Also by the proposed system combination of improved node participation by placement of mobale nodes to the network area and energy efficient routing of clustering protocol of EQDC Protocol (Equi-Quadrant Division Clustering) and optimised Shortest Path Routing by means of minimum spanning tree.Able to obtain the shortest distance between nodes in network hence minimum path between a source and base station.Resulting to obtain increasing the network efficiency of the whole Wireless Sensor Network.On Random distribution of nodes into a predefined network area consisting of major issues like coverage hole, sensing range of each node etc... These are solved by improved node participation in the proposed system. Accordingly, the network area divided into equal four square quadrant of having a single cluster head in each quadrant. Also by optimized shortest path routing of minimum spanning tree for obtaining minimum path from each cluster head to base station. The purpose of improved node participation, EQDC and optimized shortest path routing is to increase data reliability, mitigate coverage hole issue, reduce energy consumption for wireless sensor network (WSN) and increase network efficiency. The experimental results of proposed methodology shows that EQDC protocol, enhanced node engagement and OSPR protocol shows the tackle coverage hole challenges, curtail power usage, high data reliability, optimized network productivity.

1.1 Problem Statement

We consider the case where we already have a set of sensors installed in a 2-dimensional area A, by means of random distribution of sensor nodes to a predefined network area but these sensors do not completely cover the field due to issues of coverage hole, dense and sparse distribution of nodes in the network area. The sensing range of a sensor node is defined by radius r, i.e. it can monitor any object that is within a distance of r from it. The problem we consider here is how to efficiently identify the areas that are not covered and how to restore the desired coverage levels with minimal cost.And after rectifying the issue concerned with the distribution of the nodes the next step is to efficiently routing the sensor nodes to the base station and to increase the efficiency of the network by means of giving consideration to the throughput,average residual energy etc of the Wireless Sensor Network.

1.2 Objective

- To mitigate the issue concerned of Coverage hole
- To attain more node participation in the sensor network
- Formation of improved energy routing clustering protocol
- Optimising the leach protocol
- Combination of improved energy routing of clustering protocol and optimised leach protocol
- To find the shortest path from the source node to the base station
- To improve energy efficiency of wireless sensor network using Proposed System

1.3 Scope

Wireless Sensor Networks (WSNs) have a broad range of applications and can be used in various fields.

- **Navigation System**
  Check the position of satellite via GPS which can measure movement.Used to provide indoor positioning and navigation, which is useful in places where GPS signals are weak or unavailable. By deploying wireless sensors in a building, for example, the system can provide location information to users inside the building.Also includes Indoor navigation, Autonomous vehicles, Aviation, Maritime navigation etc.

- **Quality Control System**
  Have a wide range of potential applications in Quality Control Systems (QCS), particularly in industries that require real-time monitoring and control of production processes.It includes Monitoring product quality,used to monitor the quality of products at various stages of production by deploying wireless sensors in the production line.In Real-time process control,used to monitor and control various aspects of the production process in real-time.In Predictive maintenance,can be used to monitor the health of machines and equipment in real-time, and predict when maintenance is required. By deploying wireless sensors on machines and equipment, the system can collect data on various parameters such as vibration, temperature, and power consumption, and analyze the data to identify any signs of wear or damage.

- **Business**
  Technologies used business company to serve their customer.In Supply chain management,used to monitor the location and condition of products during transportation and storage. In Inventory control, used to monitor the level of inventory in real-time, and ensure that the inventory is being stored under the right conditions.In Environmental monitoring, used to monitor environmental conditions in the workplace, such as temperature, humidity, and air quality.

- **Networking**
  A wide range of potential applications in networking, particularly in areas such as data collection, processing, and communication.In Data collection, used to collect data from various sensors and devices, and transmit the data to a central location for processing and analysis.In Data processing, used to process data in real-time, and perform various tasks such as data filtering, compression, and aggregation.In

Communication, used to establish communication between various devices and systems, and enable real-time control and monitoring.In Network managemen, used to manage various aspects of the network, such as routing, energy consumption, and security. By deploying wireless sensors with network management capabilities, the system can optimize the network performance and ensure reliable and secure communication.

- **Management**
  It have a wide range of potential applications in management, particularly in areas such as monitoring, control, and optimization.In Asset management,used to monitor the location, condition, and usage of various assets, such as vehicles, equipment, and tools.In Facility management, sed to monitor and control various aspects of facility management, such as lighting, HVAC, and security.In Supply chain management, used to monitor and optimize various aspects of the supply chain, such as inventory, transportation, and delivery.In Environmental management, used to monitor and control various aspects of environmental management, such as pollution, waste, and natural resources. By deploying wireless sensors throughout the environment, the system can collect data on various parameters and implement various measures to reduce the negative impact on the environment.

CHAPTER 2

LITERATURE SURVEY

2.1 A Hybrid Fault-Tolerant Routing based on Gaussian Network for Wireless Sensor Network [1]

2.1.1 Overview

In hybrid fault-tolerant routing to solve fault-tolerant issue in wireless sensor networks (WSNs) based on hierarchical topology. The hierarchical topology is a combination of clustering and the labeling of sensor nodes as Gaussian integers. Accordingly, the network area is divided into small square grids, the cluster head of each grid is represented by a Gaussian integer. These cluster heads are connected together to create a Gaussian network. Through node symmetry, and the shortest distance in the Gaussian network, as well as the advantages of multi-path routing, this paper proposes a hybrid fault-tolerant clustering routing protocol based on Gaussian network for wireless sensor network (FCGW). The purpose of FCGW is to improve fault tolerance, increase data reliability and reduce energy consumption for wireless sensor networks. The experimental results of the proposed scheme show that FCGW protocol has high data reliability. In addition, the FCGW protocol consumes about 48% of the energy in the network, while other protocols consume 70% more energy

In wireless sensor network (WSN), the metrics such as data reliability, optimal energy consumption, memory limit, and data latency are major challenges in effective implementation of network. In particular, in WSN, due to limited resource of sensor nodes, as well as harsh communication environments such as rain, wind, snow and water, always leads to faulty connections. Therefore, improving fault tolerance in the network will improve the quality of service and availability of the WSN. Since then, addressing fault tolerance issues has been an important requirement in WSN design.

2.1.2 Advantage

- To improve fault tolerance And Increase data reliability and reduce energy consumption
- It is fault tolerant. The protocol can reroute data around failed nodes, ensuring that the data is always able to reach its destination
- It is reliable. The protocol uses multipath routing to increase the chances that at least one of the paths will be able to deliver the data, even if some of the paths fail
- It is energy efficient. The protocol uses energy-efficient routing techniques to conserve energy

2.1.3 Disadvanatage

- Concentrated on fault tolerence And Network Efficiency not mentioned
- It is more complex than other routing protocols
- It requires more overhead in terms of communication and computation
- It may not be suitable for all WSNs

2.2 Energy efficiency clustering based on Gaussian network for wireless sensor network [2]

2.2.1 Overview

The sensors' connection model design and energy optimisation of routing protocol in wireless sensor networks based on the Gaussian network connection model. Accordingly, by the node-symmetric and four different directions to four adjacent nodes of each node in the Gaussian network model, the study recommends a new wireless sensor network connection model, on which the network area will be divided into some virtual square grids. In the new wireless sensor network connection model, they describe each virtual square grid as a node in the Gaussian network, therefrom, they propose a routing method, which is a combination of the shortest path routing protocol in the Gaussian network and clustering protocol to improve the routing efficiency of the wireless sensor network. Some simulations were implemented in NS2, the results show that the proposed routing is very good.

A wireless sensor network (WSN) is a collection of hundreds of thousands of low-powered sensors.WSNs are used very popularly in certain application areas such as health monitoring, weather monitoring, environmental monitoring, traffic monitoring etc. The WSN applications have many important benefits, but some issues such as limited hardware, limited lifetime, limited bandwidth, routing efficiency etc., so there has always been interest in research to improve the quality of the service in the WSN. Among them, proposals for topology design and optimization of routing protocol in WSNs are always of concern. Especially, energy-efficient routing protocol prolonging network lifetime for the WSN is one of the key research studies.

2.2.2 Advantage

- Energy optimisation of routing protocol in WSN based on the Gaussian network connection model
- It is energy efficient. The protocol can reduce energy consumption compared to other routing protocols
- It is scalable. The protocol can be used in WSNs of any size
- It is robust. The protocol is able to adapt to changes in the network topology

2.2.3 Disadvantage

- Participation of Coverage hole is not mentioned
- It is more complex than other routing protocols
- It requires more overhead in terms of communication and computation
- It may not be suitable for all WSNs

2.3 On the Utilization of Shortest Paths in Complex Networks [3]

2.3.1 Overview

Considerable effort has been devoted to the study of network structures and connectivity patterns and their influence on network dynamics. A widely used assumption in network analysis models is that traffic follows the shortest paths connecting pairs of non-neighboring vertices. For example, graph centrality measures, community extraction algorithms, and core-periphery detection algorithms use this assumption. However, this is a very restricted perspective and can be misleading as a consequence of its focus on shortest path communications. In this work, we study the utilization of shortest paths in complex networks in different data dissemination scenarios. We also explore whether there are general properties that can make networks utilize shortest paths more effectively. By conducting simulations on a set of real-world and artificial networks, we show that the utilization of shortest paths in complex networks may not be as common as assumed. This implies that longer paths can be as important (in some cases) as the shortest paths. Our results show that at least two factors clearly influence shortest path utilization in a network: the structure of the network and the data dissemination algorithm. We also find that the type of a network is not a good indicator of its shortest path utilization.

2.3.2 Advantage

- The utilization of shortest paths in complex networks
- Shortest paths are the most efficient way to route traffic in many cases
- Shortest paths can be used to study the dynamics of complex networks
- Shortest paths can be used to improve the efficiency of routing algorithms

2.3.3 Disadvantage

- Improved node participation is not addressed
- Shortest paths may not be the most efficient way to route traffic in all cases
- Shortest paths may be congested, leading to performance degradation
- Shortest paths may not be available in all cases, leading to routing failures

2.4 DORA: A Destination Oriented Routing Algorithm for Energy-Balanced Wireless Sensor Networks [4]

2.4.1 Overview

A new multi-chain routing strategy named the destination oriented routing algorithm (DORA). The proposed algorithm uses the chain routing feature of Power-Efficient Gathering in Sensor Information Systems (PEGASIS) that considers both the practical transmission power and the direction from any source node to the sink, and then determines the preferred forwarding node. The proposed design generates a new multi-chain routing scheme to transmit packets for energy-balanced WSNs. Based on the multi-chain routing, the optimal transmission distance between any two nodes is derived by the mathematical analysis model. With the optimal distance, the design criterion is to select the furthest forwarding node within the communication range and the direction to the sink, thus forming the multi-chain structures by accurate distance, direction, and multiple paths. With the shorter chain routing, unnecessary energy loss can be reduced to prolong the global network lifespan. Simulation results demonstrate that the proposed DORA doubles the network lifespan, compared with that of conventional PEGASIS and improves 60% lifespan on the RPC protocol.

2.4.2 Advantage

- To produce optimal transmission distance between nodes
- It improves energy efficiency in WSNs
- It is scalable to large WSNs
- It is robust to node failures

2.4.3 Disadvantage

- Participation of Coverage hole is not mentioned
- It is more complex than other routing protocols
- It requires more overhead in terms of communication and computation
- It may not be suitable for all WSNs

2.5 Learning-based Adaptive Sensor Selection Framework for Multisensing WSN [5]

2.5.1 Overview

Wireless sensor nodes equipped with multiple sensors often have limited energy availability. To optimize the energy sustainability of such sensor hubs, in this paper a novel adaptive sensor selection framework is proposed. Multiple sensors monitoring different parameters in the same environment often possess cross-correlation, which makes the system predictive. To this end, a learning-based optimization strategy is developed using Upper Confidence Bound algorithm to select an optimum active sensor set in a measurement cycle based on the cross-correlations among the parameters, energy consumed by the sensors, and the energy available at the node. Further, a Gaussian process regressor-based prediction model is used to predict the parameter values of inactive sensors from the cross-correlated parameters of active sensors. To evaluate the performance of the proposed framework in real-life applications, an air pollution monitoring sensor node consisting of seven sensors is deployed in the campus that collects data at a default high sampling rate. Simulation results validate the efficiency and efficacy of the proposed framework. Compared to the current state-of-the-art the proposed algorithm is 54% more energy efficient, with complexity $\mathrm{O}(2\hat{\mathrm{P}})$ for P sensors in the node, while maintaining an acceptable range of sensing error.

2.5.2 Advantage

- To reinforce for miniature sensor hub with monitoring
- It improves energy efficiency in WSNs
- It is scalable to large WSNs
- It can be used in a variety of applications

2.5.3 Disadvantage

- Efficiency of Network is not mentioned
- It requires a training dataset
- It is more complex than traditional sensor selection methods
- It may not be suitable for all WSNs

2.6 Wireless-Sensor Network Topology Optimization in Complex Terrain: A Bayesian Approach [6]

2.6.1 Overview

Existing methods for wireless sensor network (WSN) topology optimization employ simplifying assumptions of a fixed communication radius between network nodes, which is ill-suited for IoT networks deployed in complex terrain. This article proposes a data-driven approach to WSN topology optimization, employing a Bayesian link classifier trained on LIDAR-derived terrain characteristics and an in-situ survey of link quality. The classifier is trained to predict where good network links (packet delivery ratio, PDR¿0.5) are likely to form in a region given complex terrain attributes. Then, given numerous candidate wireless node placements throughout the domain, the classifier is used to construct an undirected weighted graph of the potential connectivity across the domain. Edge weights in the connectivity graph are proportional to the probability of forming a good link between the nodes. A novel modified cycle-union (MCyU) algorithm for generating a 2-vertex-connected, Steiner minimal network is then applied to the undirected weighted graph of potential network element placements. This ensures a survivable network design, while maximizing the probability of good links within the final network. The total number and spatial distribution of network elements produced by the algorithm is compared to an existing wireless sensor network, deployed for environmental monitoring in remote regions. In addition, the MCyU algorithm has been evaluated in three graph test cases to compare with state-of-the-art solutions, where MCyU outperforms in terms of weight minimization and computation time.

2.6.2 Advantage

- To develop and assess method for optimisation in WSN
- It can account for the variability of terrain and other factors
- It can be used to develop more robust and accurate topologies
- It can improve the performance of WSNs in complex terrain

2.6.3 Disadvantage

- Neither lifespan efficiency of node nor coverage hole participation is addressed

- It requires a training dataset
- It is more complex than traditional methods
- It may not be suitable for all WSNs

2.7 Node position estimation for efficient coverage hole-detection in wireless sensor network [7]

2.7.1 Overview

In recent times, Wireless Sensor Networks (WSNs) have active involvement in diverse applications such as: environment monitoring, security & surveillance, health care, precision agriculture, industrial applications and many more. Generally, sensor nodes are deployed randomly in such applications. Therefore, estimation of node location is a legitimate problem in WSNs, due to the fact that uncovered region can result in coverage-holes in the network. One state-of-art solution of this problem is Global Positioning System (GPS) but GPS based solution to localize a node might not be worthy due to the cost of extra hardware and power requirements. Therefore, a low cost solution to this problem might be Computational Geometry based approach to localize a node. In this paper, we first find out distance between Anchor Node (AN) and Unknown Node (UN) based on RSSI Profiling. Subsequently, the node location is estimated using Trilateration. Finally, a Delaunay Triangle is constructed on the basis of node location information. Then the property of empty circle is used to recognize whether coverage hole is present or not in the given ROI. Correctness of the algorithm is checked based on the simulations and theoretical proofs.

WSNs, well-known for their specialization on surveillance, are specifically used to monitor physical events like light, sound, temperature, pressure etc. with the help of small sensor nodes those have very less power requirements. A lot of applications like home automation, fire detection, air quality monitoring, habitat monitoring, endangered species recovery, battle field monitoring etc. depend on WSNs. In all such applications one needs to know the exact position of the sensor node and that is why localization is an important characteristic in research related to WSN. Although GPS can estimate the node location accurately, but it is not feasible due to the following drawbacks: GPS might exhibit errors while finding locations if satellite link is down; cost of GPS receiver might again increase the expenditure of the whole WSN; GPS is not energy efficient whereas, energy scarcity is very common problem faced in WSNs; sensor nodes are intentionally made very small in size but if accompanied with GPS receiver, due to increased size they might not be deployed very close to the event. So we have planned to identify the node location using GPS-free schemes.

2.7.2 Advantage

- Low cost approach to localisation
- Improved efficiency of coverage hole detection
- Reduced energy consumption
- Improved data quality

2.7.3 Disadvantage

- Node participation is not mentioned
- The cost of node position estimation can be high, depending on the method that is used
- The complexity of node position estimation can be high, depending on the method that is used
- The accuracy of node position estimation can be low, depending on the method that is used

2.8 A Secure and Energy Efficient Barrier Coverage Scheduling for WSN-Based IoT Applications [8]

2.8.1 Overview

Barrier coverage scheduling is an energy conservation scheme in which a subset of sensor nodes with overlapped sensing area (also called barrier) is activated to meet the key Quality of Service (QoS) requirements such as energy-efficiency, coverage, and connectivity. However, sudden and unexpected node failures in a barrier due to security attacks such as Denial of Service (DoS) poses a challenge for maintaining the desired QoS levels. In this paper, we propose a secure barrier coverage scheduling scheme called SEC2 , which prevents QoS degradation in the event of security breaches. This scheme uses a fully weighted attributed dynamic graph model in which a novel attribute-based weight balancing greedy strategy is used to construct barriers. A weighted averaging-based K-means Spectral, and Hierarchical (WKSH) cluster ensemble scheme is proposed to secure a barrier from malicious attacks. WKSH is a graph-based anomaly detection scheme based on weighted Euclidean distance computation and weighted average consensus. The experimental result shows that SEC2 guarantees the required QoS at all times. Moreover, the proposed WKSH shows better accuracy in the classification and detection of attacks in the barrier. Index Terms—Anomaly detection, Barrier coverage, Dynamic Graph, Energy Efficiency, Green IoT, Security, WSN.

WSNs provide the necessary network infrastructure for various service-critical green IoT applications such as military, mining, and habitat monitoring. The key QoS requirements of such applications include energy efficiency (or extended network lifetime), coverage, and connectivity. Meeting these requirements is a challenge in resource-constrained WSNs. Energy efficiency is an essential requirement to extend the network lifetime of power-constrained WSNs. In addition to this requirement, coverage and connectivity must also be guaranteed at all times for improved event detection accuracy and reduced detection delay. Coverage and connectivity are the two most essential QoS requirements for mission-critical surveillance applications that are least tolerant of security lapse.

2.8.2 Advantage

- Anomaly detection scheme is proposed also with coverage issue
- Barrier coverage scheduling can significantly improve the energy efficiency of WSNs by reducing the number of active sensor nodes
- Barrier coverage scheduling can also extend the network lifetime of WSNs by reducing the energy consumption of the sensor nodes
- Barrier coverage scheduling can make WSNs more robust to node failures. This is because if a sensor node fails, it can be replaced by another sensor node without affecting the overall coverage of the network
- Barrier coverage scheduling is scalable to large WSNs. This is because the algorithm can be easily extended to cover larger areas

2.8.3 Disadvantage

- Mitigation to coverage issue by node participation is not addressed

- Barrier coverage scheduling is a complex algorithm. This is because it requires the calculation of the optimal barrier coverage solution
- Barrier coverage scheduling can introduce some overhead in the network. This is because the algorithm requires the exchange of messages between the sensor nodes and the sink node
- Barrier coverage scheduling can be vulnerable to attacks. This is because the algorithm requires the exchange of sensitive information, such as the location of the sensor nodes

2.9 Improving lifetime of wireless sensor networks based on nodes distribution using Gaussian mixture model in multi-mobile sink approach [9]

2.9.1 Overview

Saving energy in Wireless Sensor Networks (WSNs), is critical in different applications, such as environment monitoring, keeping human awareness and etc. Many studies have investigated energy consumption and improved the WSN lifetime longevity by reducing the energy consumption. Still, proposed approaches overlook the nodes' distribution role in energy model and routing protocol, which is a key factor in a WSN. In this work, we propose a novel approach; namely GDECA; which assumes nodes' distributions are mixtures of Gaussian distribution, as an assumption applied in real world. So GDECA rely on a distribution estimation borrowed from Machine Learning (ML) to fit the Gaussian Mixture Model (GMM) to the nodes and calculate the parameters for these distributions. Next, the estimated parameters are employed in Cluster Head CH selection policy. Besides, sinks routing is determined based on nodes distribution. Results showed the improvement close to 40–50% in energy consumption. As another outcome, GDECA keeps all the nodes active until end of the simulation. Observations also demonstrate that sinks path calculation using this approach is optimum, and randomly changing number of sinks increases energy consumption.

2.9.2 Advantage

- Optimized energy-efficient routing
- The multi-mobile sink approach with GMM can significantly improve the lifetime of WSNs by reducing the energy consumption of the sensor nodes

- Make WSNs more robust to node failures. This is because if a sensor node fails, the data can be routed to the mobile sink that is closest to the failed node. This ensures that the data is still collected and transmitted to the sink node
- The multi-mobile sink approach with GMM is scalable to large WSNs. This is because the algorithm can be easily extended to cover larger areas

2.9.3 Disadvantage

- Coverage hole issue and deployment of movable node is not mentioned
- The multi-mobile sink approach with GMM is a complex algorithm. This is because it requires the calculation of the optimal placement of the mobile sinks
- The multi-mobile sink approach with GMM can introduce some overhead in the network. This is because the algorithm requires the exchange of messages between the sensor nodes and the mobile sinks
- The multi-mobile sink approach with GMM can be vulnerable to attacks. This is because the algorithm requires the exchange of sensitive information, such as the location of the sensor nodes

2.10 A Survey on Holes Problem in Wireless Underground Sensor Networks [10]

2.10.1 Overview

The problem of the existing holes in the Wireless Underground Sensor Networks (WUSNs). Anomalies in wireless sensor networks can weaken the network by impeding its functionalities. These existing anomalies commonly known as holes may cause irreversible devastating problems and may prove fatal to humans. There can be different holes in the network, such as coverage hole, routing hole, jamming hole, sink/ worm hole, etc. This paper intends to discuss various holes problems in WUSNs, reasons behind their occurrence, and different techniques to mitigate the issues. An efficient hole detection technique shall increase network lifetime, use lesser resources and give accurate results with zero false positives. To put a light on the present scenario, methods/ techniques used to cope with different types of holes have been discussed in detail presenting their analysis, constraints, and advantages. Besides, it would also touch upon the comparative analysis in terms of soundness and limitations of these techniques. The study is concluded with future research directions.

Wireless underground sensor networks (WUSNs) are an extension of wireless sensor networks in which wireless sensors are placed beneath the earth's surface. WUSN is an emerging and promising research area. It is a field that will make possible an extensive number of novel applications that were not doable before and shall aid in existing applications in both civilian and military lobbies. The wireless medium acts as a mainstay between sensors for collaborative computing. The same is evident in agriculture monitoring, intruder detection, border patrol, sports ground upkeep, infrastructure monitoring, and environmental monitoring, etc. With shifting research trends toward IoT, WSNs and CRNs (Cognitive Radio Networks) need more attentions. Wireless Sensor Networks (WSNs) comprise small sensor nodes varying from tens to thousands. These tiny nodes are contained with battery power, sensing, communication, and data processing abilities. To put it in another way, WSNs consist of a number of embedded systems that can interact with the environment, process the collected data and pass this information to their neighbor destination nodes.

2.10.2 Advantage

- To Mitigate coverage hole issue
- Provides a comprehensive overview of the holes problem in WUSNs
- Discusses the causes of holes in WUSNs and the different types of holes that can occur
- Reviews the different techniques that have been proposed to solve the holes problem in WUSN
- Provides a comparative analysis of the different techniques that have been proposed to solve the holes problem in WUSNs

2.10.3 Disadvantage

- Node participation in coverage hole is not addressed
- Does not provide any experimental results to evaluate the effectiveness of the different techniques that have been proposed to solve the holes problem in WUSNs
- Does not discuss the security implications of the holes problem in WUSNs

2.11 Enhanced LEACH protocol for increasing a lifetime of WSNs [11]

2.11.1 Overview

WSNs that stand for wireless sensor networks and include many low-cost and low power-sensing tools, local processing, and the capacity of wireless communication face some problems in two aspects: the lifetime of the network and its energy. Therefore, the aim of this paper is to overcome these limitations through enhancing the LEACH (low energy adaptive clustering hierarchy) protocol, the protocol of cluster routing, in which, LEACH is extended by identifying a cluster head according to the lowest degree of distance from the base station in order to decrease power consumption in cluster head nodes and in the whole network. Hence, the results clarify the ability of LEACH in enhancing the network lifetime as well as in reducing and minimizing the consumption of power.

2.11.2 Advantages

- Identifies a cluster head according to the lowest degree of distance from the base station in order to decrease power consumption in nodes
- Enhanced LEACH protocol can significantly improve the lifetime of WSNs by reducing the energy consumption of the sensor nodes
- Enhanced LEACH protocol divides the network into a number of clusters, with each cluster having a cluster head. The cluster heads are responsible for collecting data from the sensor nodes in their cluster and transmitting it to the sink node. This reduces the number of sensor nodes that need to transmit data to the sink node, which reduces their energy consumption

2.11.3 Disadvantage

- Mitigation to coverage issue by node participation is not addressed
- The Enhanced LEACH protocol is a more complex protocol than the original LEACH protocol. This is because the Enhanced LEACH protocol uses a number of additional techniques to improve the lifetime of the network

- Enhanced LEACH protocol introduces some overhead in the network. This is because the Enhanced LEACH protocol requires the exchange of additional messages between the sensor nodes and the sink node
- The Enhanced LEACH protocol is vulnerable to attacks. This is because the Enhanced LEACH protocol uses a number of techniques that can be exploited by attackers, such as the cluster formation and power-aware routing techniques

2.12 Enhancing the Lifetime of Wireless Sensor Networks Using Fuzzy Logic LEACH Technique-Based Particle Swarm Optimization [12]

2.12.1 Overview

Wireless sensor networks (WSNs) have attracted significant attention because of their widespread use in health care, habitat tracking, disaster prevention, agriculture, monitoring areas, fire tracking, and other real-life applications. The lifetime of WSNs must be prolonged to increase their use for various applications. One of the most effective methods for improving the network's lifetime is clustering with the optimal cluster head (CH). This study proposes a fuzzy Logic (FL) low-energy adaptive clustering hierarchy (LEACH) technique-based particle swarm optimization (PSO). It employs hybrid PSO and a K-means clustering algorithm for cluster formation. It selects the primary CH (PCH) and secondary CH (SCH) using FL. Extensive simulations were conducted using a simulation program to validate the proposed protocol's performance. Furthermore, the proposed protocol was compared with traditional algorithms, such as fuzzy c-means (FCM) clustering and FLS-based CH selection to enhance the sustainability of WSNs for environmental monitoring applications, LEACH-Fuzzy clustering protocol, and LEACH based on energy consumption equilibrium. The results confirmed that the proposed protocol efficiently balances energy consumption to improve wireless sensor network performance and to maximize throughput. The simulated results indicated that network lifetime was improved by more than 46% and packet transmission by 17.6%

In this study, a FL LEACH Technique-based PSO is proposed for energy efficiency and for maximizing the number of bits transferred to BS.The setup phase consists of cluster formation using a hybrid PSO and K-means clustering algorithm. The sensor network is in a fixed state. If the nodes have been deployed, they cannot be relocated. Because we are testing fixed nodes, there is no need to repeat cluster formation per round.The steady phase includes three main steps: PCH selection, SCH selection, and intracluster data communication. This phase is repeated in each round,because the energy of CMs, PCHs, and SCHs changes during the rounds; thus, there is a need to repeat it.First, cluster formation is performed depending on the hybrid PSO and K-means clustering algorithm. Input data, which are the locations of the (N) sensors, are clustered into a specified number of clusters (Nc).Nc should be obtained using Gap statistic clustering evaluation. The gap statistic compares the total within intracluster variation for various values of Nc to their predicted values under the data's null reference distribution. The value that maximizes the gap static will be used to determine the best clusters.. Distance metric (Euclidean distance) is used for to compute the criterion values. This indicates that the clustering structure differs significantly from a random uniform distribution of points.Second, PCH is selected based on the node's residual energy, distance from the node to the cluster's center, and distance from the node to BS. Third, SCH is selected based on the node's energy level and distance from the node to PCH within its cluster. Furthermore, the BS runs the FL to select the entire PCH and SCH nodes in the network. Finally, intracluster data communications, all CMs sense data and transfer them to SCH within the cluster. Each SCH acquires and aggregates the data from all CMs for transmission to PCH. Each PCH receives data from all SCHs and forwards them to the BS.

2.12.2 Advantages

- It proposes a Fuzzy Logic(FL) low-energy adaptive clustering hierarchy(LEACH) technique-based particle swarm optimisation
- It is more energy efficient than LEACH
- It can guarantee that the cluster heads will have enough energy to complete their tasks
- It is more efficient in terms of energy consumption.

2.12.3 Disadvantage

- It is more complex than LEACH
- It requires more computation power
- Do not mentioning about the coverage hole issue and its mitigation measures

Table 2.1: Literature Comparison Table (1)

Literature Review			
Literature	Methodology	Merit	Demerit
Coverage Hole and Boundary Node Detection	Proposed algorithm for coverage hole and boundary node refining	To detect Coverage hole and Boundary Nodes	Unable to obtain the shortest path between the interconnecting nodes in the wireless sensor network
Hybrid Fault tolerant Routing	Labelling sensor nodes as Gaussian Integer and by creating Gaussian Network	To solve fault tolerant issue in WSN	Neither shortest path utilization nor coverage hole identification is mentioned
Coverage hole detection by optimized Gaussian Mixture Model	Detection of energy depleted sensor nodes	To mitigate coverage hole issue	Not recommended for efficient routing of sensor nodes
Utilization of Shortest path in Complex Network	Graph centrality measures, community extraction and core-periphery algorithms	To increase network efficiency	Energy depletion due to coverage holes is not considered
Holes problem in Wireless Underground Sensor Network	Analysis, constraints, advantages on holes related problem	To mitigate coverage hole issue	Routing of nodes to sink node is not addressed
Holes problem in Wireless Sensor Network	Discussed various hole problems, characteristics and their study	To detect coverage hole	Routing of nodes to sink node is not addressed
Novel Coverage Hole detection and Recovery	Active contour model-based coverage hole detection	To detect coverage hole and its recovery	Optimized path is not obtained by means of active contour model-based coverage hole detection

Table 2.2: Literature Comparison Table (2)

Literature Review			
Literature	Methodology	Merit	Demerit
Energy efficiency clustering based on Gaussian Network	Network area divided into virtual square grid as nodes in Gaussian Network	Sensor connection design model and energy routing protocol	No consideration on coverage hole identification
Destination Oriented Routing Algorithm	Chain routing feature of PEGASIS	To produce optimal transmission distance between nodes	Poor results when compared to Gaussian Network model
Adaptive Sensor Selection Framework	Learning-based Optimization Strategy	To reinforce miniature sensor hub with monitoring	Network efficiency is attained by sensor selection instead of energy-efficient routing protocols
Topology Optimization: Bayesian Approach	Data-driven approach to topology optimization	To develop and assess methods for optimization in WSN	Expensive to evaluate the output of Bayesian approach
Energy-Efficient Path Planning Strategy	Optimized energy-efficient Path Planning Strategy	To extend network lifespan and path optimization	Need for coverage hole identification and mitigation measures are not proposed
Safety Architecture Proposal for Low-Latency Sensor	Enhance IOLW to make safety-critical systems	Security-for-safety mechanism is introduced	Safety is introduced but connecting nodes to the base station is not discussed

Table 2.3: Literature Comparison Table(3)

Literature Review			
Literature	Methodology	Merit	Demerit
Gaussian Mixture Model in multimobile sink approach	Optimized energy-efficient routing using GDECA method	Considering node distribution is significant in energy saving	GDECA keeps all the nodes active until simulation, causing the depletion of energy
A Secure and Energy Efficient Barrier Coverage Scheduling for WSN	Secure barrier coverage scheduling scheme called SEC2	Anomaly detection scheme called WKSH is proposed	Focused on security of the network using barrier coverage scheduling
Node Position Estimation for Efficient Coverage Hole Detection in WSN	Computational Geometry Based approach to localize a node	Low-cost approach to localization	Efficient routing from nodes to the base station is not discussed
Enhanced LEACH protocol for increasing the lifetime of WSN	Identifies a cluster head according to the lowest degree of distance from the base station to decrease power consumption in nodes	Increased lifetime in WSN	Mitigation of coverage hole by node participation is not mentioned
Enhancing the Lifetime of WSN Using Fuzzy Logic LEACH Technique-Based Particle Swarm Optimization	Proposes Fuzzy Logic LEACH (FL-LEACH) technique-based Particle Swarm Optimization (PSO)	To improve network lifetime. It is more energy efficient than LEACH. It can guarantee that the cluster heads will have enough energy to complete their tasks. It is more efficient in terms of energy consumption.	Node participation in coverage hole is not mentioned. It is more complex than LEACH. It requires more computation power.

CHAPTER 3

PROPOSED SYSTEM

3.1 Proposed System Architecture

Wireless sensor networks are infrastructure-less and fully distributed systems of self-configurable and self-organized nodes that wish to share the IoT devices' data over the air.Architecture used in WSN is :

Layered Network Architecture - Makes use of a few hundred sensor nodes and a single base station.

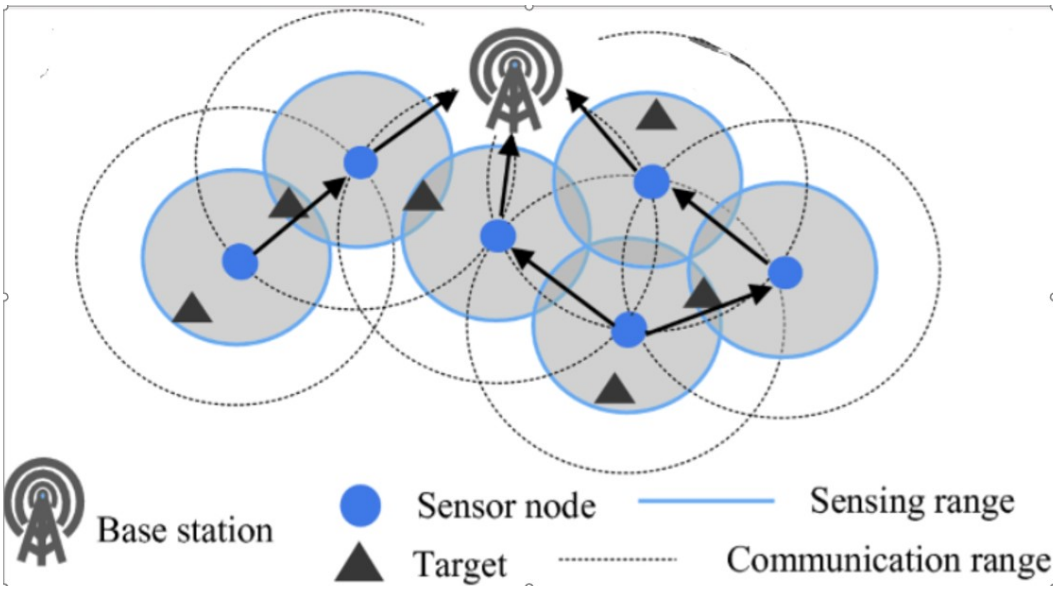

FIGURE 3.1: Layered Network Architecture.

3.2 Workflow

- Random Distribution of WSN
- Implement Delaunay triangulation for Identifying Coverage holes
- Optimally placing movable nodes in coverage holes and locally retriangulate
- Implementation of improved energy routing Clustering protocol
- Implementation of Optimised leach protocol
- Combining both improved clustering protocol and optimised leach protocol
- Obtaining the shortest path from the source node to the base station
- Resulting to obtain improved energy efficient Wireless Sensor Network

3.3 Basic Concepts

3.3.1 Random distribution

Random distribution of nodes refers to a spatial arrangement of nodes or points in a given area or domain where their positions are determined by a random process. It is commonly used in various fields, including computer science, mathematics, physics, and network analysis, to model and analyze systems or phenomena where randomness plays a role. Here are some basic concepts related to random distribution of nodes:

- **Randomness:** Random distribution implies that the positions of nodes are determined by a random or probabilistic process. The exact location of each node is not predetermined or influenced by any specific pattern or rule.
- **Uniformity:** In a uniform random distribution, each point or node has an equal probability of appearing anywhere within the defined area. This means that the distribution is spatially homogeneous, and nodes are equally likely to occur in any given region of the domain.
- **Density:** The density of a random distribution refers to the concentration or number of nodes within a specific area or volume. Higher density indicates a larger number of nodes in a given region, while lower density means fewer nodes in the same area.

- **Independence:** Random distributions assume that the positions of individual nodes are independent of each other. This means that the location of one node does not influence the location of another node in the distribution.
- **Random Number Generation:** To create a random distribution of nodes, a random number generator is often employed to determine the positions of the nodes. The generator produces a sequence of numbers that are statistically random and can be used to assign coordinates to the nodes within the defined domain.
- **Spatial Constraints:** Depending on the specific application, random distributions of nodes may be subject to certain spatial constraints. For example, nodes may need to be confined within a specific boundary, avoid overlap with other nodes, or follow certain rules related to distances or connectivity.
- **Applications:** Random distributions of nodes have numerous applications, such as generating random graphs, simulating particle systems, modeling wireless sensor networks, studying the behavior of complex systems, and conducting Monte Carlo simulations.

3.3.2 Delaunay triangulation

Delaunay triangulation is a geometric algorithm used to partition a set of points into a triangulated network. It has applications in various fields, including computer graphics, computational geometry, and wireless communication. When it comes to identifying coverage holes in a network, Delaunay triangulation can be a useful tool. Here are some basic concepts related to Delaunay triangulation and its application in identifying coverage holes:

- **Triangulation:** Triangulation is the process of dividing a region into triangles, where each point in the region becomes a vertex of one or more triangles. Delaunay triangulation is a specific type of triangulation that ensures certain optimality properties.
- **Delaunay Triangulation:** In a Delaunay triangulation, a set of points is connected by edges to form triangles such that no point lies inside the circumcircle of any triangle. This property guarantees that the resulting triangulation is as close to an equilateral triangle as possible, avoiding skinny or elongated triangles.
- **Voronoi Diagram:** The Voronoi diagram is closely related to Delaunay triangulation. It divides the plane into polygons, with each polygon consisting of all points closer to a particular input point than to any other input point. The edges of the Voronoi polygons coincide with the Delaunay triangulation edges.

- **Identifying Coverage Holes:** In the context of wireless communication or sensor networks, coverage holes refer to areas where there is inadequate signal coverage or sensor detection. By performing a Delaunay triangulation of the network's sensor nodes or signal sources, one can identify the triangles that do not have sufficient coverage. These triangles correspond to the coverage holes in the network.

- **Hole Detection:** To detect coverage holes using Delaunay triangulation, one can examine the triangles in the triangulated network and check for triangles that have a large circumradius or a small area. A large circumradius indicates sparsely placed nodes, and a small area suggests insufficient coverage. Such triangles can be flagged as potential coverage holes.

- **Hole Remediation:** Once coverage holes are identified, appropriate measures can be taken to address them. This might involve repositioning existing nodes, adding new nodes, adjusting transmission power levels, or employing other optimization techniques to improve coverage in the identified areas.

- **Optimization Criteria:** Depending on the specific application, the criteria for identifying coverage holes may vary. The selection of optimal triangles can be based on factors such as signal strength, signal-to-noise ratio, connectivity, or other performance metrics specific to the network or system being analyzed.

Delaunay triangulation provides a robust and efficient way to analyze the spatial distribution of points and identify coverage holes in various network scenarios. By leveraging its geometric properties, one can gain insights into coverage issues and devise strategies to enhance network performance.

3.3.3 Improved Node Participation

When it comes to optimally placing movable nodes in coverage holes and locally retriangulating, several concepts and techniques can be applied. Here are the basic concepts related to these processes:

- **Coverage Holes:** Coverage holes refer to areas within a network where the signal strength or sensor coverage is insufficient. These areas may have been identified through techniques such as Delaunay triangulation or other coverage analysis methods.

- **Movable Nodes:** Movable nodes are additional nodes that can be placed strategically within the network to improve coverage and fill the identified holes. These nodes are not fixed and can be repositioned as needed to achieve optimal coverage.

- **Node Placement Strategies:** Various strategies can be used to place movable nodes in coverage holes effectively. Some common approaches include:
  - **Centroid-Based Placement:** The centroid of each coverage hole can be calculated, and a movable node can be placed at the centroid to provide coverage in that area.
  - **Voronoi-Based Placement:** The Voronoi diagram, derived from Delaunay triangulation, can be utilized to determine the regions that are closest to each coverage hole. Movable nodes can be placed at the centers of these Voronoi regions to cover the respective holes.
  - **Optimization Algorithms:** Optimization techniques such as genetic algorithms, particle swarm optimization, or simulated annealing can be employed to find optimal positions for movable nodes based on certain criteria, such as minimizing the total number of nodes required or maximizing coverage with a limited number of nodes.
- **Local Retriangulation:** Once movable nodes are placed in coverage holes, the local retriangulation process adjusts the existing triangulation in the immediate vicinity of the placed nodes. This ensures that the network connectivity and triangulation quality are maintained or improved.
- **Incremental Delaunay Triangulation:**Incremental Delaunay Triangulation: Incremental Delaunay triangulation is a technique that adds or removes nodes from an existing Delaunay triangulation to update the triangulation efficiently. When placing movable nodes in coverage holes, the local retriangulation can be achieved by adding the movable nodes to the existing triangulation and adjusting the neighboring triangles accordingly.
- **Delaunay Refinement:** Delaunay refinement methods enhance the quality of triangulations by iteratively refining the triangulation based on certain criteria. This process can be employed to improve the quality of the triangulation around the placed movable nodes and their neighboring triangles, ensuring better connectivity and optimized coverage.
- **Performance Evaluation:** The effectiveness of the placement of movable nodes and local retriangulation can be assessed through performance evaluation metrics. These metrics may include measures such as signal strength, coverage area, connectivity, or other application-specific criteria. Evaluating the performance helps validate the effectiveness of the node placement strategy and fine-tune the system if necessary.

By strategically placing movable nodes in coverage holes and locally retriangulating the network, optimal coverage can be achieved, improving the overall performance and connectivity of the network. The specific techniques and algorithms employed will depend on the requirements, constraints, and characteristics of the network being analyzed.

3.3.4 Implementing a clustering protocol

Implementing a clustering protocol involves applying a set of algorithms and procedures to organize a network into clusters or groups of nodes. Clustering protocols are commonly used in various types of networks, including wireless sensor networks, ad hoc networks, and distributed systems. Here are some basic concepts related to the implementation of clustering protocols:

- **Network Organization:** Clustering protocols aim to divide a network into clusters, where each cluster consists of a subset of nodes. The goal is to create a structure that promotes efficient communication, resource management, or task distribution within the network.

- **Cluster Formation:** The process of cluster formation involves selecting cluster heads or leaders that coordinate the activities within their respective clusters. Typically, cluster heads are chosen based on specific criteria, such as node energy level, connectivity, or other metrics that reflect their suitability for the role.

- **Cluster Formation Criteria:** The criteria for selecting cluster heads can vary depending on the specific protocol and network requirements. Common factors considered during cluster head selection include node energy, distance to other nodes, connectivity, or a combination of these factors. The selection criteria are designed to optimize network performance, energy efficiency, or other relevant objectives.

- **Cluster Maintenance:** Once clusters are formed, cluster maintenance involves managing the cluster structure and ensuring its stability over time. This includes tasks such as monitoring the energy levels of cluster heads, re-electing cluster heads if necessary, detecting and handling node failures or departures, and managing cluster membership.

- **Cluster Communication:** Communication within and between clusters is a fundamental aspect of clustering protocols. Nodes within the same cluster typically communicate directly with their respective cluster head, which then handles inter-cluster communication. Depending on the protocol, cluster heads may exchange information, aggregate data, or coordinate tasks to achieve network objectives efficiently.

- **Data Aggregation:** Data aggregation is a technique often used in clustering protocols to reduce the amount of data transmission and save energy. Cluster heads collect data from their member nodes and aggregate it before transmitting it to higher-level nodes or the sink node. Data aggregation helps in reducing redundant data transmission, network congestion, and energy consumption.

- **Cluster Dynamics:** Clustering protocols should be designed to adapt to changes in the network, such as node failures, node mobility, or changing network conditions. Dynamic clustering protocols incorporate mechanisms to handle these changes, such as re-electing cluster heads, redistributing cluster members, or updating cluster parameters based on network dynamics.

- **Protocol Evaluation:** The implementation of a clustering protocol should be evaluated using appropriate metrics to assess its performance and effectiveness. Common metrics include network lifetime, energy efficiency, scalability, data accuracy, network coverage, and overhead.

It's important to note that specific clustering protocols may have different implementation details and algorithms. The choice of clustering protocol depends on the characteristics of the network, application requirements, and desired network behavior.

3.3.5 LEACH

LEACH (Low Energy Adaptive Clustering Hierarchy) is a popular clustering protocol designed for wireless sensor networks to achieve energy efficiency and prolong network lifetime. Here are some basic concepts related to the LEACH protocol:

- **Clustering:** LEACH organizes the network into clusters, with each cluster consisting of a cluster head (CH) and a set of sensor nodes. Clustering helps in reducing energy consumption by localizing communication and aggregating data within clusters.

- **Randomized Cluster Head Selection:** LEACH utilizes a randomized algorithm to select cluster heads. In each round, sensor nodes decide whether to become a cluster head based on a predetermined probability threshold (usually a random number between 0 and 1). This probabilistic selection ensures the distribution of cluster head roles across the network and avoids overloading specific nodes.

- **Cluster Formation:** After cluster head selection, non-cluster head nodes join the nearest cluster based on the signal strength from the cluster head. Each node becomes a member of only one cluster.
- **Data Aggregation:** LEACH employs data aggregation techniques to reduce the amount of data transmission and conserve energy. Cluster heads collect data from their member nodes, aggregate it, and transmit aggregated data to the base station or the sink node. Data aggregation reduces redundant transmissions and improves network efficiency.
- **Cluster Rotation:** To distribute energy consumption evenly across the network and avoid early depletion of cluster heads, LEACH introduces a mechanism called cluster rotation. After each round, the cluster heads change dynamically to evenly distribute the energy load. New cluster heads are chosen based on the randomized selection process.
- **TDMA-Based Scheduling:** Time Division Multiple Access (TDMA) is used for scheduling communication within clusters. Each cluster head assigns time slots to its member nodes, allowing them to transmit data without interference.
- **Energy Efficiency:** LEACH aims to achieve energy efficiency by minimizing the energy consumed for communication and data aggregation. By using clustering, data aggregation, and cluster rotation, LEACH balances energy consumption among nodes and extends the network lifetime.
- **Protocol Adaptation:** LEACH is designed to adapt to changes in network conditions. It incorporates mechanisms to handle node failures, node mobility, or changes in network topology. Cluster heads may be reselected periodically to account for changes in node energy levels and connectivity.
- **Protocol Evaluation:** The performance of the LEACH protocol is typically evaluated using metrics such as network lifetime, energy efficiency, data accuracy, and scalability. Simulation or analytical models are often employed to assess the protocol's effectiveness in different scenarios.

LEACH has been a pioneering protocol in the field of wireless sensor networks and has inspired numerous variations and enhancements. It provides a flexible and energy-efficient approach for clustering and data aggregation, enabling efficient data gathering and prolonging the network's operational lifetime.

3.3.6 Shortest Path Routing

Shortest path routing is a fundamental concept in network routing, aiming to find the most efficient path between a source node and a destination node in a network. It is widely used in various types of networks, including computer networks, transportation networks, and telecommunications networks. Here are some basic concepts related to shortest path routing:

- **Graph Representation:** A network is typically represented as a graph, where nodes represent the network entities (e.g., routers, intersections, or nodes) and edges represent the connections between them (e.g., communication links, roads, or edges). The graph can be either directed (one-way connections) or undirected (bidirectional connections).

- **Source and Destination Nodes:** Shortest path routing involves determining the optimal path from a source node to a destination node within the network. The source node represents the node where the routing process initiates, and the destination node is the intended final destination.

- **Cost or Weight Metric:** Each edge in the network graph is associated with a cost or weight metric that represents the measure of distance, delay, or other factors. The cost can reflect various characteristics, such as link bandwidth, transmission delay, hop count, congestion level, or any other relevant metric. The goal of shortest path routing is to find the path with the minimum total cost or weight.

- **Dijkstra's Algorithm:** Dijkstra's algorithm is a widely used algorithm for finding the shortest path in a graph. It starts from the source node and explores neighboring nodes, calculating the cumulative cost of reaching each node. The algorithm iteratively selects the node with the minimum cost and updates the cost of its adjacent nodes until the destination node is reached.

- **Bellman-Ford Algorithm:** The Bellman-Ford algorithm is another well-known algorithm for finding the shortest path. It can handle networks with negative edge weights but is generally slower than Dijkstra's algorithm. The Bellman-Ford algorithm iteratively relaxes the edges in the network, gradually refining the estimates of the shortest path until convergence is reached.

- **Shortest Path Tree:** When applying shortest path routing algorithms, a shortest path tree is constructed. The tree represents the minimum-cost paths from the source node to all other nodes in the network. It provides a comprehensive view of the network's connectivity and allows for efficient routing decisions.

- **Routing Table:** Shortest path routing protocols maintain routing tables that store information about the shortest paths to different destinations in the network. The routing table contains the next-hop information, which specifies the next node or interface towards the destination for each possible destination in the network.

- **Dynamic Routing:** In dynamic routing, the shortest path is continuously updated based on changes in the network topology or link conditions. Routing protocols such as OSPF (Open Shortest Path First) and RIP (Routing Information Protocol) use dynamic routing algorithms to adapt to network changes and ensure efficient routing in real-time.

- **Evaluation Metrics:** Shortest path routing can be evaluated using various metrics, depending on the network's objectives and requirements. Common evaluation metrics include path length (total cost), path latency, network congestion, packet loss, or other performance indicators.

Shortest path routing is a key concept in network communication and plays a vital role in determining efficient data paths. The selection of the shortest path algorithm and the choice of cost metric depend on the specific network environment, application requirements, and performance objectives.

3.4 LEACH Protocol

A wireless sensor networks consist of tiny sensor nodes to monitor physical or environmental conditions such as temperature, pressure, sound, humidity etc. The network must possess self configuration capabilities as the positions of the individual sensor nodes are not predetermined. Routing strategies and security issues are a great research challenge now days in WSN . A number of routing protocols have been proposed for WSN but the most well known are hierarchical protocols like LEACH and PEGASIS. Hierarchical protocols are defined to reduce energy consumption by aggregating data and to reduce the transmissions to the Base Station. LEACH is considered as the most popular routing protocol that use cluster based routing in order to minimize energy consumption.

- Wireless sensor networks (WSN) are a gathering of several low-power and low-cost network sensors which are used to sense an environment, collect data, process the collected data, and transmit the handled data to a base station (BS).

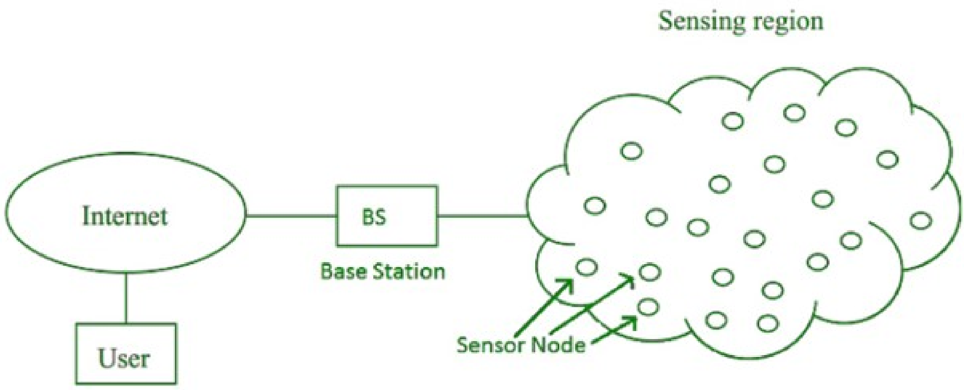

FIGURE 3.2: A Simple Wireless Sensor Network Representation

- WSNs depends on several small disposable independent devices called sensor nodes to form a network. The specific nodes in WSN can sense an environment, process the sensed data, or send it to a central unit for processing through a wireless link. The daily demand for WSN keeps increasing, ranging from military use to national, ground, and space usage.

- The major problems in the WSN are the large number of nodes used, their low power rating, and their restriction to short distance communication.

- Researchers have widely grouped sensor nodes into clusters in direction to attain the aim of network scalability; each group has a cluster-head (CH).

- There are several benefits of clustering and the greatest of it all is the implementation of an enhanced organization strategy which additionally extends the lifetime of the sensor batteries and further improves the network operation life. The other nodes in the cluster can opt to operate in low power mode saving energy etc.

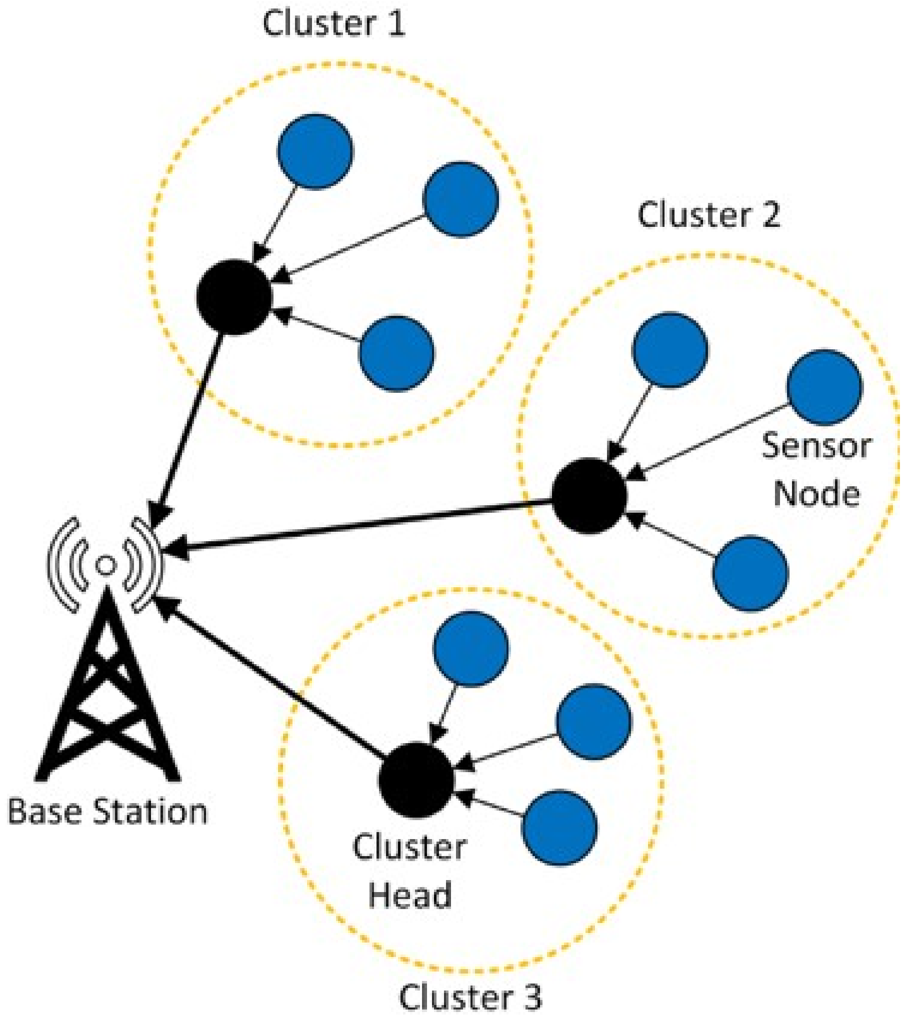

FIGURE 3.3: Clustering Of Nodes

3.4.1 Data Collection Methods vs LEACH

- **Direct Transmission**
  - Each sensor node transmits directly to the sink, regardless of distance
  - Most efficient when there is a small coverage area. Easy for the nodes to lose power
- **Multi Hop**
  - Data is transferred to nearby nodes and keeps traversing until it reaches the BS.

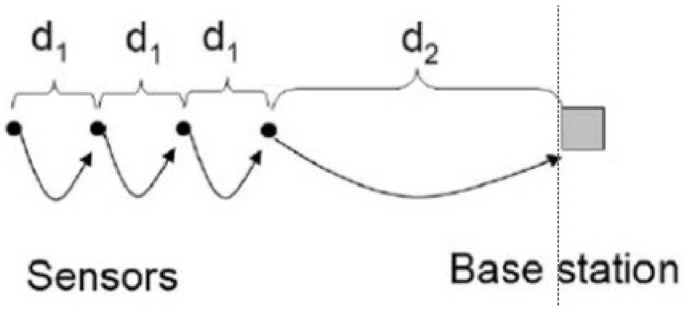

FIGURE 3.4: Multi Hop

- **Static Clustering**
  - Fixed CH and clusters. CH collects data from the cluster nodes.Once CH dies, that particular cluster is useless

3.4.2 LEACH - Low Energy Adaptive Clustering Hierarchy

LEACH is dynamic because the job of the cluster head rotates,BS is fixed.LEACH network has two phases in which it operates:

- **The Setup Phase :** cluster heads are chosen
  - During the first step cluster head sends the advertisement packet to inform the cluster nodes that they have become a cluster head
  - The node becomes cluster head for the current round based on energy availability
  - Once the node is elected as a cluster head it cannot become cluster head again until all the nodes of the cluster have become cluster head once.This helps is balancing the energy concumption

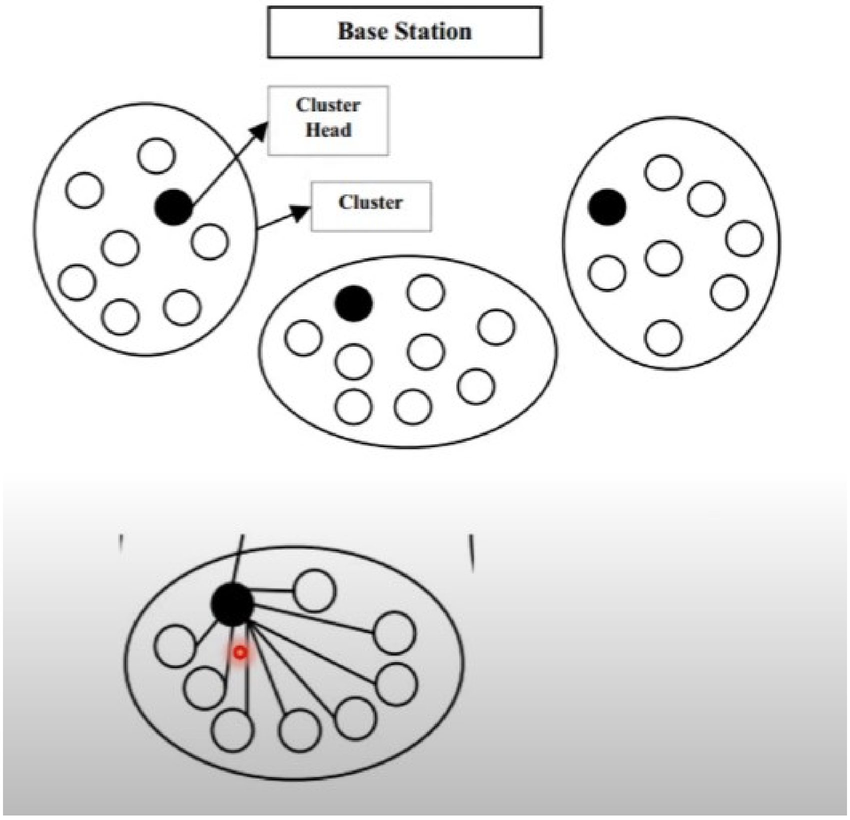

FIGURE 3.5: Setup Phase (Cluster Head Advertisement)

- The non cluster head nodes receive the cluster head advertisement andthen send join request to the cluster head informing that they are the members of the cluster under that cluster head as in Fig 3.6.
- These non cluster head nodes saves a lot of energy by turning off their transmitting all the time and turn it ON only when they have something to transmit to the cluster head

- In the third step, each of the chosen cluster head creates a transmission schedule for the member nodes nodes of their cluster
- TDMA Schedule is created according to the number of nodes in the cluster.Each then transmit its data in the allocated time schedule

- **The Steady State :** with chosen CH, the data is transmitted between nodes finally to the BS.
- Collect information and transmit it to their respective cluster heads.

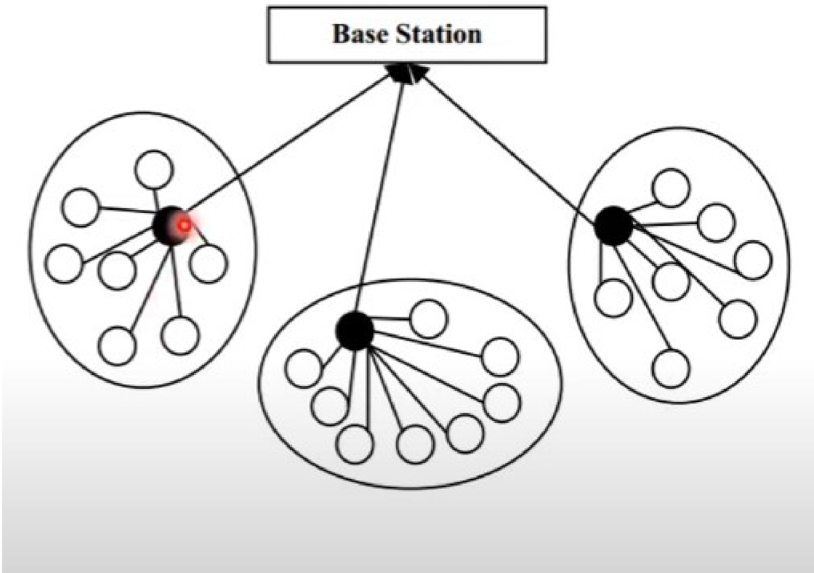

FIGURE 3.6: Setup Phase (Cluster Setup)

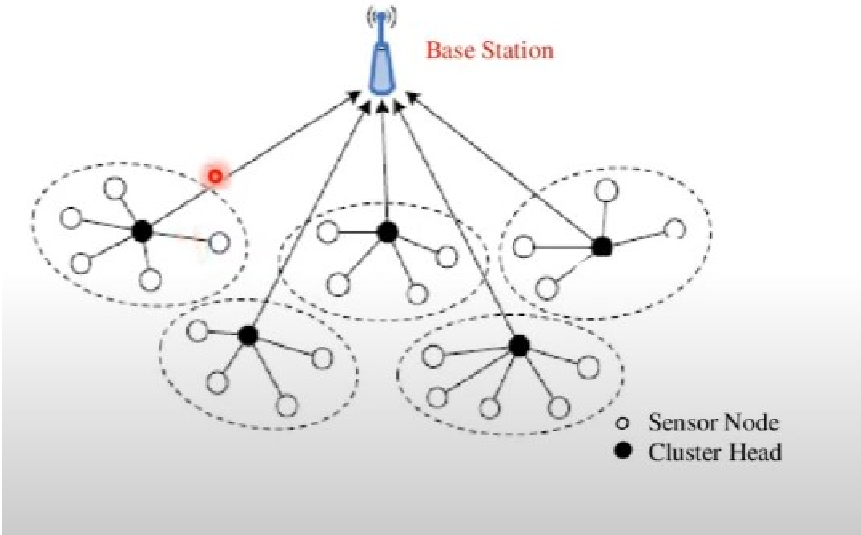

FIGURE 3.7: Setup Phase (Creation of transmission schedule)

- Data Fusion: Cluster heads collect data from their member nodes and perform data aggregation and fusion operations to minimize redundant data and reduce the amount of data transmitted to the base station.

- Data Transmission: Cluster heads transmit the aggregated data to the base station or a designated sink node using multi-hop communication.

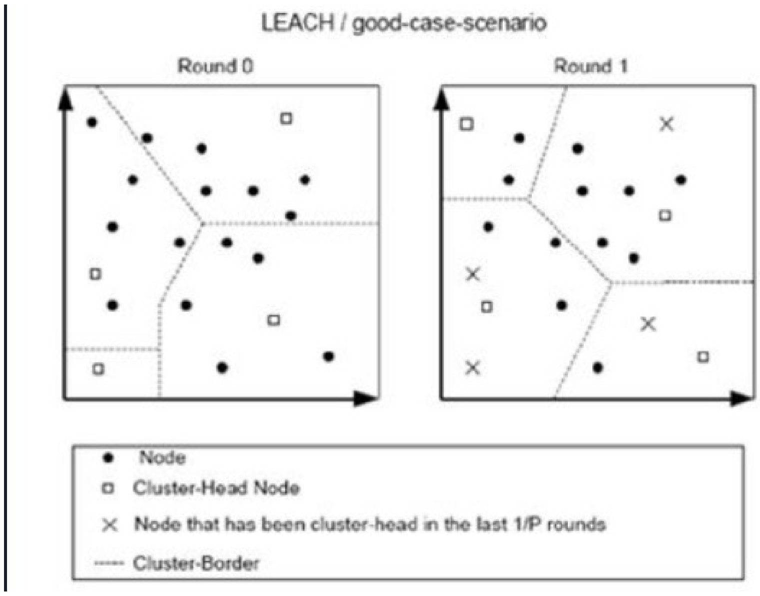

FIGURE 3.8: Leach

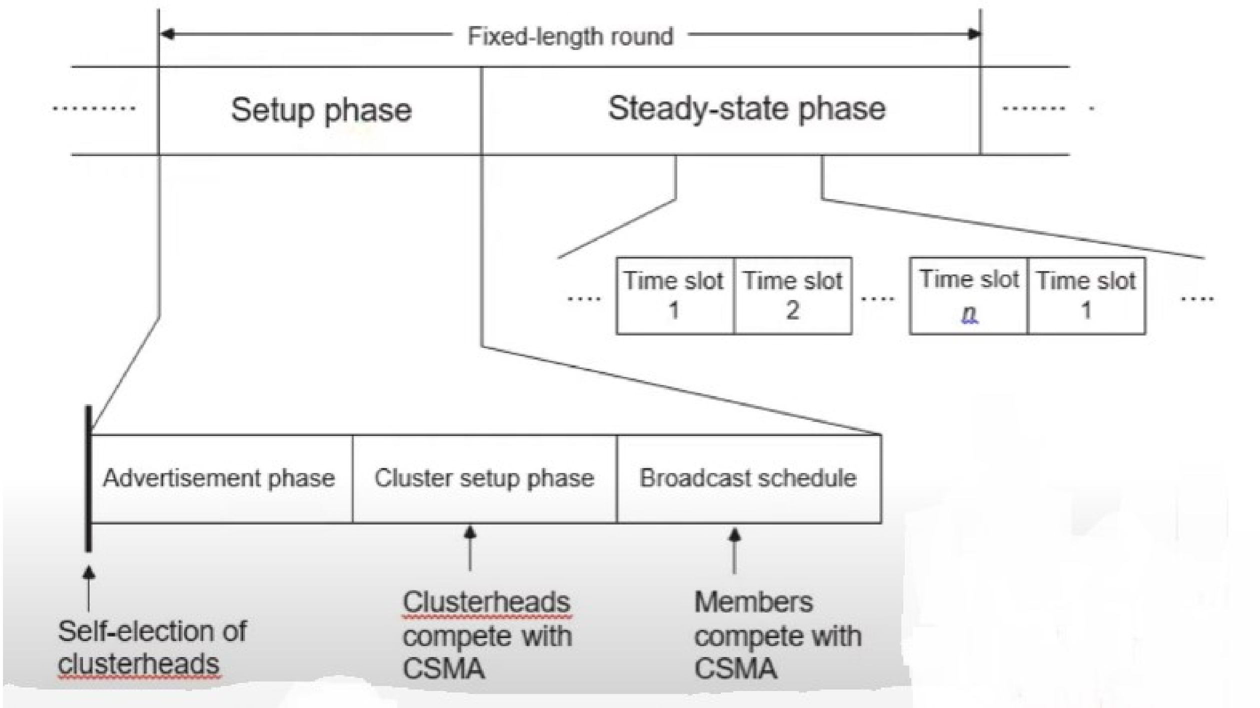

FIGURE 3.9: Organisation Of Leach Rounds

- Cluster heads may employ routing protocols to relay data through intermediate nodes towards the destination.

3.5 Proposed Approach

3.5.1 Random Distribution Of Nodes

A large number of sensor nodes (or sensors) and one or more management nodes, typically called sinks or base stations. Sensors monitor physical phenomena and produce sensory data. A sink, on the other hand, collects data from sensors. Such a network can be used to monitor the environment, detect, classify and locate specific events, and track targets over a specific region of interest (RoI).

Sensor placement is a fundamental issue in wireless sensor networks (WSNs). The sensor-positions can be predetermined to guarantee the quality of surveillance provided by the WSN.However, in remote or hostile sensor field, randomised sensor placement often becomes the only option.Categorise random placement strategies into simple and compound. An empirical study has been carried out yielding a detailed analysis of random deployment intrinsic properties, such as coverage, connectivity, fault-tolerance, and network lifespan.The performance of a hybridisation of the simple diffusion model that places a large number of nodes around the sink and the constant diffusion that provides high coverage and connectivity rates.

In harsh environments such as a battlefield or a disaster region, deterministic deployment of sensors is very risky and infeasible. In this case, random deployment often becomes the only option.

To generate a random distribution of nodes in a wireless sensor network (WSN), you can follow these steps:

- Determine the size and shape of the area where the nodes will be deployed. This will typically be a rectangular region, but it could also be circular or some other shape.
- Generate a set of random coordinates within the deployment area. This can be done using the 'rand' function in Matlab or a similar function in another programming language. The number of coordinates generated should be equal to the number of nodes in the WSN.
- Optionally, you can add constraints to the random node placement. For example, you might want to ensure that nodes are not placed too close to each other, or that they are distributed evenly throughout the deployment area.

- Once you have generated the random node coordinates, you can use them to initialize the positions of the nodes in your WSN simulation or deployment.

3.5.2 Delaunay Triangulation

In computational geometry, a Delaunay triangulation for a given set P of discrete points in a general position is a triangulation DT(P) such that no point in P is inside the circumcircle of any triangle in DT(P). Delaunay triangulations maximize the minimum of all the angles of the triangles in the triangulation; they tend to avoid sliver triangles.

We construct the Delaunay Triangulation to indicate the topology of a WSN. As we know, the circumcircle of a Delaunay triangle is the circle that contains no vertex of any other triangle, which is called an empty circle in the remainder of the paper.Consider the coverage holes inside the sensor network, rather than those that are outside the boundary of the network. If the radius of an empty circle is greater than the sensing radius of sensor nodes, the empty circle definitely contains some region that is uncovered by any node of the network. Then, we develop a method for clustering the nodes that enclose the same coverage hole into separated groups, which are the corresponding groups of boundary nodes for the different coverage holes.By identification of these coverage holes for the deployment of movable nodes to mitigate coverage hole issue.

To perform Delaunay triangulation on a set of nodes in a wireless sensor network (WSN), you can follow these steps:

- Determine the set of nodes that you want to triangulate. This could be all the nodes in the WSN, or a subset of nodes.
- Extract the coordinates of the nodes. You will need the x- and y-coordinates of each node in order to perform the triangulation.
- Use a Delaunay triangulation algorithm to triangulate the nodes. There are several options for doing this, including using a library or function in a programming language such as Matlab.
- Extract the triangulation from the output of the Delaunay triangulation algorithm. This will typically be a list of triangles, where each triangle is defined by the indices of the nodes that make up the triangle.
- Use the triangulation to perform further analysis or visualization. For example, you could plot the triangulation using the coordinates of the nodes, or use the triangles to calculate the connectivity of the nodes in the WSN.

3.5.3 Coverage Hole Detection

It is very difficult to have a sensor network with 100% sensing coverage. If a region in which no point is covered by any node exists in the WSN, a coverage hole emerges. Coverage holes may appear anywhere in the field of WSN at any time due to many reasons. Firstly, sensor nodes in WSN are usually deployed in remote and hostile environments and left unattended for a relatively long period of time. It is possible that some nodes run out of energy and no more join in the network. The affected region gives rise to a hole. Another reason of coverage hole formation is the physical destruction of sensor nodes. Furthermore, there may be some areas in the region of interest where it is impossible for sensor nodes to normally operate. For example, some sensors deployed in the indoor environment cannot find the existence of targets due to obstacles such as walls and furniture that cause the network to be partitioned and uncovered. Moreover, when the cost constraint makes it impossible for a precise deployment, sensor nodes are randomly deployed in the field. In random deployment, the density of sensors in some parts of ROI becomes lower than it is required. Therefore, it will incur the emergence of coverage holes in these parts. These holes are formed due to non-uniform deployment among the ROI. Therefore, coverage holes may appear right after a random deployment and at any time during the lifetime of a sensor network.

To detect coverage holes in a wireless sensor network (WSN), you can follow these steps:

- Define the coverage area for each node in the WSN. This will typically be a circular region centered at the node, with a radius equal to the transmission range of the node.
- Identify the set of nodes that are active, meaning they are currently transmitting or receiving data.
- For each active node, calculate the coverage area of the node. This can be done by plotting the coverage area on a grid or map of the WSN deployment area.
- Check for areas of the deployment region that are not covered by any active node. These areas are potential coverage holes.
- Optionally, you can use additional information such as the signal strength or quality at different locations in the deployment region to confirm the presence of coverage holes and to estimate their size and shape.

Mitigating coverage holes in a WSN can have several benefits, including:

- Improved coverage and connectivity: By filling in coverage holes, the WSN can provide more consistent and reliable coverage across the entire network, enabling better communication and data transfer between nodes.
- Enhanced system performance: By ensuring that all parts of the network are adequately covered, the WSN can operate more efficiently and effectively, leading to improved performance and reliability.
- Increased energy efficiency: By reducing the need for nodes to transmit over longer distances or through multiple hops, coverage hole mitigation can help to conserve energy and extend the lifetime of the nodes.
- Enhanced security: By eliminating coverage holes, the WSN can be more resistant to attacks or disruptions, as there are fewer opportunities for attackers to exploit gaps in the network.

3.5.4 Coverage Hole Mitigation

By means of delaunay triangulation we are able to obtain holes present in the network and these coverage hole are patched by movable nodes placed in these coverage hole area.And thus covering the hole area resulting of covering the entire network area.

Placing the mobile nodes in the hole area and then retriangulating again including recently placed movable nodes in the hole area to ensure the coverage hole is patched .To improve node participation and coverage in a wireless sensor network (WSN) with movable nodes, you can follow these steps:

- Identify the coverage holes in the WSN, as described in the previous section.
- Determine which of the nodes in the WSN are movable and have the ability to move to new locations.
- The strategy employed to enhance coverage in the identified coverage holes involves strategically relocating movable nodes to specific locations. This relocation process aims to optimize node placement within the holes and ensure improved signal coverage. One approach within this strategy involves identifying triangles in the triangulation where the length of an edge exceeds two times the sensing radius. In such cases, a mobile node is placed at the identified location to address the coverage deficiency.

- Implement the node movement strategy in the WSN. This could be done by sending commands to the nodes to move to new locations, or it could involve more advanced techniques such as reinforcement learning or evolutionary algorithms.
- Monitor the performance of the WSN with the movable nodes and adjust the node movement strategy if there of any need.

3.5.5 Proposed Clustering Approach

The proposed system is designed to enhance the efficiency of wireless sensor networks through an improved clustering protocol. The entire area of the network is divided into four equal quadrants, with each quadrant having a designated cluster head. This division ensures that each sensor node communicates directly with the nearest cluster head. This localized routing helps in efficient data aggregation and reduces unnecessary transmissions, leading to optimized network performance.The nodes which are closer to the base station does not participate in clustering they directly send packets to the base station as sending through the cluster head is overhead in the given scenario.

Once the packets are received by the cluster heads, they are further routed towards the base station. The cluster heads act as intermediate nodes and facilitate the transmission of aggregated data from their respective quadrants to the base station. This hierarchical approach ensures that the data from multiple sensors is efficiently collected and transmitted to the central base station for further processing and analysis.

By employing this improved clustering protocol, the proposed system aims to minimize energy consumption, reduce network congestion, and improve the overall efficiency of the wireless sensor network. The division of the network into quadrants and the utilization of cluster heads streamline the data flow and enable more effective communication within the network, ultimately enhancing its performance and capabilities.

3.5.6 Proposed Routing Method

After identifying the minimum spanning tree and establishing the path between the base station and the nearest cluster head, the routing protocol sets up a communication route for data transmission. This route enables efficient and direct communication between the base station and the entire network.The use of a minimum spanning tree ensures that the routing protocol selects the most optimal path, considering both the distance and connectivity between the cluster heads. This approach minimizes the overall energy consumption and reduces communication delays within the network.

Furthermore, the routing protocol dynamically adjusts the routing path based on changes in the network topology or the addition/removal of cluster heads. This flexibility ensures adaptability to the evolving network conditions, allowing for efficient and reliable data transmission at all times.

By utilizing the minimum spanning tree to connect cluster heads and establishing a direct path to the base station, the routing protocol enhances network efficiency and reduces the chances of congestion or data loss. This approach optimizes the utilization of network resources and improves the overall performance of the WSN, making it well-suited for various applications that require reliable and efficient data communication in wireless sensor networks.

3.6 Features

3.6.1 Mitigate Coverage Hole Issue

Mitigating coverage hole issues in wireless networks is crucial for ensuring reliable and seamless connectivity. Coverage holes, which are areas with weak or no signal coverage, can lead to disruptions in communication and a poor user experience. To address this, several strategies can be implemented. Optimizing signal strength through techniques like adjusting transmit power levels and optimizing antenna placement helps extend coverage. Proper placement of access points (APs) and strategic channel planning minimize interference and maximize coverage. Mesh networking enables APs to communicate and fill coverage gaps, while seamless roaming and handoff ensure uninterrupted connectivity during client mobility. Regular wireless site surveys, load balancing, and network monitoring help identify and address coverage holes proactively. Additionally, the use of repeaters or range extenders, along with regular maintenance and upgrades, can further improve coverage and mitigate coverage hole issues. By employing these measures, wireless networks can provide comprehensive and reliable coverage, enhancing the overall performance and user satisfaction.In proposed system coverage hole issue is mitigated by placing mobile nodes.

3.6.2 Improved Node Participation

Improved node participation in wireless sensor networks is essential for enhancing network efficiency and achieving optimal performance. Node participation refers to the active involvement of sensor nodes in network operations, such as data sensing, processing, and forwarding. By improving node participation, several benefits can be achieved. Firstly, it leads to better data collection and accuracy, as more nodes actively contribute to sensing the environment and gathering relevant information. This enables more comprehensive monitoring and analysis of the target area or application domain. Secondly, improved node participation enhances network resilience and fault tolerance. When multiple nodes are actively participating, the network becomes more robust against node failures or malfunctions, as other nodes can compensate for the lost functionality. Additionally, enhanced node participation enables efficient resource utilization. By distributing tasks and responsibilities among a larger number of active nodes, the workload can be effectively shared, reducing the burden on individual nodes and prolonging the network's overall lifetime. Moreover, increased node participation promotes collaborative data processing and aggregation, allowing nodes to collectively analyze and summarize data before transmission, thereby reducing redundant or unnecessary data transmissions and conserving energy. Overall, improved node participation plays a vital role in optimizing the performance, reliability, and energy efficiency of wireless sensor networks, ultimately contributing to the successful deployment and operation of various applications and services.

3.6.3 Improved Clustering Protocol

The proposed system works with an improved clustering protocol, in which the whole area is divided into 4 equal quadrants.Each quadrants consists of a single cluster head. From each sensor nodes packets are routed to nearest cluster heads. The packets are then routed to the base station from each cluster heads.

3.6.4 Shortest Path Routing

Shortest path routing using minimum spanning tree (MST) is a fundamental technique employed in wireless networks to efficiently route data packets between nodes. The MST algorithm constructs a tree that connects all nodes in the network with the minimum total edge weight. By utilizing this tree structure, shortest path routing can be achieved by selecting the path with the smallest cumulative edge weight from a source node to a destination node. This approach offers several advantages in wireless networks. Firstly, it reduces the complexity and overhead associated with maintaining a complete routing table, as only a subset of edges in the MST needs to be considered for routing decisions. This leads to improved scalability and reduced memory requirements. Additionally, MST-based routing ensures that the selected paths are loop-free, preventing packet collisions and reducing network congestion. Moreover, MST-based routing is resilient to node failures, as alternative paths can be readily identified within the tree structure. This increases the network's fault tolerance and robustness. Furthermore, the use of MST promotes energy efficiency, as the shortest paths typically require less transmission power and reduce energy consumption compared to longer routes. Overall, shortest path routing using minimum spanning tree provides an effective and efficient mechanism for routing data packets in wireless networks, enabling reliable and optimized communication among nodes.

3.6.5 Improved network efficiency

Improved network efficiency is a critical goal in wireless sensor networks to optimize resource utilization and enhance overall performance. Several techniques can be employed to achieve improved network efficiency. In proposed system, by organizing nodes into clusters and designating cluster heads, data aggregation and fusion can be performed at cluster heads, minimizing redundant transmissions and conserving energy. Additionally, the use of adaptive and dynamic routing algorithms that adapt to changing network conditions and traffic patterns can improve routing efficiency.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 DFD (Data Flow Diagram)

4.1.1 DFD Of leach protocol

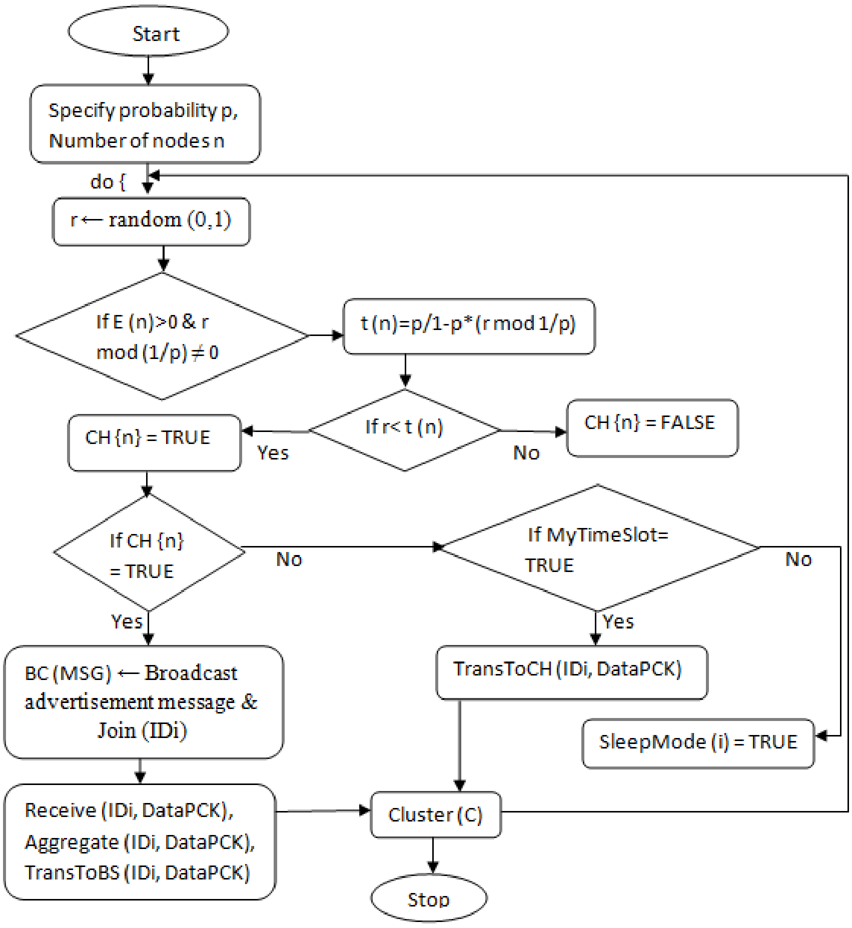

FIGURE 4.1: Data Flow Diagram of LEACH Protocol

The Data Flow Diagram (DFD) for the LEACH (Low Energy Adaptive Clustering Hierarchy) protocol illustrates the flow of data and control messages among various components involved in the protocol. The diagram shows the major components of the LEACH protocol and their interactions.

- Base Station:

  Sends control messages to Cluster Heads. Receives aggregated data from Cluster Heads.

- Cluster Heads:

  Receive control messages from the Base Station. Perform data aggregation from Cluster Members. Send aggregated data to the Base Station.

- Cluster Members:

  Collect data from the Sensor Nodes. Send collected data to the Cluster Heads.

- Sensor Nodes:

  Sense and gather data from the environment. Send collected data to their respective Cluster Members.

The DFD demonstrates the flow of control messages from the Base Station to the Cluster Heads and the flow of data from the Sensor Nodes to the Cluster Heads and finally to the Base Station. The Cluster Heads play a crucial role in aggregating data from their Cluster Members before forwarding it to the Base Station. This data flow helps in achieving energy efficiency and minimizing communication overhead in the wireless sensor network.

Overall, the DFD represents the data and control flow in the LEACH protocol, showcasing the interaction between the Base Station, Cluster Heads, Cluster Members, and Sensor Nodes.

4.1.2 DFD of delaunay triangulation

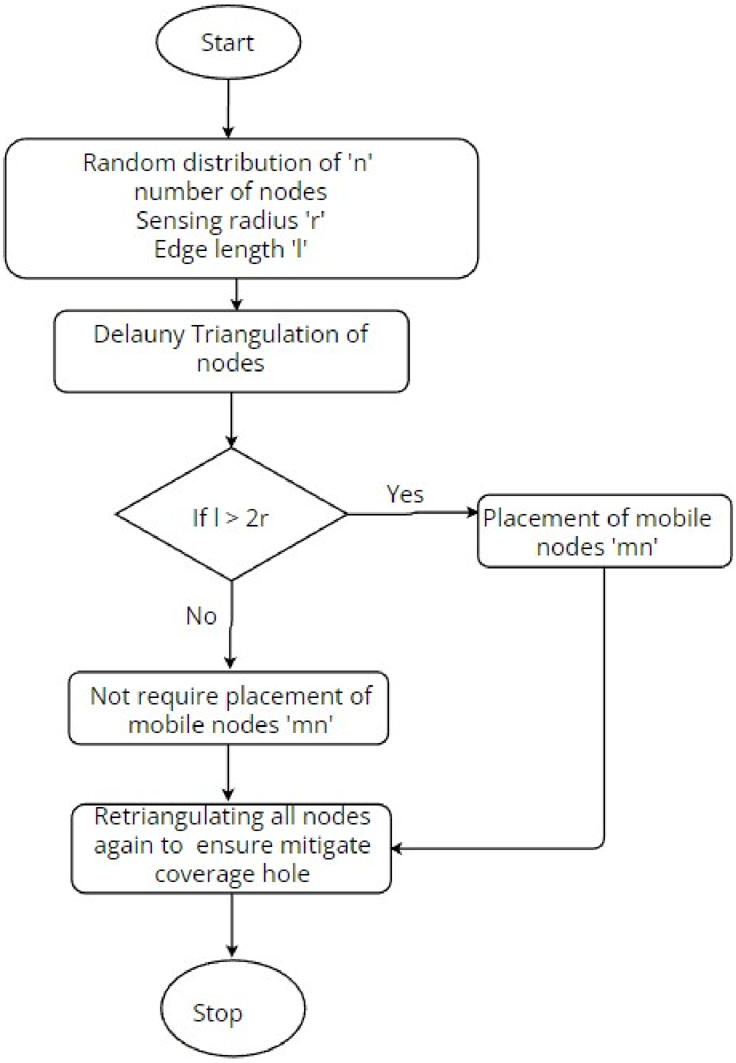

Figure 4.2: Data Flow Diagram of Proposed delaunay triangulation

4.1.3 DFD of Proposed System

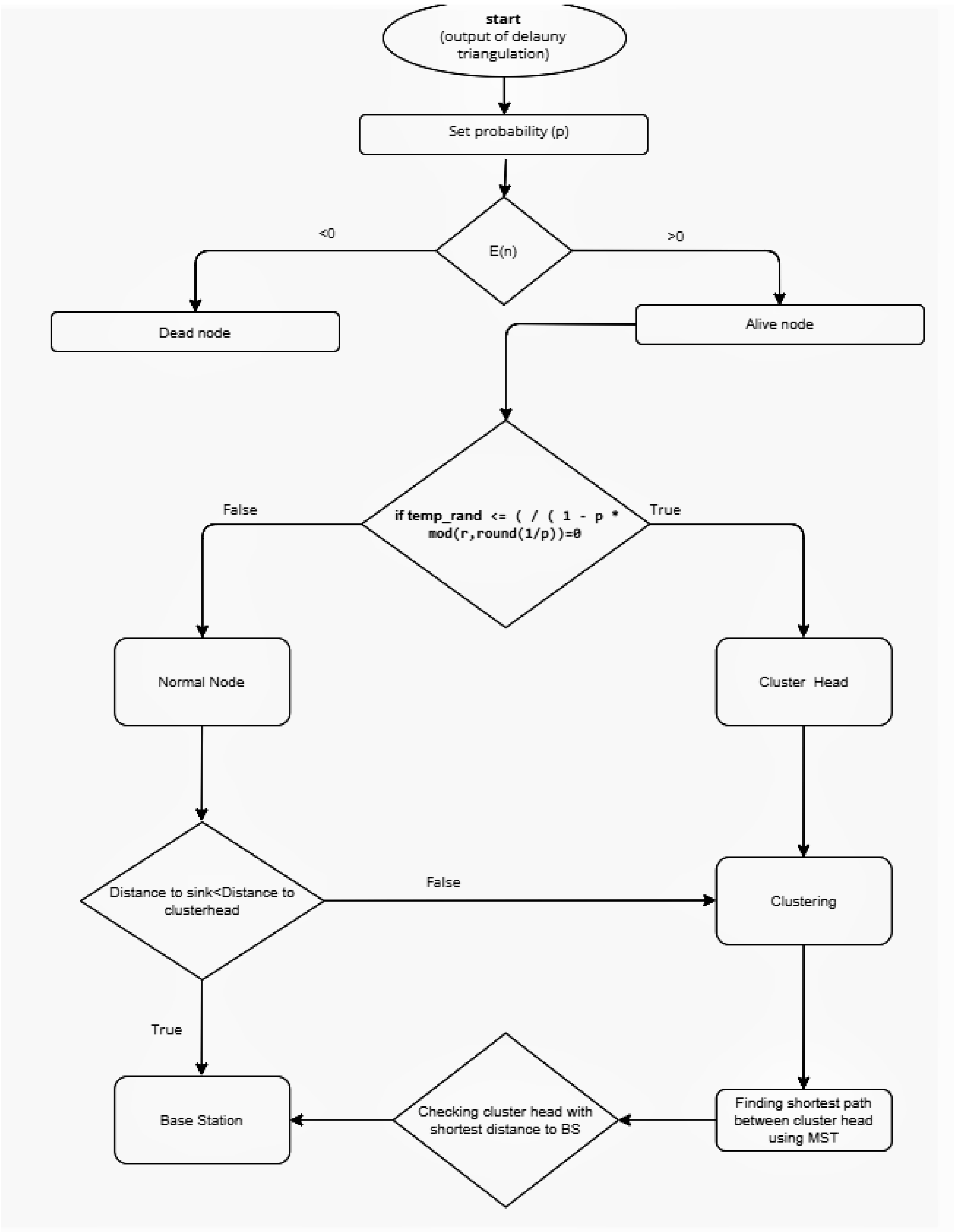

FIGURE 4.3: DFD of Proposed system After the Delauny Triangulation

4.2 Algorithm

4.2.1 Algorithm for Delaunay Triangulation

1. Create an empty triangulation T.
2. Compute the convex hull of the points in P. This forms the outer boundary of the triangulation. Add the boundary edges to T.
3. Sort the points in P based on their x-coordinate in ascending order.
4. For each point p_i in P:
   (a) Find the neighboring points of p_i that lie to the right of p_i in the sorted order. Let's call these neighboring points $N(p_i)$.
   (b) For each point p_j in $N(p_i)$:
      i. Check if the circumcircle of triangle (p_i, p_j, p_k) contains no other points in P, where p_k is any other point in P.
      ii. If the circumcircle is empty, create a new triangle with vertices (p_i, p_j, p_k) and add it to T.
5. Output: Triangulation T, which represents the Delaunay triangulation of the input points P.

4.2.2 Algorithm for identifying coverage hole and patching it

1. Create an empty list of coverage holes.
2. For each edge e in the triangulation T:
   (a) Calculate the length of the edge $|e|$.
   (b) Calculate the radius of the sensor node r as half of the communication range.
   (c) If $|e| > 2 \times r$, a coverage hole is detected.
   (d) Add the edge e to the list of coverage holes.
3. For each coverage hole in the list:
   (a) Calculate the midpoint of the edge.
   (b) Place a mobile node at the midpoint to patch the coverage hole.
4. Output: The patched sensor network with no coverage holes.

4.2.3 Algorithm for minimum spanning tree

1. Initialize a matrix `distMatrix` with `inf` values of size (`cluster`-1, `cluster`-1), where `cluster` represents the number of clusters.

2. Calculate distances between cluster heads:

   (a) Use nested loops to iterate over each pair of cluster heads.

   (b) Calculate the Euclidean distance between the positions of the cluster heads and store the result in the corresponding cells of `distMatrix`.

   (c) Update the symmetric cells to maintain the distance matrix property.

3. Minimum spanning tree algorithm:

   (a) Initialize a binary array `visited` of size (1, `cluster`-1) to keep track of visited cluster heads. Set the first cluster head as visited (`visited`(1) = 1).

   (b) Initialize a matrix `routeMatrix` of size (`cluster`-1, `cluster`-1) to store the connectivity information between cluster heads.

   (c) Start a loop until all cluster heads are visited (`sum(visited) < cluster-1`).

   (d) Find the cluster head with the shortest distance to the visited set:

      i. Initialize `minDist` with a large value (`inf`) and `minIdx` as 0.

      ii. Iterate over the visited cluster heads (`if visited(i)`).

      iii. Among the non-visited cluster heads (`if ~visited(j)`), find the one (`j`) with the shortest distance to the visited cluster head (`distMatrix(i, j)`).

      iv. Update `minDist` and `minIdx` if a shorter distance is found.

   (e) Connect the cluster head with the visited set:

      i. Set `routeMatrix(minIdx, find(visited == 1, 1)) = 1` to indicate a connection between `minIdx` and the first visited cluster head found (`find(visited == 1, 1)`).

      ii. Mark `minIdx` as visited (`visited(minIdx) = 1`).

4. Plotting the route:

   (a) Iterate over each pair of cluster heads in `routeMatrix`.

   (b) If there is a connection (`routeMatrix(i, j) == 1`), plot a line between the positions of the cluster heads with a red dashed line.

4.2.4 Algorithm for leach protocol

1. Initialization:

   (a) Set the total number of nodes in the network (N) and the desired percentage of cluster heads (P).

   (b) Create N nodes and randomly distribute them in the network area which is divided into four equal grids

   (c) Initialize all nodes with energy levels and other parameters.

2. Round Setup:

   (a) Each node broadcasts an advertisement message to announce its presence and energy level.

   (b) Each non-cluster head node calculates a probability value to become a cluster head using the formula:

      - `probability = P / (1 - P * (round mod (1/P)))`

   (c) Each non-cluster head node selects a random number between 0 and 1 and becomes a cluster head if the random number is less than or equal to the calculated probability.

3. Cluster Formation:

   (a) A cluster head is formed on the condition that only one cluster head is formed in a grid

   (b) Each non-cluster head node selects the nearest cluster head and joins its cluster.

   (c) Each cluster head creates a cluster and sends a message to all the nodes in its cluster to join.

4. Data Transmission:

   (a) Each node collects data from its environment.

   (b) Each node aggregates the data received from its cluster members.

   (c) Each cluster head performs data fusion and generates a single aggregated message.

   (d) Each cluster head sends the aggregated message to the base station (sink) using multi-hop communication.

5. Energy Management:

(a) Each node monitors its energy level.
(b) If a node's energy level drops below a certain threshold, it will request to become a cluster head in the next round.
(c) Cluster heads rotate periodically to distribute the energy consumption evenly across the network.

6. End of Round:
   (a) The current round ends, and the next round begins with the Round Setup phase.
7. Repeat the above steps until the desired simulation time or network lifetime is reached.

4.3 Implementation Code

In implementation phase consisting of initially random distribution of sensor to a predefined area where of having a single base station or sink.And later implementation of delaunay triangulation for identifying coverage holes and optimally placing movable nodes in these coverage holes and locally retriangulate.Then implemeting Gaussian network model for shortest path routing.
GitHubRepository: https://github.com/SID41214/An energy efficient wsn with EQDC protocol and improved node partcipation

4.3.1 Declaration of variables

```
close all;
clear;
clc;
xm=100;
ym=100;
x=0; % added for better display results of the plot
y=0; % added for better display results of the plot
p=0.1; % Probability of node transmitting data
Eo=0.5; % initial energy of node
ETX=50*0.000000001;  % represent the energy consumed during transmission
ERX=50*0.000000001 % represent the energy consumed during receiving
Efs=10e-12;%Energies required to transmit a bit over a unit distance
% in free space condition
Emp=0.0013e-12;%Energies required to transmit a bit over a unit
% distance in  multi-path condition
EDA=5*0.000000001;
rmax=2000;
do=sqrt(Efs/Emp);
Et=0;
% Number of Nodes in the field %
global n;
n=100;
global mn;
mn=10;
%loop variable to determine the termination of the network
operatingNodes=n+mn;
% Number of Dead Nodes in the beggining %
dead_nodes=0;
% Coordinates of the Sink (location is predetermined in this simulation)
sinkx=50;
sinky=120;
LeaderNodeThreshold=.03;
global SN;
global MN;
global totalNodes;
totalNodes=n+mn+1;
```

4.3.2 Creating wireless sensor network

```
%%% Creation of the Wireless Sensor Network %%%
pts=[n,2];
pts=[rand([n,1])*xm,rand([n,1])*xm];
disp(pts);
 %%% randomly generating mobile node points
mnpts=[mn,2];
mnpts=[rand([mn,1])*xm,rand([mn,1])*xm];
%disp(pts);
 % X-axis coordinates of sensor node
 for i=1:mn
    MN(i).x=mnpts(i,1);
    MN(i).y=mnpts(i,2);
    MN(i).radius=50; %assuming 20meters
    MN(i).sel=0;
end

%% Adding base station to the sensor node list

%adding sink point to pts array

%aading sink point to pts array
sink=[sinkx,sinky];
disp('sink is..');
disp(sink);
pts=[sink;pts];
disp(pts);

% Plotting the WSN %
for i=1:n+1

   SN(i).id=i; % sensor's ID number
   SN(i).radius=20; %assuming 20 meters
   SN(i).x=pts(i,1);
   SN(i).y=pts(i,2);

     hold on;
    figure(1)
    plot(x,y,xm,ym,SN(i).x,SN(i).y,'ob',sinkx,sinky,'*r');
    title 'Wireless Sensor Network';
    xlabel '(m)';
    ylabel '(m)';
end

%trying to create coordinates array and y coordinates array seperately to
%give coordinates in the grpah
xCoord=(n+1+mn);
yCoord=(n+1+mn);

for i=1:n+1
   xCoord(i)=SN(i).x;
   yCoord(i)=SN(i).y;
end
```

4.3.3 Delaunay triangulation

```
 % Calculates Euclidaean Distance Between Each Node and the Sink
 % (Base Station)
 for i=1:n+1
    dts(i)=sqrt((sinkx-xCoord(i))^2 + (sinky-yCoord(i))^2);
    T(i)=dts(i);
 end

%Delaunay triangulating all the sensor nodes

DT = delaunayTriangulation(pts);
figure(2)

e=DT.edges;
plot(sinkx, sinky, 'r*', 'markersize', 5);
hold on

pl=triplot(DT,'-ob');
hold on
plot(pts(1),pts(2),'-r','LineWidth',2);

%nodes to be displayed
title 'Delaunay Triangulation of the network with Sink and Static sensor nodes';

DT = delaunayTriangulation(pts);
figure(3)
hold on
for i=1:mn

     plot(x,y,xm,ym,MN(i).x,MN(i).y,'+r');

end
pl=triplot(DT,'-ob');
title 'Current positions of mobile sensor nodes'

%identifying gaps

%identifying edge lengths of DT

edgelengths = sqrt(sum((DT.Points(e(:,1),:) - DT.Points(e(:,2),:)).^2,2));
%disp('edge lengths')
disp(edgelengths);
%disp(edgelengths);
noOfEdges=size(edgelengths);
mnx=(mn);
mny=(mn);
pending_edge=(0);
%checking if it is greater than 2*radius of static node
 global k;
 k=0;
 in=1;
 selectedEdge=[];
for i=1:noOfEdges
 edgeX1=DT.Points(e(i,1),1);
```

```
 edgeY1=DT.Points(e(i,1),2);
 edgeX2=DT.Points(e(i,2),1);
 edgeY2=DT.Points(e(i,2),2);

 if k<=mn

if edgelengths(i)> 2*SN(1).radius

%check if the found hole is patchable or not
   if 2*MN(1).radius>abs(edgelengths(i)-2*SN(1).radius)

 %  disp('patchable hole found');
       selectedEdge=[selectedEdge,i];

 if k~=mn
 k=k+1;
 end

 mnx(k)=(edgeX1+edgeX2)/2;
 mny(k)=(edgeY1+edgeY2)/2;

   end

end
end

   MNx=(mn);
   MNy=(mn);

end
disp(k);

if k<mn
    for i=1:length(edgelengths)

 edgeX1=DT.Points(e(i,1),1);
 edgeY1=DT.Points(e(i,1),2);
 edgeX2=DT.Points(e(i,2),1);
 edgeY2=DT.Points(e(i,2),2);\

        if edgelengths(i)>30 && k<mn && ismember(i,selectedEdge)==0
            k=k+1;
            selectedEdge=[selectedEdge,i];
             mnx(k)=(edgeX1+edgeX2)/2;
               mny(k)=(edgeY1+edgeY2)/2;
        end
    end
end

%replot with mobile nodes moved to the positions according to the
%conditions

DT = delaunayTriangulation(pts);
figure(5)
```

```
pl=triplot(DT,'-ob');
title 'Movable sensor nodes placed at the mid points of the longest edges';
hold on;

for i=1:k

     plot(mnx(i),mny(i),'+r');
     pts(end+1,1)=mnx(i);
     pts(end,2)=mny(i);

end
disp(pts);
%graph is to be created here
newDT = delaunayTriangulation(pts);
%calling edges to get this delaunay graph

e=newDT.edges;
%creating grpah object to get graph of DT and minimum spanning tree with
%all mobile nodes placed finally

%xCood and yCoord list to be updated with mobile nodes coordinates

for i=1:mn

    xCoord(n+1+i)=mnx(i);
    yCoord(n+1+i)=mny(i);

end

for i=1:mn

   SN(n+1+i).x=pts(n+1+i,1);
   SN(n+1+i).y=pts(n+1+i,2);

 end

newDT = delaunayTriangulation(pts);

%calling edges to get this delaunay graph
e=newDT.edges;
%creating grpah object to get graph of DT and minimum spanning tree with
%all mobile nodes placed finally

figure(6)
pl=triplot(newDT,'-ob');
title 'Retriangulating the network with mobile nodes'
hold on;
```

4.3.4 Quadrant division,finding clusterheads and clustering nodes

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%       Equi-Quadrant Division Clustering (EQDC) protocol        %%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for h=1:1
    S(n+1).xd=sinkx;
    S(n+1).yd=sinky;
    Et=0;
    %% composing sensors
    for i=1:1:n
        XR(i)=S(i).xd;
        YR(i)=S(i).yd;
        distance=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );
        S(i).distance=distance;
        S(i).G=0;
        %initially there are no cluster heads only nodes
        S(i).type='N';
        S(i).E=Eo;
        Et=Et+S(i).E;

    end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
countCHs=0;  %variable, counts the cluster head
cluster=1;  %cluster is initialized as 1
flag_first_dead=0; %flag tells the first node dead
flag_half_dead=0;  %flag tells the 10th node dead
flag_all_dead=0;  %flag tells all nodes dead
first_dead=0;
half_dead=0;
all_dead=0;
allive=n;
%counter for bit transmitted to Bases Station and to Cluster Heads
packets_TO_BS=0;
packets_TO_CH=0;
packets_TO_BS_per_round=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for r=0:1:rmax
    figure(20)
    clf
    r
    title('Normal nodes:Blue||CH:Red||Dead:Empty circle||Round no:'+string(r));
    hold on

    line([50,50],[0,120],'color','green');
    line([100,100],[100,0],'color','green');
    line([0,100],[0,0],'color','green');
    line([0, 100], [60, 60], 'color', 'green');
    line([0,0],[0,100],'color','green');
    line([0, 0], [0, 120], 'color', 'green');
    line([0, 100], [120, 120], 'color', 'green');
    line([100, 100], [0, 120], 'color', 'green');
```

```
    packets_TO_BS_per_round=0;
    %Operations for epochs
    if(mod(r, round(1/p) )==0)
        for i=1:1:n
            S(i).G=0;
            S(i).cl=0;
        end
    end

    %Number of dead nodes
    dead=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     for i=1:1:n
        %checking if there is a dead node
        if (S(i).E<=0)
            plot(S(i).xd,S(i).yd,'o');
            dead=dead+1;
            if (dead==1)
              if(flag_first_dead==0)
                 first_dead=r;
                 flag_first_dead=1;
              end
            end
            if(dead==0.5*n)
              if(flag_half_dead==0)
                  half_dead=r;
                  flag_half_dead=1;
              end
            end
            if(dead==n)
              if(flag_all_dead==0)
                  all_dead=r;
                  flag_all_dead=1;
              end
            end
        end
        if S(i).E>0
            S(i).type='N';
        end
    end
    STATISTICS.DEAD(h,r+1)=dead;
    STATISTICS.ALLIVE(h,r+1)=allive-dead;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %head finding
    ch1=0;ch2=0;ch3=0;ch4=0;ch=0;
    countCHs=0;
    cluster=1;
    for i=1:1:n
    if(S(i).E>0)
    temp_rand=rand;
    if ( (S(i).G)<=0)

    %Election of Cluster Heads for normal nodes
```

```
    if ( temp_rand <= ( p/ ( 1 - p * mod(r,round(1/p)) )) )

    if(S(i).xd>0 && S(i).xd<=50 && S(i).yd>0 && S(i).yd<=60 && ch1==0)
     ch1=i;
     ch=i;
     countCHs=countCHs+1;
     packets_TO_BS=packets_TO_BS+1;
     packets_TO_BS_per_round=packets_TO_BS_per_round+1;
     PACKETS_TO_BS(r+1)=packets_TO_BS;
    end
    if(S(i).xd>0 && S(i).xd<=50 && S(i).yd>60 && S(i).yd<=120 && ch2==0)
      ch2=i;
      ch=i;
      countCHs=countCHs+1;
      packets_TO_BS=packets_TO_BS+1;
      packets_TO_BS_per_round=packets_TO_BS_per_round+1;
      PACKETS_TO_BS(r+1)=packets_TO_BS;
    end


    if(S(i).xd>50 &&S(i).xd<=100 && S(i).yd>60 &&S(i).yd<=120 && ch3==0)
    ch3=i;
    ch=i;
    countCHs=countCHs+1;
    packets_TO_BS=packets_TO_BS+1;
    packets_TO_BS_per_round=packets_TO_BS_per_round+1;
    PACKETS_TO_BS(r+1)=packets_TO_BS;
    end

    if(S(i).xd>50 && S(i).xd<=100 && S(i).yd>0 && S(i).yd<=60&& ch4==0)
     ch4=i;
     ch=i;
     countCHs=countCHs+1;
     packets_TO_BS=packets_TO_BS+1;
     packets_TO_BS_per_round=packets_TO_BS_per_round+1;
     PACKETS_TO_BS(r+1)=packets_TO_BS;
    end
    if (ch>0)
     S(ch).type='C';
     S(ch).G=round(1/p)-1;
     C(cluster).xd=S(ch).xd;
     C(cluster).yd=S(ch).yd;
     distance=sqrt( (S(ch).xd-(S(n+1).xd) )^2 + (S(ch).yd-(S(n+1).yd) )^2 );
     C(cluster).distance=distance;
     C(cluster).id=ch;
     X(cluster)=S(ch).xd;
     Y(cluster)=S(ch).yd;
     cluster=cluster+1;

%Calculation of Energy dissipated
     distance;
     if (distance>do)
       S(ch).E=S(ch).E- ( (ETX+EDA)*(4000) + Emp*4000*(distance*distance
       *distance*distance ));
       end
```

```
        if (distance<=do)
          S(ch).E=S(ch).E- ( (ETX+EDA)*(4000)  + Efs*4000*
          ( distance * distance ));
            end
            end
           ch=0;
           if (ch1>0 && ch2>0 && ch3>0 &&ch4>0)
             break;
             end
             end
           end
         end
     end
     plot(S(n+1).xd,S(n+1).yd,'o', 'MarkerSize', 12, 'MarkerFaceColor', 'r');
    for i=1:1:n
        if( S(i).type=='N' && S(i).E>0 )
         if(cluster-1>=1)
         min_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );
         min_dis_cluster=0;
         for c=1:1:cluster-1
         temp=min(min_dis,sqrt( (S(i).xd-C(c).xd)^2 + (S(i).yd-C(c).yd)^2 ) );
         if ( temp<min_dis )
         min_dis=temp;
         min_dis_cluster=c;
          end
 end
%Calculating the culsterheads%
        if(min_dis_cluster~=0)
         min_dis;
         if (min_dis>do)
         S(i).E=S(i).E- ( ETX*(4000) +
         Emp*4000*( min_dis * min_dis * min_dis * min_dis));
         end
         if (min_dis<=do)
          S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*( min_dis * min_dis));
          end
          S(C(min_dis_cluster).id).E = S(C(min_dis_cluster).id).E-
          ( (ERX + EDA)*4000 );
         if(S(i).type=='N')
          plot(S(i).xd,S(i).yd,'o','MarkerSize', 5,'MarkerFaceColor','black');
          line([S(i).xd,S(C(min_dis_cluster).id).xd],
          [S(i).yd,S(C(min_dis_cluster).id).yd]);
         elseif(S(i).type=='C')
         plot(S(i).xd,S(i).yd,'o', 'MarkerSize', 8, 'MarkerFaceColor', 'r');
           else
           plot(S(i).xd,S(i).yd,'bo');
           line([S(i).xd,S(n+1).xd],[S(i).yd,S(n+1).yd]);
           end
           packets_TO_CH=packets_TO_CH+1;
           else
           min_dis;
         if (min_dis>do)
         S(i).E=S(i).E- ( ETX*(4000) +
         Emp*4000*( min_dis * min_dis * min_dis * min_dis));
         end
```

```
      if (min_dis<=do)
       S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*( min_dis * min_dis));
        end
        packets_TO_BS=packets_TO_BS+1;
        packets_TO_BS_per_round=packets_TO_BS_per_round+1;
        PACKETS_TO_BS(r+1)=packets_TO_BS;
      % Plot the line to the sink
       plot([S(i).xd, S(n+1).xd], [S(i).yd, S(n+1).yd], 'g--');
       end
        S(i).min_dis=min_dis;
        S(i).min_dis_cluster=min_dis_cluster;
       else
         min_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );
           if (min_dis>do)
            S(i).E=S(i).E- ( ETX*(4000) + Emp*4000*
           ( min_dis * min_dis * min_dis * min_dis));
           end
           if (min_dis<=do)
            S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*
            ( min_dis * min_dis));
           end
           packets_TO_BS=packets_TO_BS+1;
           packets_TO_BS_per_round=packets_TO_BS_per_round+1;

      end
 end

 if(S(i).type=='C'&&S(i).E>0)
 plot(S(i).xd,S(i).yd,'o', 'MarkerSize', 8, 'MarkerFaceColor', 'r');
 elseif(S(i).type~='N'&&S(i).E>0)
 plot(S(i).xd,S(i).yd,'o', 'MarkerSize', 5,'MarkerFaceColor','black');
 elseif S(i).type == 'N' && S(i).E > 0 && S(i).min_dis_cluster == 0
 plot(S(i).xd, S(i).yd, 'o','MarkerSize', 5,'MarkerFaceColor','black');
       line([S(i).xd,S(n+1).xd],[S(i).yd,S(n+1).yd],'Color','black');

 end
```

4.3.5 Finding minimum spanning tree of cluster heads and routing to base station

```
% Distance matrix initialization
distMatrix = inf(cluster-1, cluster-1);

% Calculate distances between cluster heads
for i = 1:cluster-1

    for j = i+1:cluster-1

    distMatrix(i, j) = sqrt((C(i).xd - C(j).xd)^2 + (C(i).yd - C(j).yd)^2);
    distMatrix(j, i) = distMatrix(i, j);

    end

end

% Minimum spanning tree algorithm
visited = zeros(1, cluster-1);
visited(1) = 1;
routeMatrix = zeros(cluster-1, cluster-1);

while sum(visited) < cluster-1

    minDist = inf;
    minIdx = 0;

for i = 1:cluster-1

if visited(i)

 % Find the cluster head with the shortest distance to the visited set
            for j = 1:cluster-1
             if ~visited(j) && distMatrix(i, j) < minDist
                    minDist = distMatrix(i, j);
                    minIdx = j;
            end
        end
     end
  end

% Connect the cluster head to the visited set
routeMatrix(minIdx, find(visited == 1, 1)) = 1;
visited(minIdx) = 1;
end

% Plotting the route
for i = 1:cluster-1

   for j = 1:cluster-1

    if routeMatrix(i, j) == 1
      line([C(i).xd,C(j).xd],[C(i).yd,C(j).yd],'Color','r','LineStyle','--');
    end
```

```
    end

end


STATISTICS.COUNTCHS(h,r+1)=countCHs;

%% Connection of Cluster head and Sink

    cl_list=[0 0 0 0];
    cl_sequence=[0 0 0 0];
    cl_number=0;

    for ii=1:n

        if (S(ii).type == 'C' && S(ii).E>0)
            cl_number=cl_number+1;
            cl_list(cl_number)=ii;
        end

    end

    min_dis_cluster=0;
    min_dis=999999999;
    for c=1:1:cluster-1
          if cl_list(c)==0
            break;
        end
        temp=min(min_dis,sqrt( (S(cl_list(c)).xd-S(n+1).xd)^2
        + (S(cl_list(c)).yd-S(n+1).yd)^2 ));

           if ( temp<min_dis )

               min_dis=temp;
               min_dis_cluster=c;

           end

    end

    cl_sequence(1)=min_dis_cluster;

    if min_dis_cluster==0

        continue
    end

    plot([S(n+1).xd,S(cl_list(min_dis_cluster)).xd,
    [S(n+1).yd,S(cl_list(min_dis_cluster)).yd],'r--');
 end
```

4.3.6 Code for statistics

```
STATISTICS.PACKETS_TO_CH(h,r+1)=packets_TO_CH;
STATISTICS.PACKETS_TO_BS(h,r+1)=packets_TO_BS;
STATISTICS.PACKETS_TO_BS_PER_ROUND(h,r+1)=packets_TO_BS_per_round;
STATISTICS.THROUGHPUT(h,r+1)=STATISTICS.PACKETS_TO_BS(h,r+1)
+STATISTICS.PACKETS_TO_CH(h,r+1);

 En=0;
for i=1:n
    if S(i).E<=0
        continue;
    end
    En=En+S(i).E;
end
ENERGY(r+1)=En;
STATISTICS.ENERGY(h,r+1)=En;


end
first_dead_LEACH(h)=first_dead
half_dead_LEACH(h)=half_dead
all_dead_LEACH(h)=all_dead


% cluster head display-------

end
for r=0:rmax
    STATISTICS.DEAD(h+1,r+1)=sum(STATISTICS.DEAD(:,r+1))/h;
    STATISTICS.ALLIVE(h+1,r+1)=sum(STATISTICS.ALLIVE(:,r+1))/h;
    STATISTICS.PACKETS_TO_CH(h+1,r+1)=sum(STATISTICS.PACKETS_TO_CH(:,r+1))/h;
    STATISTICS.PACKETS_TO_BS(h+1,r+1)=sum(STATISTICS.PACKETS_TO_BS(:,r+1))/h;
    STATISTICS.PACKETS_TO_BS_PER_ROUND(h+1,r+1)=
    sum(STATISTICS.PACKETS_TO_BS_PER_ROUND(:,r+1))/h;
    STATISTICS.THROUGHPUT(h+1,r+1)=sum(STATISTICS.THROUGHPUT(:,r+1))/h;
    STATISTICS.COUNTCHS(h+1,r+1)=sum(STATISTICS.COUNTCHS(:,r+1))/h;
    STATISTICS.ENERGY(h+1,r+1)=sum(STATISTICS.ENERGY(:,r+1))/h;
end

first_dead=sum(first_dead_LEACH)/h;
half_dead=sum(half_dead_LEACH)/h;
all_dead=sum(all_dead_LEACH)/h;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r=0:rmax;
figure(21)
plot(r,STATISTICS.DEAD(h+1,r+1));
title('Dead Nodes')
xlabel('No of Rounds')
ylabel('Number of Dead Nodes')
figure(22)
plot(r,STATISTICS.ALLIVE(h+1,r+1));
title('Live Nodes')
xlabel('No of Rounds')
```

```
ylabel('Number of live Nodes')
figure(23)
plot(r,STATISTICS.PACKETS_TO_BS(h+1,r+1));
title('pkts to BS')
xlabel('No of Rounds')
ylabel('Number of packets')
figure(24)
plot(r,STATISTICS.PACKETS_TO_BS_PER_ROUND(h+1,r+1));
title('pkts to BS per round')
xlabel('No of Rounds')
ylabel('Number of packets')
figure(25)
plot(r,STATISTICS.PACKETS_TO_CH(h+1,r+1));
title('pkts to CH')
xlabel('No of Rounds')
ylabel('Number of packets')
figure(26)
plot(r,STATISTICS.THROUGHPUT(h+1,r+1));
title('THROUGHPUT')
xlabel('No of Rounds')
ylabel('No of bits')
figure(27)
plot(r,STATISTICS.COUNTCHS(h+1,r+1));
title('COUNTCHS')
xlabel('No of Rounds')
ylabel('Number of cluster head ')
figure(28)
plot(r,STATISTICS.ENERGY(h+1,r+1));
title('Average Residual Energy')
xlabel('No of Rounds')
ylabel('Energy in joules')
```

CHAPTER 5

EXPERIMENTAL VALIDATION AND RESULT

5.1 Tools And Technology

MATLAB is used for simulation of our proposed system. MATLAB (short for "MATrix LABoratory") is a powerful and widely-used technical computing language and software environment for numerical computation, visualization, and programming. It is used in a wide range of fields, including engineering, science, finance, and statistics, and is particularly well-suited for developing and testing algorithms, analyzing and visualizing data, and building graphical user interfaces. n addition to its core language and environment, MATLAB includes a wide range of specialized toolboxes and libraries for tasks such as machine learning, signal processing, optimization, and control systems design. These toolboxes provide access to advanced algorithms and functionality, and can be used to build complex and sophisticated models and applications.

There are several tools and technologies that can be used for a wireless sensor network (WSN) in MATLAB, Some of these include:

- MATLAB's built-in functions and toolboxes for wireless communications and networking, such as the WLAN System Toolbox and the Wireless HD Toolbox.
- Hardware tools and platforms, such as hardware-in-the-loop (HIL) simulation systems and sensor nodes, which can be used to test and validate the performance of the WSN.
- Network simulation tools, such as NS-3 and Omnet++, which can be used to model and simulate the behavior of WSNs.
- Protocol stacks and libraries, such as Contiki and TinyOS, which can be used to implement the networking protocols used in WSNs.

- Real-time operating systems (RTOS), such as FreeRTOS and VxWorks, which can be used to support the real-time requirements of WSNs.
- Cloud-based platforms, such as Microsoft Azure and Amazon Web Services (AWS), which can be used to host and manage WSNs at scale.

5.1.1 Simulation Tools

Simulation tools such as NS-2, NS-3, OMNeT++, and MATLAB/Simulink provide a virtual environment for modeling and simulating WSNs. These tools allow researchers and developers to evaluate network performance, test algorithms, and study the behavior of WSNs under various scenarios.Proposed System follows MATLAB.

5.1.2 Programming Languages

Programming languages like C, C++, Java, and Python are commonly used for WSN development. They provide the flexibility to implement network protocols, design sensing applications, and develop control logic for sensor nodes.

MATLAB is a high-level programming language that is widely used in scientific and engineering fields. It provides a flexible and powerful environment for numerical computation, data analysis, and visualization. MATLAB offers a wide range of built-in functions and libraries that facilitate mathematical operations, matrix manipulations, statistical analysis, and signal processing. Additionally, MATLAB supports various programming paradigms, including procedural programming, object-oriented programming, and functional programming. It also provides interactive development tools, such as the MATLAB Editor and the Command Window, which allow users to write and execute code in an interactive manner. MATLAB's versatility and extensive toolboxes make it a popular choice for a wide range of applications, including data analysis, image and signal processing, control systems, machine learning, and simulation.

5.1.3 Operating Systems

Operating systems specifically designed for WSNs, such as TinyOS, Contiki, and RIOT, provide a platform for efficient resource management and programming of sensor nodes. These operating systems offer low-power capabilities, real-time scheduling, and support for sensor-specific protocols and applications.MATLAB is a programming language and software environment that is designed to work on multiple operating systems. It is compatible with various operating systems, including:

- Windows: MATLAB is fully supported on the Windows operating system, including the latest versions such as Windows 10.
- macOS: MATLAB is compatible with macOS, allowing users to develop and run MATLAB code on Apple computers.
- Linux: MATLAB provides support for popular Linux distributions, including Ubuntu, Red Hat, and CentOS, among others. It offers both command-line and graphical user interfaces on Linux.

5.1.4 Sensor Platforms

Sensor platforms, such as TelosB, MICAz, and Arduino, provide hardware platforms with integrated sensing, processing, and communication capabilities. These platforms enable rapid prototyping and deployment of sensor nodes in WSNs.MATLAB provides various sensor platforms and toolboxes that allow users to interface with and work with different types of sensors. Here are some commonly used sensor platforms in MATLAB:

- Arduino: MATLAB supports Arduino hardware, which enables users to interface with a wide range of sensors and actuators. The MATLAB Support Package for Arduino provides functions and libraries to communicate with Arduino boards, read sensor data, and control actuators.
- Raspberry Pi: MATLAB also supports Raspberry Pi, a popular single-board computer. The MATLAB Support Package for Raspberry Pi allows users to access the GPIO pins on the Raspberry Pi for sensor interfacing, as well as control external devices.
- Mobile Devices: MATLAB can interface with mobile devices such as smartphones and tablets. The MATLAB Mobile app enables users to acquire data from the device's built-in sensors, including accelerometers, gyroscopes, GPS, and more.
- Kinect: MATLAB supports the Microsoft Kinect sensor, which provides depth sensing and skeletal tracking capabilities. The Kinect for Windows Sensor Support package enables users to capture depth and color data, as well as track human body movements.

- Lidar: MATLAB has toolboxes that support working with Lidar (Light Detection and Ranging) sensors. These toolboxes provide functions for processing and analyzing Lidar data, including point cloud processing and object detection.
- Image Sensors: MATLAB offers a comprehensive set of functions for working with image sensors, including image acquisition, processing, and analysis. It supports various image sensor platforms, including webcams, frame grabbers, and machine vision cameras.

5.1.5 Communication Protocols

Communication protocols like Zigbee, Bluetooth Low Energy (BLE), and 6LoWPAN provide efficient and reliable data transmission in WSNs. These protocols are designed to operate with low-power and low-data-rate requirements of sensor networks.MATLAB provides support for various communication protocols that enable data transmission and communication between devices. Some commonly used communication protocols in MATLAB are:

- Serial Communication (UART): MATLAB has built-in functions to establish serial communication with devices using the UART (Universal Asynchronous Receiver/-Transmitter) protocol. The Serial Communication Toolbox in MATLAB provides functions for configuring serial ports, sending and receiving data, and controlling communication parameters.
- TCP/IP (Transmission Control Protocol/Internet Protocol): MATLAB supports TCP/IP communication, which is a widely used protocol for data transmission over networks. The TCP/IP Communication Toolbox in MATLAB allows users to create TCP/IP client and server applications, establish connections, send and receive data packets, and handle network communication.
- Bluetooth: MATLAB provides support for Bluetooth communication through the Instrument Control Toolbox. This enables users to connect and communicate with Bluetooth-enabled devices, such as sensors, actuators, and mobile devices. The toolbox provides functions for discovering nearby Bluetooth devices, establishing connections, and exchanging data.
- UDP (User Datagram Protocol): MATLAB supports UDP communication, which is a lightweight and connectionless protocol suitable for applications that require fast and low-latency data transmission. The UDP Communication Toolbox in MATLAB enables users to create UDP client and server applications, send and receive UDP packets, and handle network communication.

- MQTT (Message Queuing Telemetry Transport): MATLAB offers support for MQTT, a lightweight publish-subscribe messaging protocol commonly used in IoT applications. The MATLAB MQTT Toolbox allows users to connect to MQTT brokers, publish and subscribe to MQTT topics, and exchange data with MQTT-enabled devices.
- CAN (Controller Area Network): MATLAB provides support for CAN communication, which is widely used in automotive and industrial applications. The Vehicle Network Toolbox in MATLAB allows users to communicate with CAN devices, send and receive CAN messages, and analyze CAN bus data.

5.1.6 Data Management Systems

Data management systems like TinyDB, Cougar, and Berkeley DB enable efficient storage, retrieval, and querying of sensor data in WSNs. These systems provide mechanisms for aggregating, processing, and analyzing data collected from sensor nodes.

5.1.7 Energy Harvesting Technologies

Energy harvesting technologies, such as solar panels, piezoelectric systems, and kinetic energy harvesters, enable the generation of electrical power from ambient energy sources. These technologies help prolong the lifetime of WSNs by reducing the dependence on battery-powered nodes.

5.1.8 Wireless Network Testbeds

Wireless network testbeds, such as TOSSIM, FIT IoT-LAB, and FlockLab, provide real-world experimental platforms for testing and validating WSN deployments. These testbeds allow researchers to evaluate network performance, assess scalability, and study the behavior of WSNs in diverse environments.

These tools and technologies collectively support the development, deployment, and management of wireless sensor networks, enabling researchers and practitioners to design and optimize efficient and reliable WSN solutions.

5.2 Random Distribution of sensor nodes

Initially, Random Distribution of nodes to predetermined area of Wireless Sensor Nodes also determining the number of nodes and predetermined location of sink(Base Station) is located.

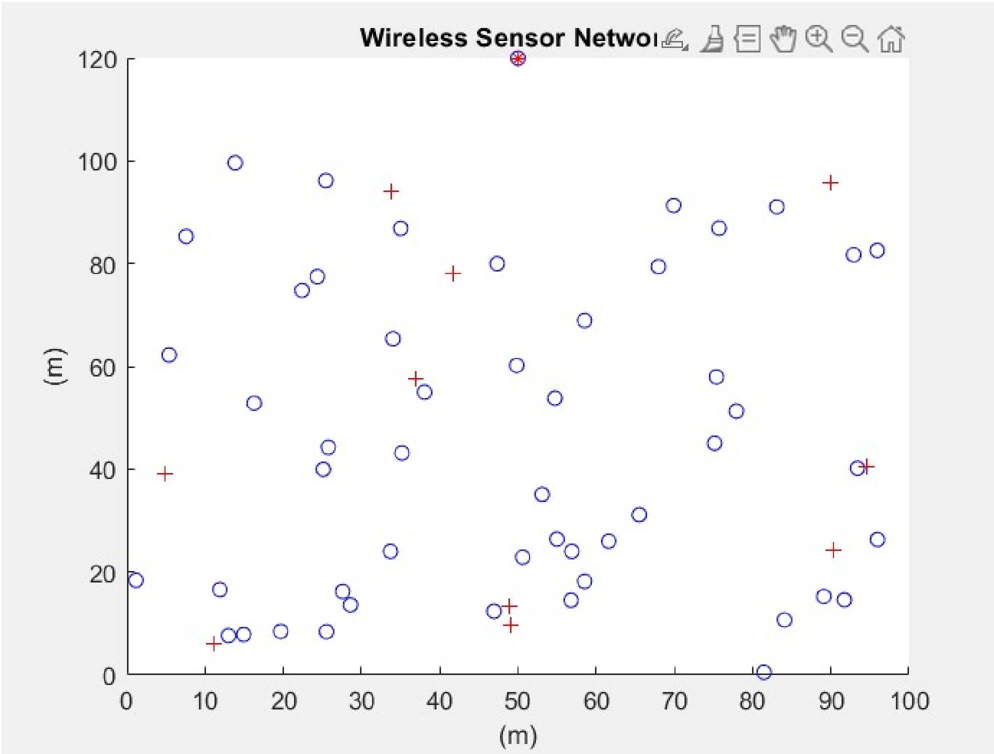

FIGURE 5.1: Random distribution of nodes with single base station including movable nodes (represented as +).

Random distribution of sensor nodes refers to the process of placing sensor nodes in a network area in a random or unpredictable manner. In this approach, the sensor nodes are deployed without any specific pattern or predefined arrangement. The nodes are scattered randomly across the network area, ensuring a more even and unbiased coverage.

5.3 Delaunay Triangulation

By creation of deployment of sensor nodes randomly, next is for delaunay triangulation of the network with base station and Static Senor Nodes in a particular location in (x, y) coordinates.

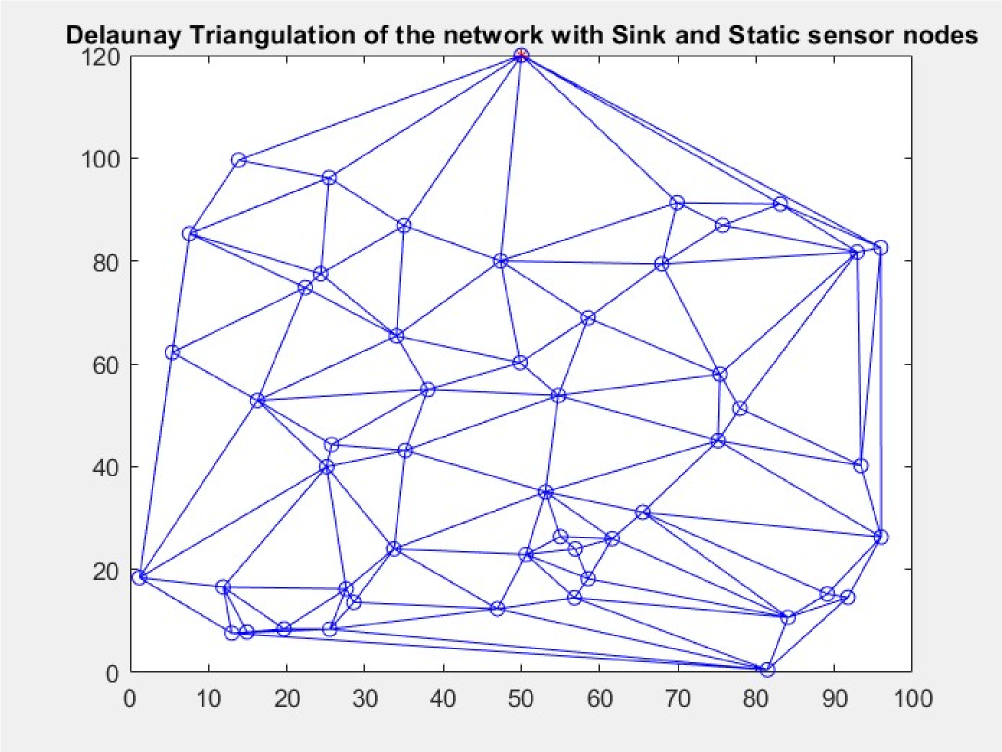

FIGURE 5.2: Delauny Triangulation of the network with Sink and Static sensor nodes

Delaunay triangulation is a geometric algorithm used to partition a set of points in a plane into a network of non-overlapping triangles.The Delaunay triangulation ensures that no point in the set is inside the circumcircle of any triangle formed by the points. In other words, the triangles in the triangulation are defined in such a way that the circumcircle of each triangle contains only its three vertices and no other points. This property makes Delaunay triangulation useful in various fields such as computer graphics, mesh generation, terrain analysis, and wireless sensor networks.

5.4 Current location of mobile sensors

Movable senor nodes are present at random places and showing current positions of that nodes. And then needed to assign these nodes to the longest edge of formed triangle in triangulation.

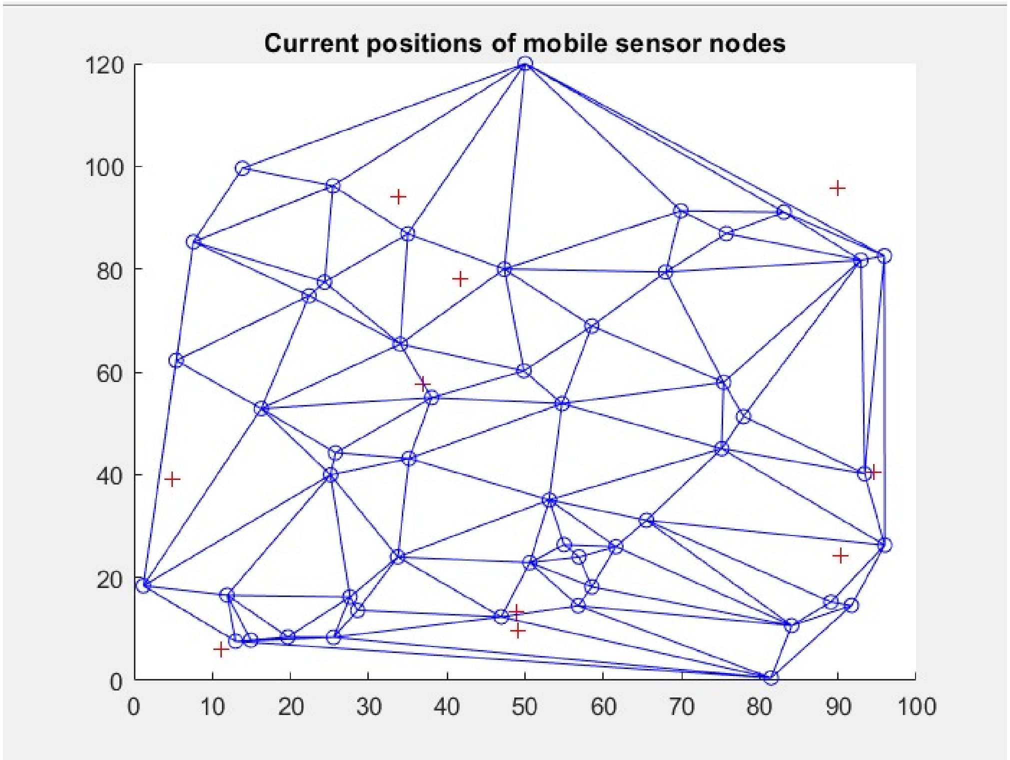

FIGURE 5.3: Current positions of mobile sensor nodes (represented as +).

5.5 Placement of Movable Sensor Nodes

By Identifying longest edge, these movable nodes are placed at circum center. And having the longest edge of the triangle able to place these movable nodes in to mitigate issue of coverage hole.

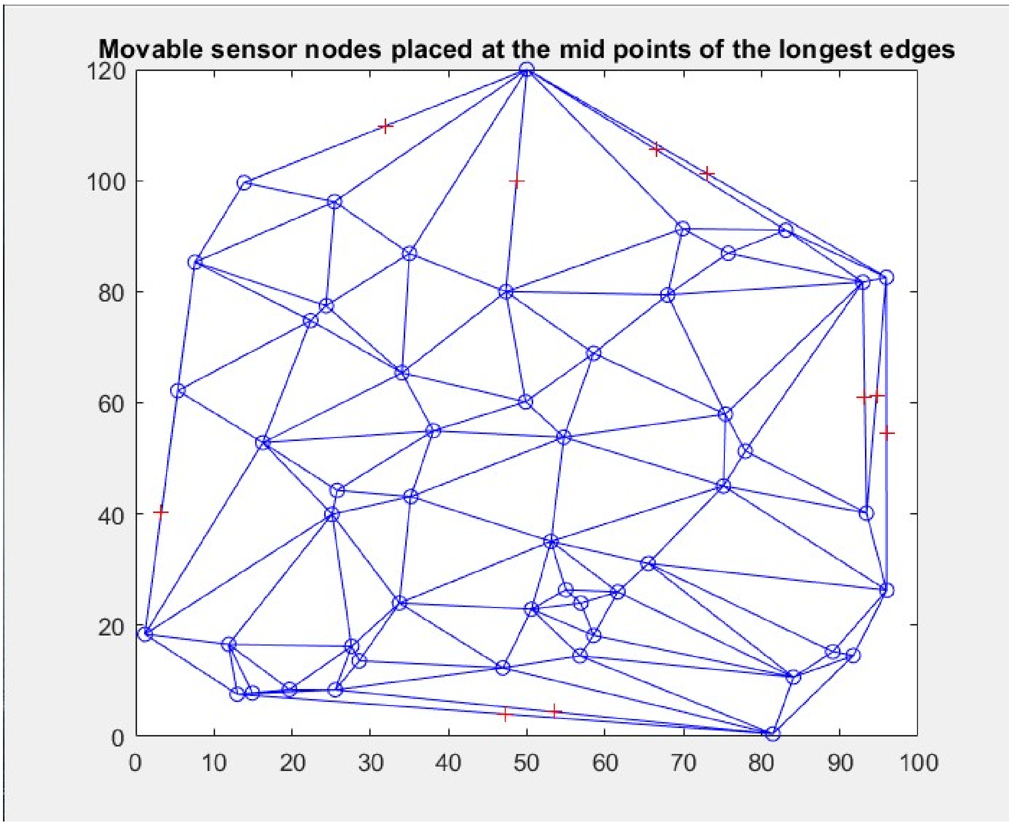

Figure 5.4: Movable Sensor Nodes placed at the mid points of the longest edges.

5.6 Retriangulation with movable nodes

After placement of movable nodes in the hole area to mitigate coverage hole. Again retriagulate including recently placed movable nodes for triangulating the whole nodes covering the whole network area.

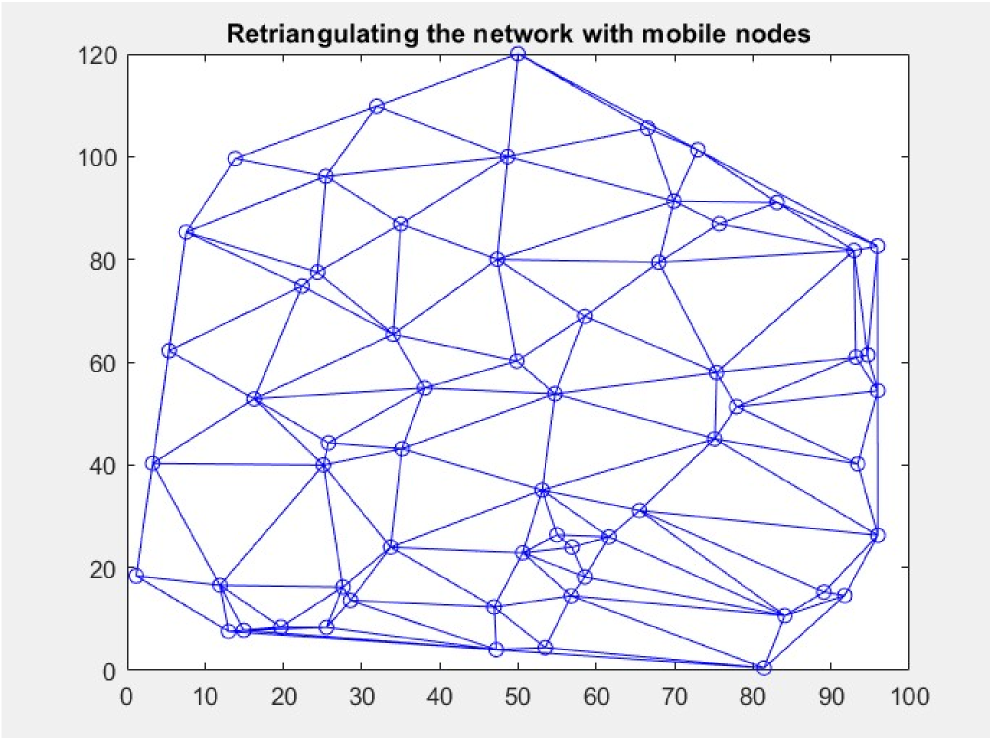

FIGURE 5.5: Retriangulating the network with mobile nodes.

Retriangulation with movable nodes refers to the process of dynamically adjusting the connectivity and arrangement of triangles in a triangulation by incorporating movable nodes. In this context, the term ”movable nodes” refers to additional nodes that can be deployed or relocated within the existing triangulation to improve certain aspects such as coverage, connectivity, or network performance.

5.7 Location of deployed nodes

Location of deployed nodes after the retriangulation of with presence of mobile nodes to the base station and its node location is shown in the figure.

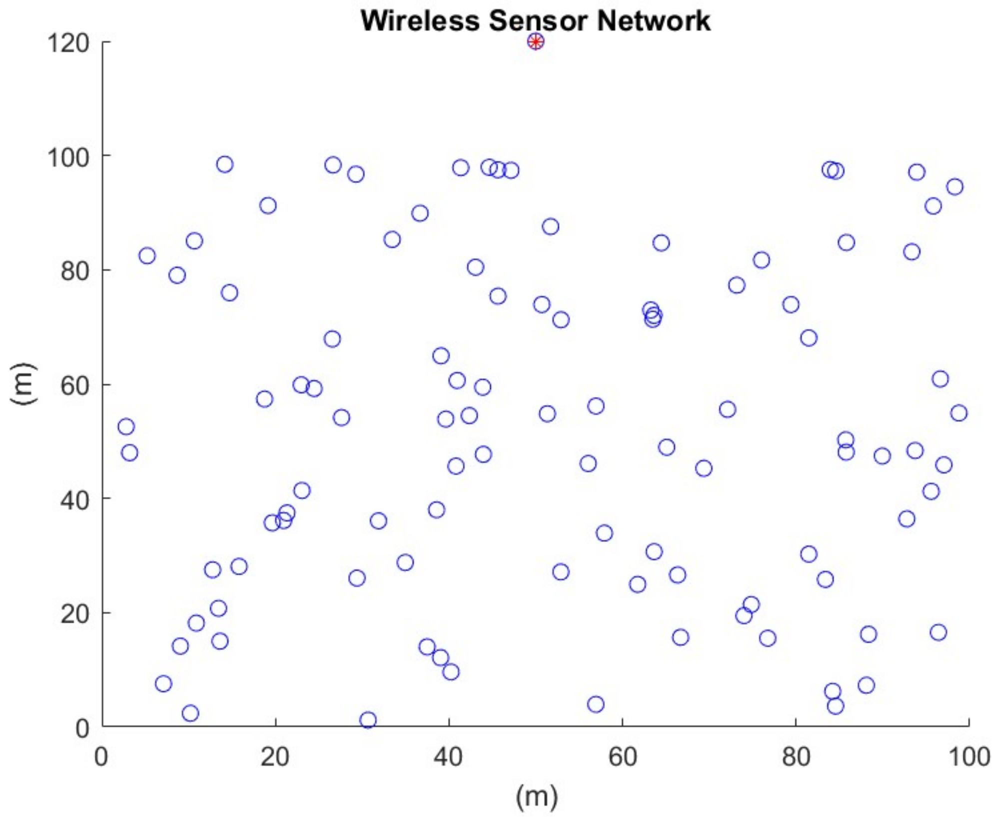

FIGURE 5.6: deployed node location after retriangulation of mobile nodes

5.8 Equi-Quadrant Division Clustering (EQDC) Protocol

After mitigating the coverage hole issue by placement of mobile nodes to the hole area to the longest edge and then retriangulated and the node location of each node is identified and the EQDC Protocol is applied as:

- Network Initialization: The network area is divided into equal quadrants, and sensor nodes are randomly distributed within each quadrant.
- Cluster Head Selection: Each quadrant selects a cluster head based on certain criteria, such as the node with the highest remaining energy or the node with the strongest signal strength. The cluster head is responsible for managing the communication within the cluster.
- Cluster Formation: Sensor nodes in each quadrant join the cluster headed by the selected cluster head closest to them.
- Data Aggregation: Sensor nodes within each cluster collect data from their surroundings and send it to the cluster head for aggregation.
- C1uster Head Communication: Cluster heads communicate with each other to exchange aggregated data and make decisions collectively.
- Data Transmission to Base Station: The cluster heads forward the aggregated data to the base station using multi-hop communication.
- Energy Management: Each sensor node monitors its energy level, and if it drops below a certain threshold, it may request to become a cluster head in the next round to evenly distribute the energy consumption.

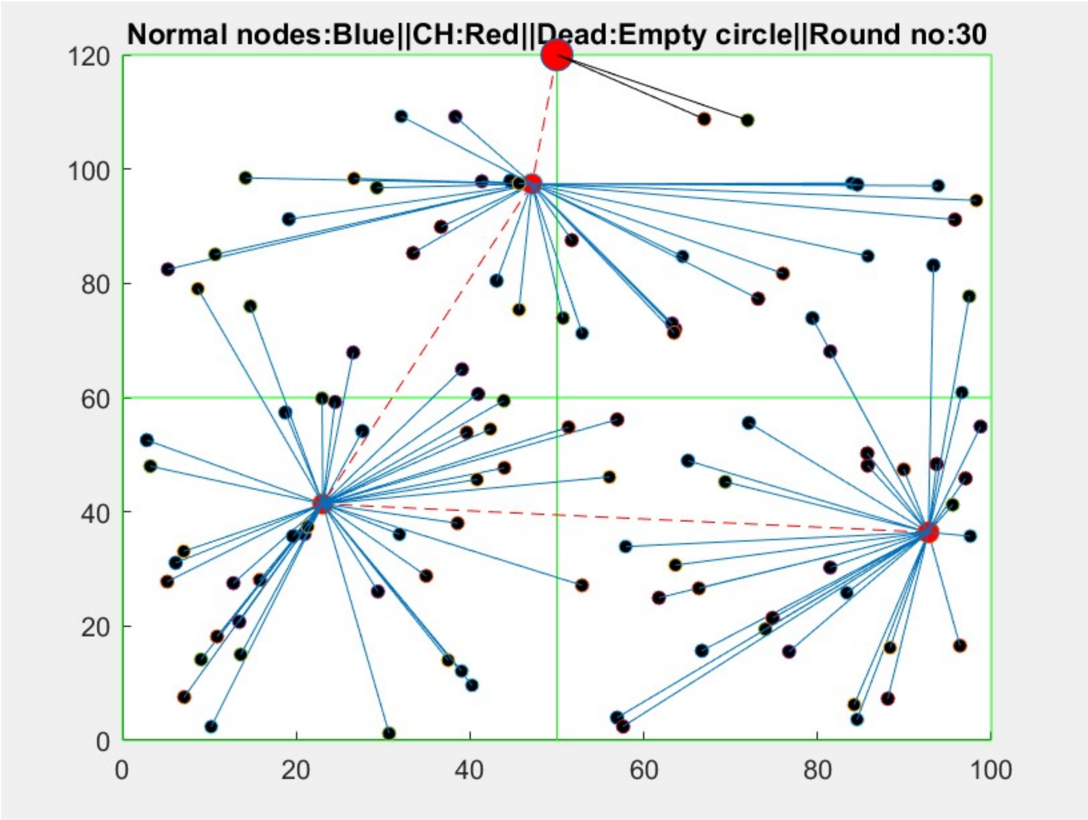

FIGURE 5.7: EQDC Protocol with routing range of 0-100 rounds

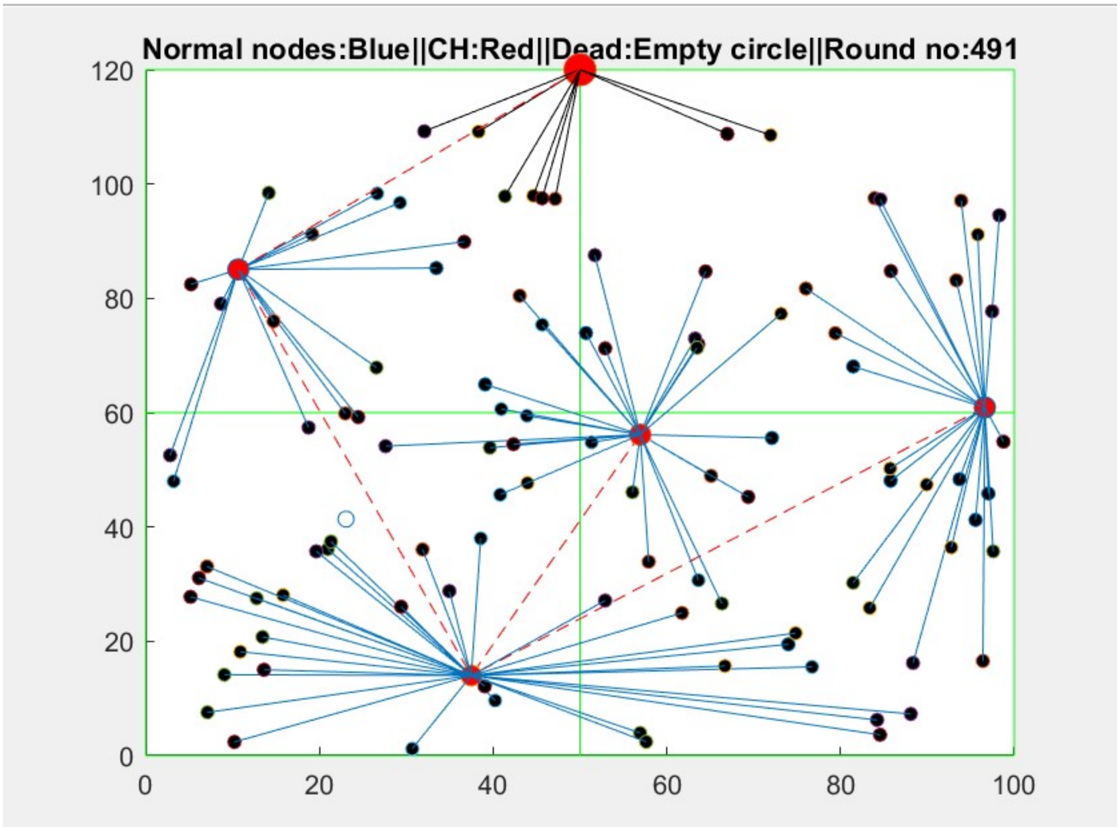

FIGURE 5.8: Able observe beginning of deadnodes(empty circle)

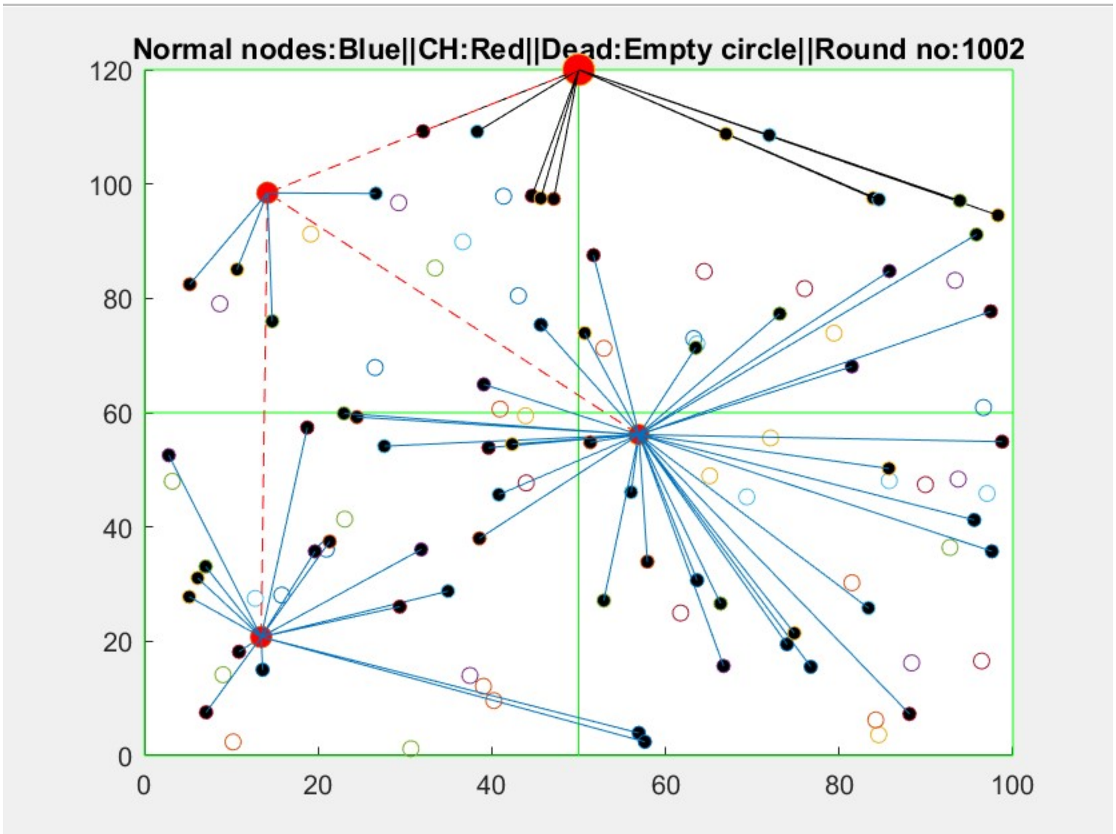

FIGURE 5.9: At 1000 rounds of routing

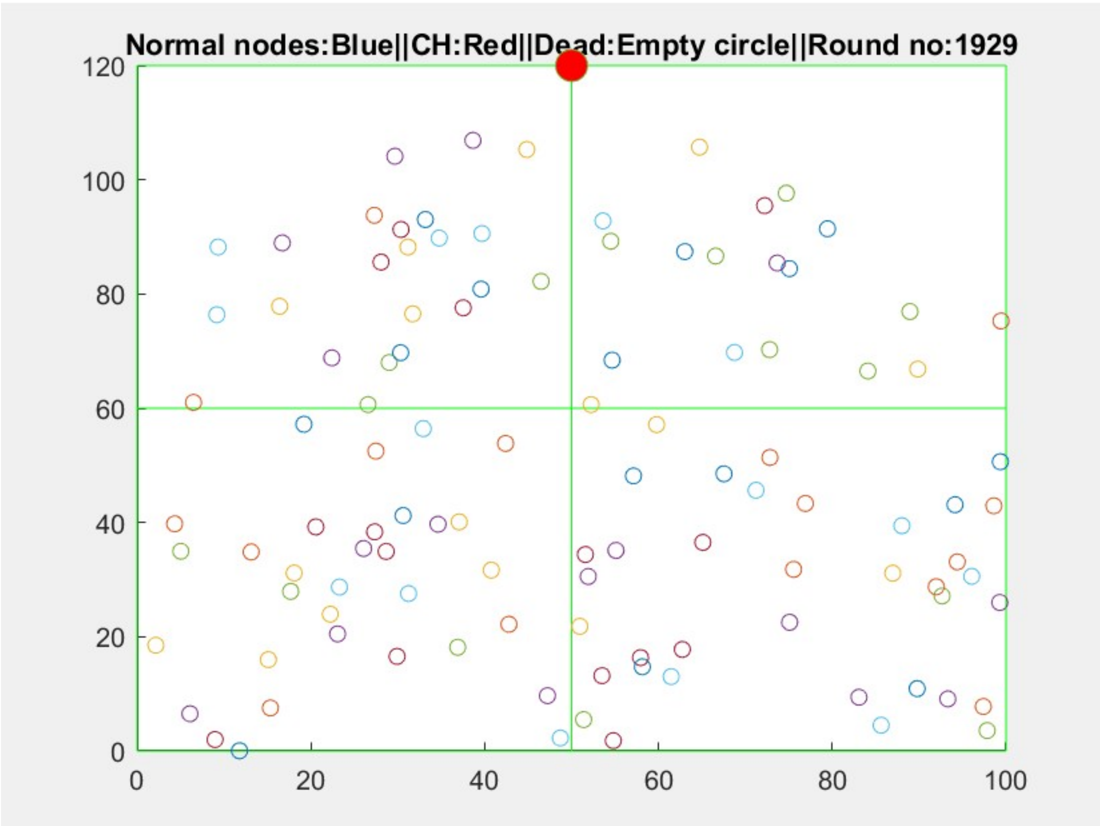

FIGURE 5.10: All nodes becomes dead after a particular round of routing

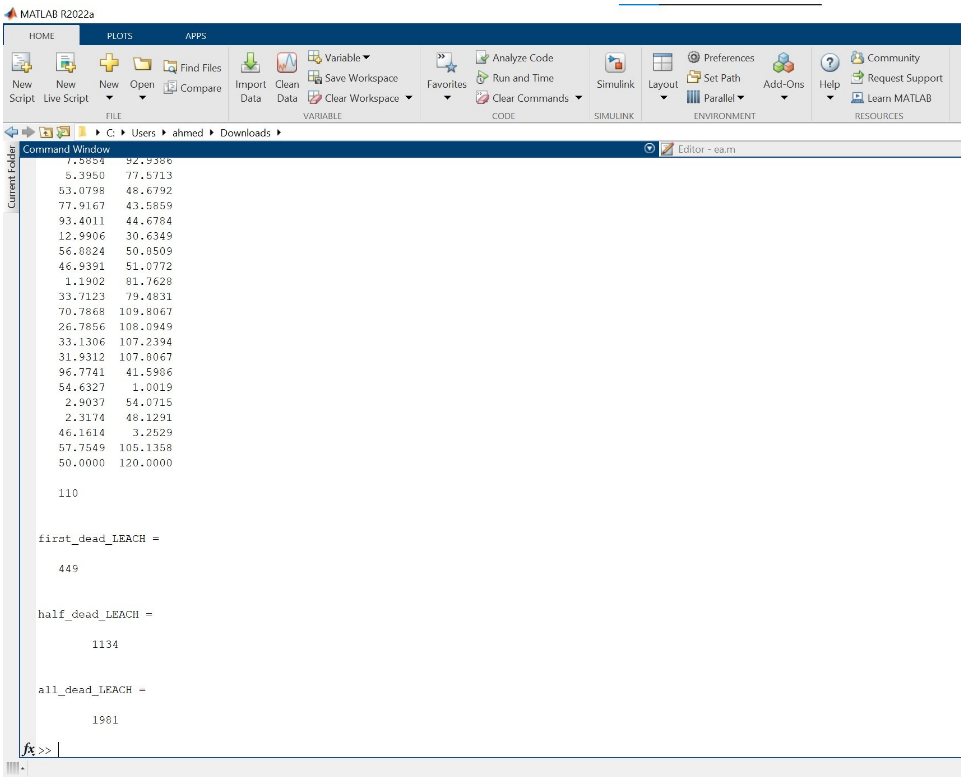

FIGURE 5.11: State of nodes in routing is shown as first,half and all deadnode in routing

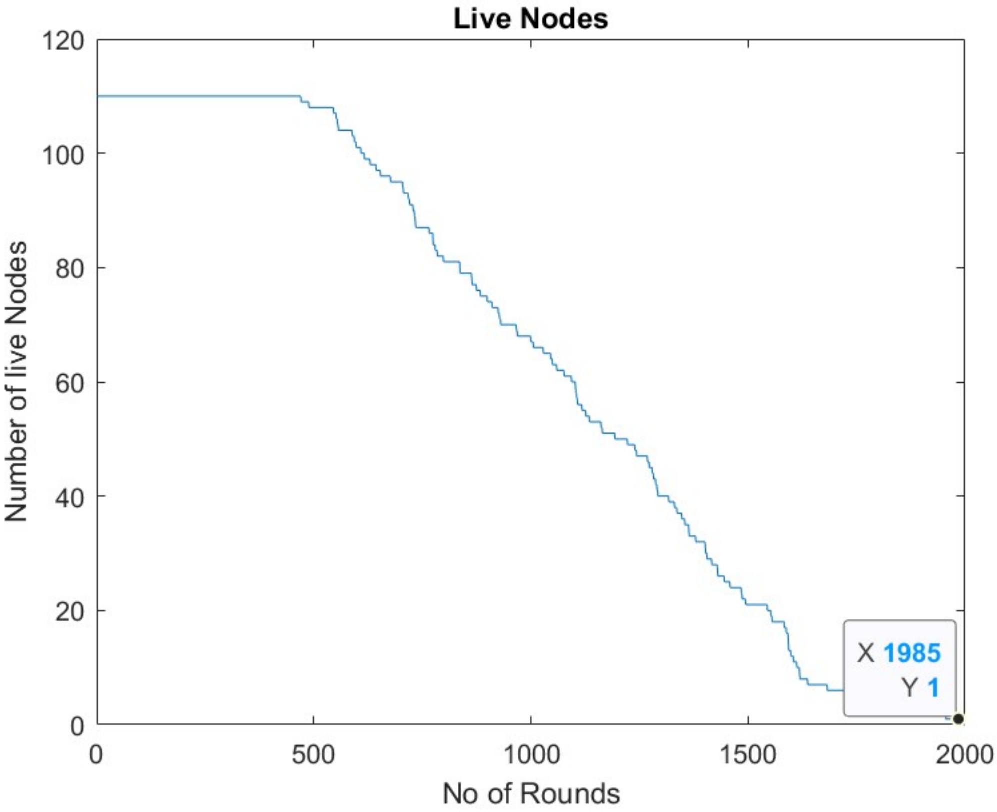

FIGURE 5.12: Plot of Live nodes

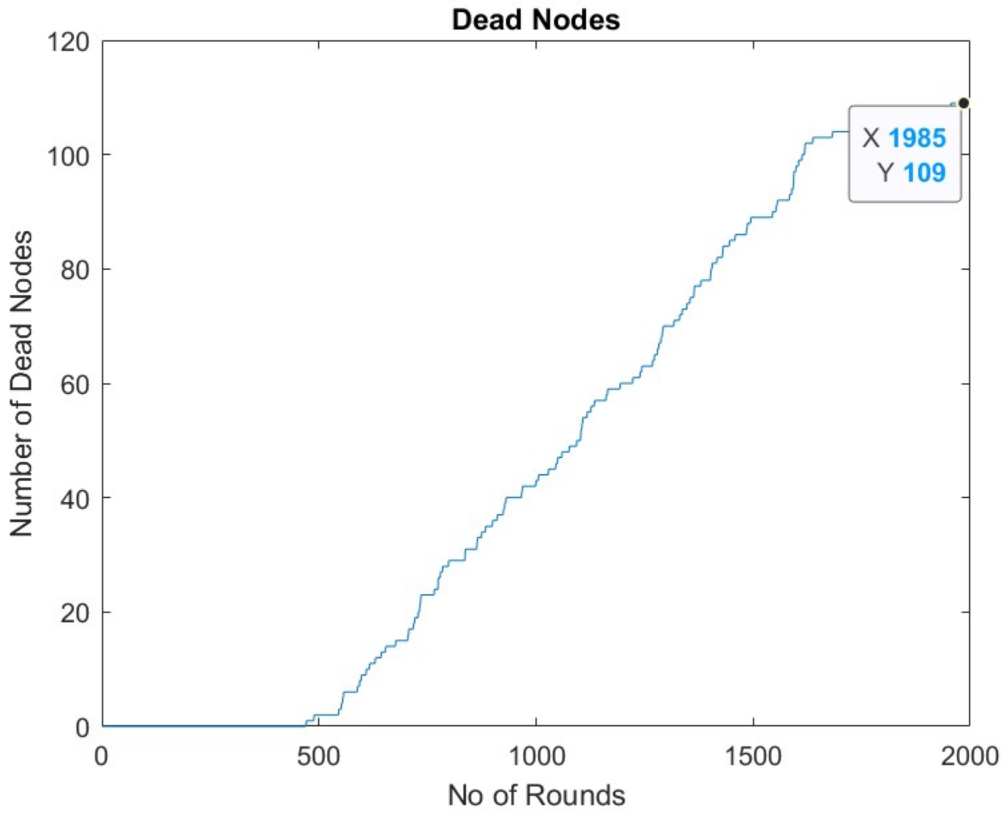

FIGURE 5.13: Plot of Dead nodes

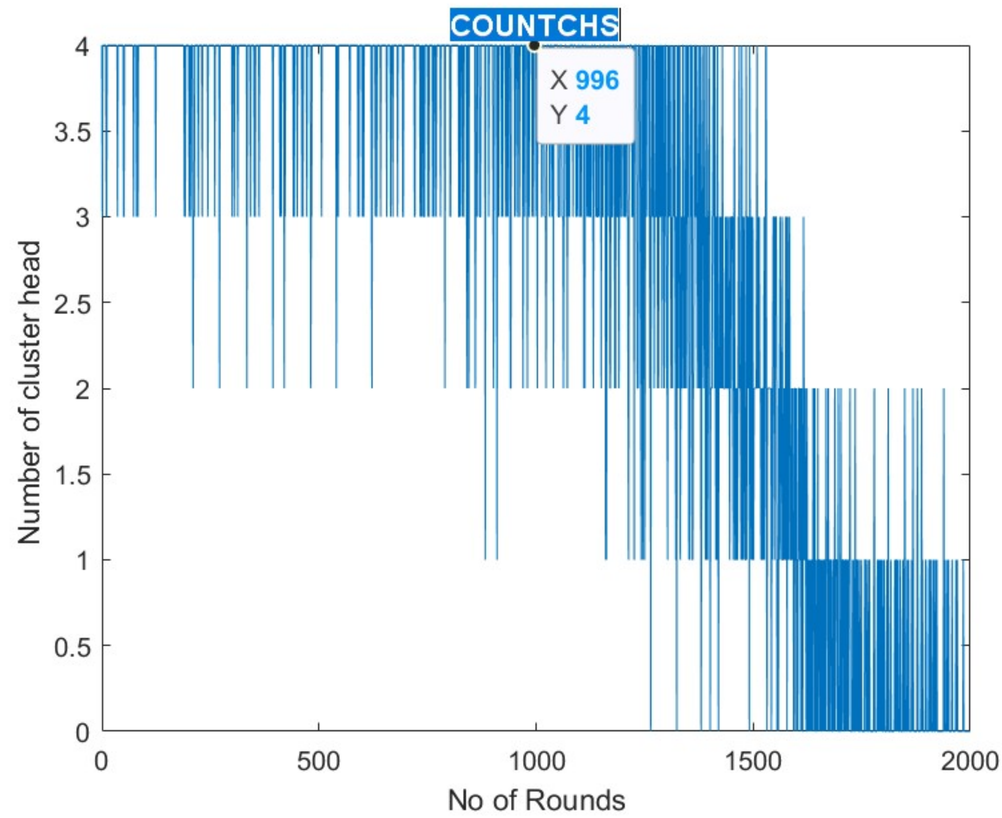

FIGURE 5.14: Plot Of count of cluster head

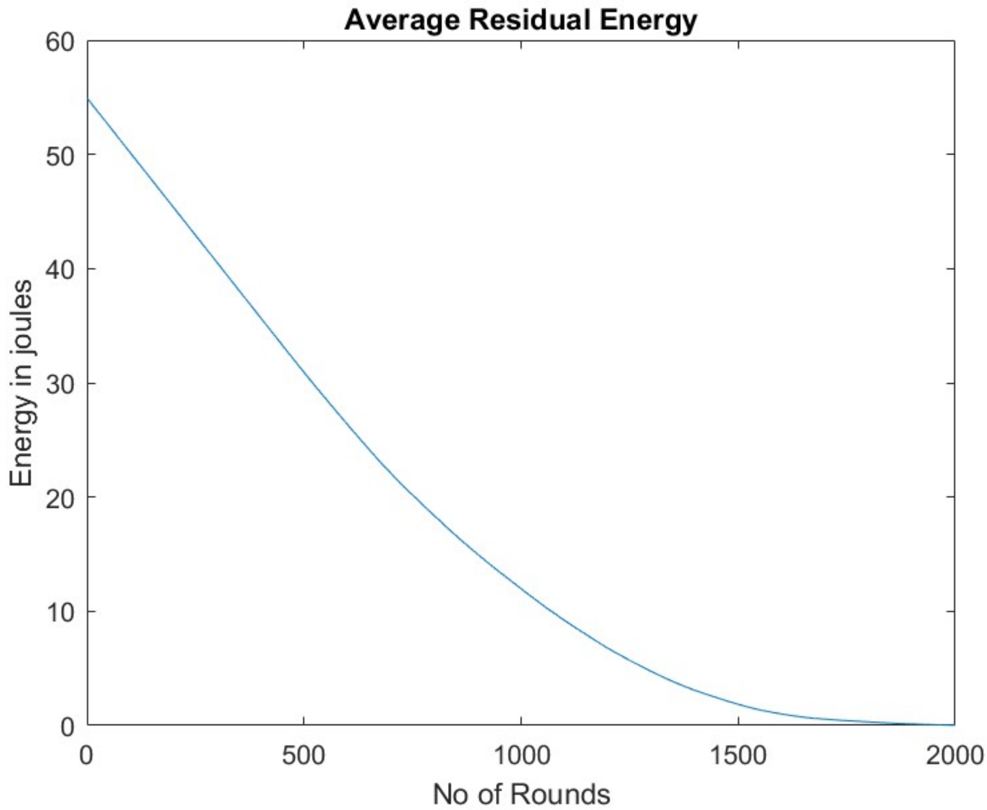

FIGURE 5.15: Plot of average residual energy

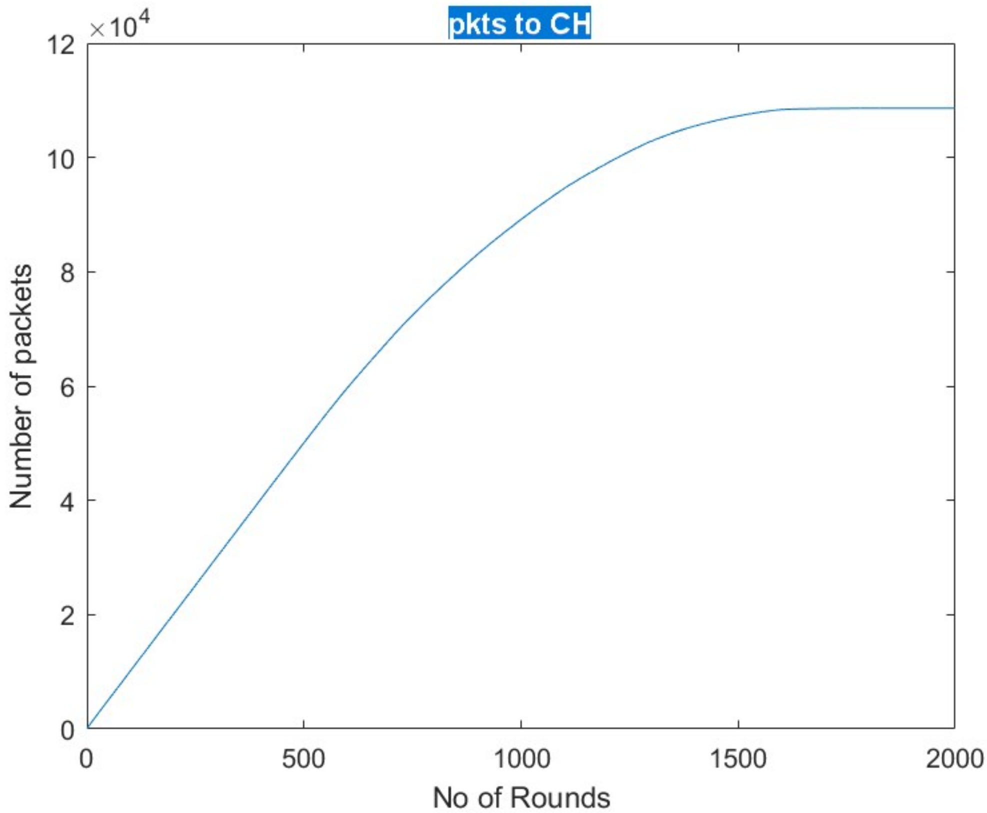

FIGURE 5.16: PLot of Packets to cluster head

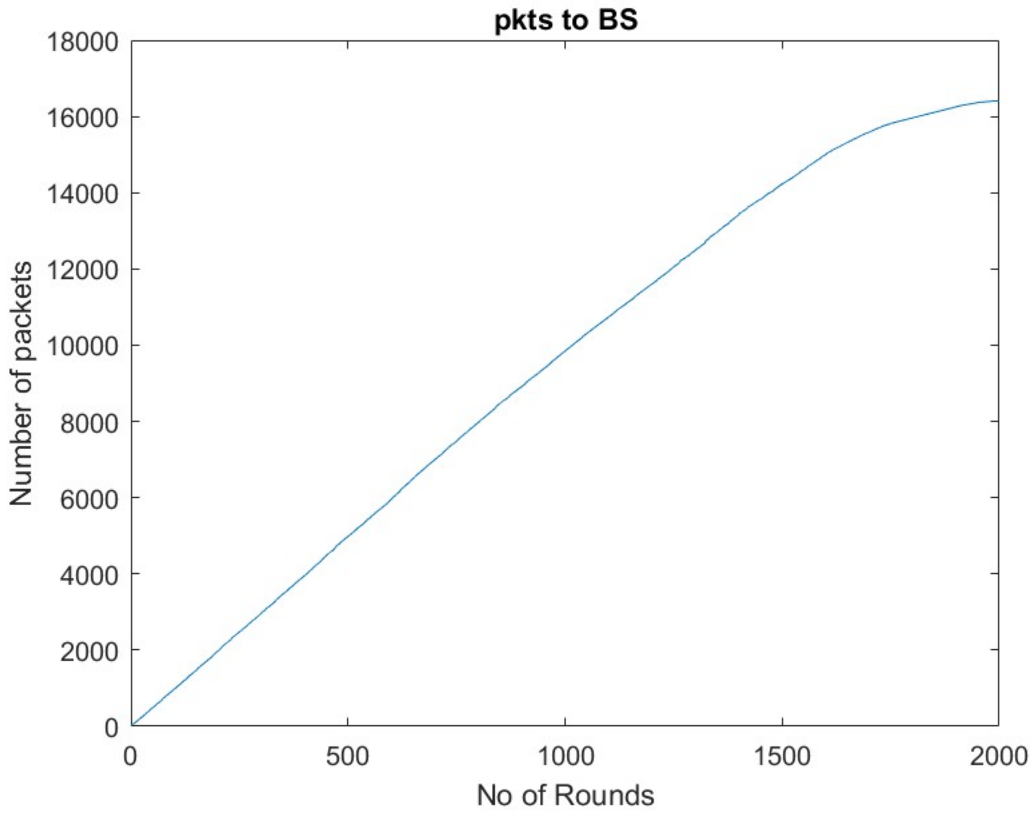

FIGURE 5.17: Plot of packets to Base station

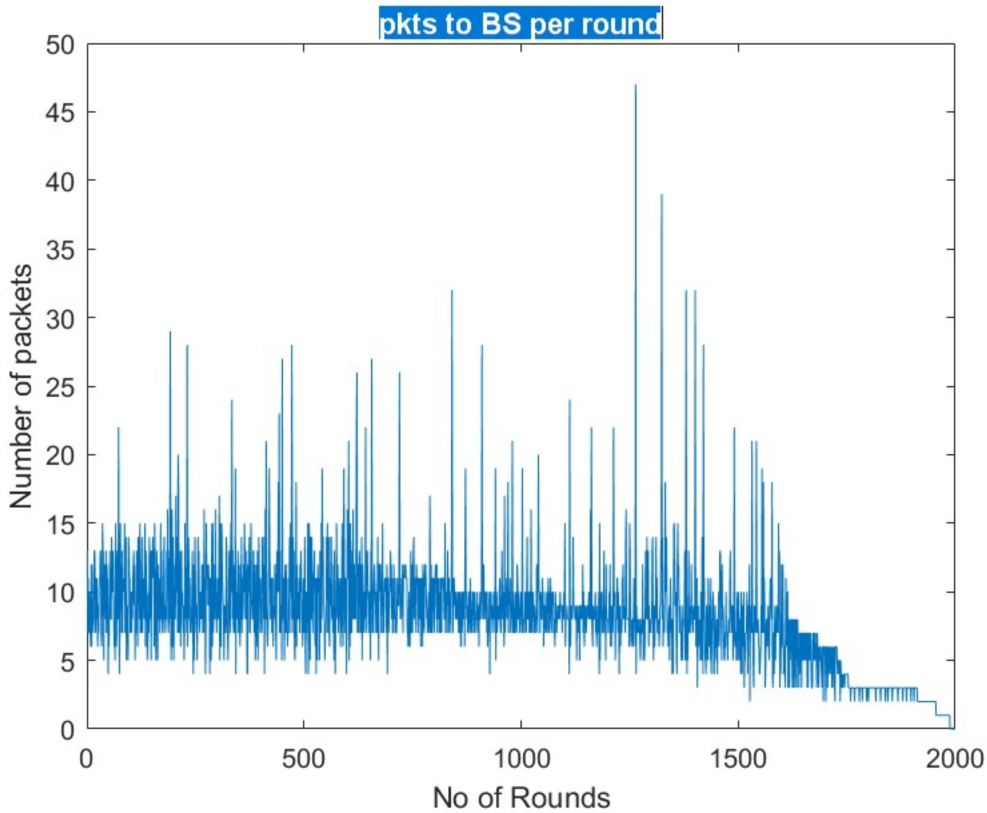

FIGURE 5.18: Plot of Packets Base station Per Round

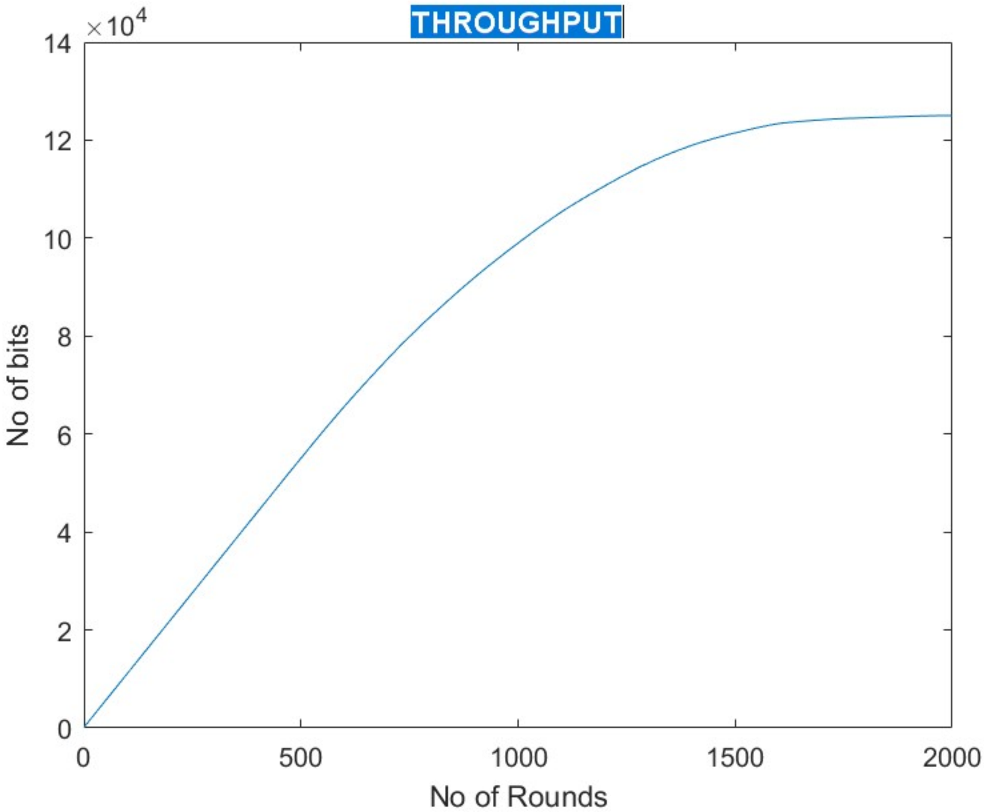

FIGURE 5.19: Plot showing Throughput

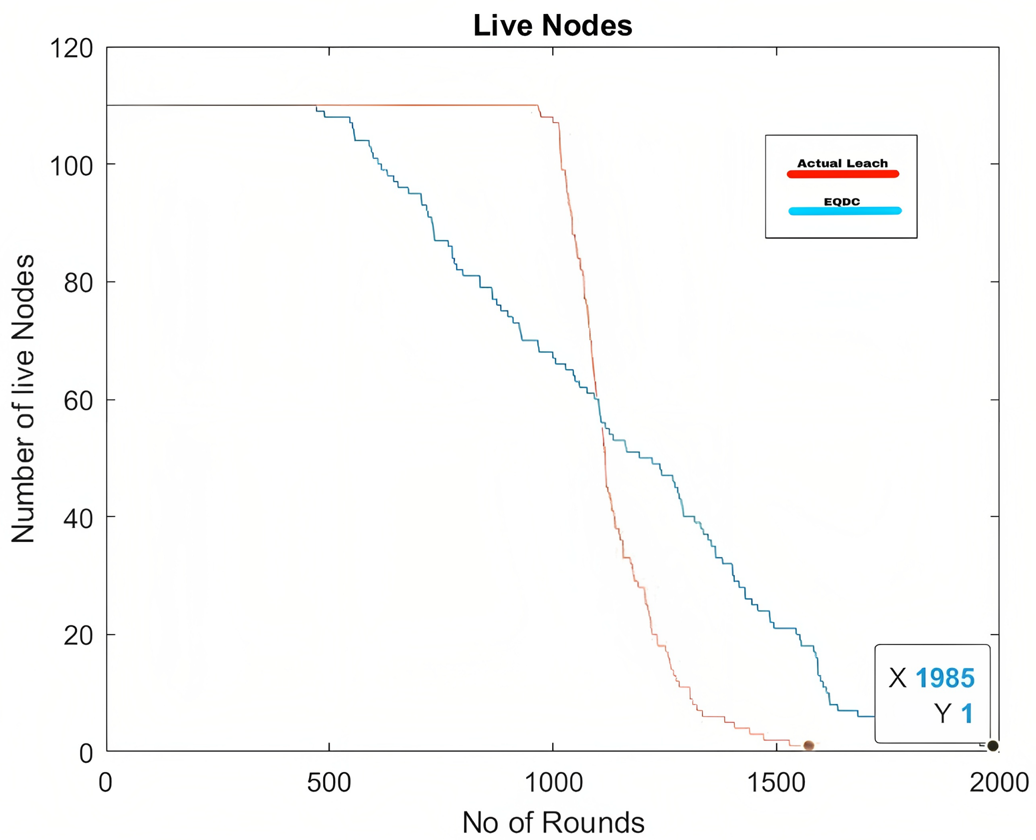

FIGURE 5.20: Comparison with normal LEACH and Our Proposed Method

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In conclusion, the random distribution of sensor nodes in a wireless sensor network (WSN) is often necessary in remote or hostile environments where deterministic deployment is risky or infeasible. Random placement strategies can be categorized as simple or compound, and empirical studies have been conducted to analyze their intrinsic properties such as coverage, connectivity, fault-tolerance, and network lifespan.

Delaunay triangulation, a computational geometry technique, is used to indicate the network's topology in a WSN. It helps identify coverage holes within the network by examining the empty circles formed by the Delaunay triangles. Clustering algorithms can then be applied to group nodes that enclose the same coverage hole, allowing for targeted deployment of movable nodes to mitigate the coverage hole issue.

To detect coverage holes, the coverage areas of active nodes in the WSN are defined, and areas not covered by any active node are identified as potential coverage holes. Additional information such as signal strength or quality can be used to confirm and characterize these coverage holes.

Mitigating coverage holes in a WSN offers several benefits, including improved coverage and connectivity, enhanced system performance, increased energy efficiency, and enhanced security. By patching coverage holes with movable nodes placed strategically within the holes, the overall coverage and performance of the network can be significantly improved.

The proposed clustering approach in the WSN involves dividing the network into quadrants with designated cluster heads. This localized routing enables efficient data aggregation, reduced transmissions, and optimized network performance. The hierarchical structure ensures effective communication and facilitates the collection and transmission of aggregated data to the base station for further analysis.

In the proposed routing method, a minimum spanning tree is used to establish an optimal path between the nearest cluster heads. This approach minimizes energy consumption, reduces communication delays, and dynamically adjusts the routing path based on network changes, enhancing network efficiency and reliability.And the nearest cluster head is connected to base station.

Overall, these approaches and techniques contribute to improving the efficiency, coverage, and performance of wireless sensor networks, making them more effective for monitoring and surveillance applications in various environments.

6.2 Future Scope

Our proposed system for wireless sensor networks (WSNs) opens up several avenues for future development and research. Here are some potential future scopes for the proposed system: **Advanced Deployment Strategies:** While random placement of nodes is often the only option in remote or hostile environments, future research can focus on developing advanced deployment strategies. These strategies could involve intelligent algorithms or machine learning techniques to optimize node placement based on factors such as coverage, connectivity, energy efficiency, and network lifespan. By leveraging advanced deployment strategies, the performance of WSNs can be further improved. **Dynamic Node Mobility:** The proposed system includes the use of movable nodes to patch coverage holes. Future research can explore dynamic node mobility, where nodes have the ability to autonomously move within the network based on changing conditions. This could involve developing algorithms that enable nodes to self-organize and reposition themselves to maximize coverage, adapt to environmental changes, and optimize network performance in real-time. **Energy Harvesting and Power Management:** Energy efficiency is a critical aspect of WSNs. Future advancements can focus on integrating energy harvesting technologies, such as solar or kinetic energy harvesting, to power the sensor nodes. Additionally, research can be directed towards developing intelligent power management techniques that optimize the energy consumption of nodes, extend their battery life, and minimize the need for manual maintenance or replacement. **Fault-Tolerant and Resilient Networks:** WSNs operate in dynamic and challenging environments, making them susceptible to node failures, communication disruptions, or malicious attacks. Future research can explore fault-tolerant and resilient mechanisms to ensure the robustness of the network. This could involve developing self-healing algorithms, redundancy strategies, or adaptive routing protocols that can detect and recover from failures, maintain connectivity, and enhance the overall reliability of the WSN. **Integration with Emerging Technologies:** The proposed system can be further enhanced by integrating it with emerging technologies. For example, the fusion of WSNs with edge computing, cloud computing, or artificial intelligence can enable advanced data processing, analytics, and decision-making capabilities within the network. This integration can lead to more intelligent and efficient utilization of network resources, improved data analysis, and enhanced real-time responsiveness. **Security and Privacy Considerations:** As WSNs are deployed in sensitive applications and collect valuable data, future research should focus on addressing security and privacy challenges. This can involve developing robust encryption algorithms, intrusion detection systems, privacy-preserving techniques, and secure communication protocols to protect the integrity, confidentiality, and privacy of data transmitted within the network.

CHAPTER 7

PUBLICATION

7.1 Paper Title

AN ENERGY EFFICIENT WSN WITH EQDC PROTOCOL AND IMPROVED NODE PARTICIPATION

7.2 Authors

PROF. SREERAM S , ABOOBACKER SIDHEEQUE K , ABOOD MOHAMMED V , AHMED KAMIL , AJINAS P

7.3 Publication

We are eagerly anticipating the publication of our research project paper in an esteemed publication. With meticulous attention to detail, rigorous analysis, and adherence to the highest standards of scientific inquiry, we have strived to create a manuscript that aligns with the expectations of research publication. Our research project not only addresses a significant problem or research question but also presents novel methodologies, insightful findings, and practical implications for the field. We have conducted extensive experiments, employed robust data analysis techniques, and critically evaluated the results to ensure the validity and reliability of our findings. By aiming for publication in journal or conference, we aspire to contribute to the broader body of knowledge, inspire further research, and make a meaningful impact in the field of engineering and technology. We eagerly look forward to the opportunity to share our research with the global scientific community through publication, fostering collaboration, and stimulating advancements in our area of expertise.

AN ENERGY EFFICIENT WIRELESS SENSOR NETWORK WITH EQDC PROTOCOL AND IMPROVED NODE PARTICIPATION

Prof. Sreeram S
Computer Science And Engineering
MEA Engineering College
kerala, India
Email: sreeram@meaec.edu.in

Aboobacker Sidheeque k
Computer Science And Engineering
MEA Engineering College
kerala, India
Email:19gcs05@meaec.edu.in

Abood Mohammed V
Computer Science And Engineering
MEA Engineering College
kerala, India
Email:19gcs18@meaec.edu.in

Ahmed kamil
Computer Science And Engineering
MEA Engineering College
kerala, India
Email:19gcs20@meaec.edu.in

Ajinas P
Computer Science And Engineering
MEA Engineering College
kerala, India
Email:19mcs11@meaec.edu.in

***Abstract*—In this paper, we have proposed an Energy efficient wireless sensor network with improved node participation by placement of movable nodes in the network area, by proposed energy-efficient Equi-Quadrant Division Clustering(EQDC) protocol and optimized shortest path routing to the base station. On random distribution of nodes into a predefined network area consisting of major issues like coverage hole, sensing range of each node . These are solved by improved node participation in the proposed system. Accordingly, the network area divided into equal four square quadrant of having a single cluster head in each quadrant. Also minimum spanning tree is used for obtaining minimum path from each cluster head to base station. The purpose of improved node participation, EQDC and optimized shortest path routing is to increase data reliability, mitigate coverage hole issue, reduce energy consumption for wireless sensor network (WSN) and increase network efficiency. The experimental results of proposed methodology shows that EQDC protocol, enhanced node engagement and OSPR protocol shows the tackle coverage hole challenges, curtail power usage, high data reliability, optimized network productivity.**

Keywords—Wireless sensor network, Equi-Quadrant division clustering protocol, Coverage hole, Mobile nodes, Minimum spanning tree, Delaunay Triangulation.

I. INTRODUCTION

The term "wireless sensor network" (WSN) refers to a system of spatially separated and specialized sensors that work together to cooperatively transmit their data to a central location while simultaneously monitoring and recording the physical conditions of the environment. Small, inexpensive, and low-power sensor nodes are the basic building blocks of WSNs, which are widely dispersed throughout the study area. Each sensor node has a sensor, a transceiver, a processing unit, and a power supply. Fig. 1 represents the basic architecture of wireless sensor network. The processing unit converts the analogue signal from the sensor's measurement of the desired physical quantity to a digital signal. The sink node or the adjacent nodes receive the digital signal that is transmitted by the transceiver. WSNs are used for a variety of purposes, including real-time data collection and decision-making, including environmental monitoring, industrial automation, healthcare, agriculture, and smart cities.

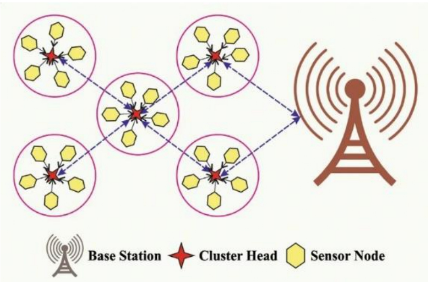

Fig. 1 Basic architecture of wireless sensor network

Achieving complete sensor coverage is one of the major challenges in wireless sensor network. A coverage hole arises in the WSN if there is a region that is not covered by any nodes. Anywhere in a WSN area, at any moment, coverage gaps can occur for a variety of reasons. Nodes are deployed either randomly or in predefined locations. Here we consider random deployment, During random deployment in some parts density of sensor becomes lower causing the emergence of coverage holes. Also, it is possible for some nodes to lose power and stop interacting with the network. The damaged area develops a hole. The physical damage of sensor nodes is another factor in the establishment of coverage holes. Additionally, there may be places in the region of interest where neurons are unlikely to operate normally. By deploying mobile nodes, coverage holes in wireless sensor networks (WSNs) are mitigated. Mobile nodes are positioned in under-covered areas to increase the sensing and communication range by detecting those areas. Mobile nodes can address coverage holes by strategically

moving and repositioning themselves, ensuring thorough network monitoring and data collecting. To maximise the efficiency of deploying mobile nodes for coverage mitigation in WSNs, however, great attention should be paid to balancing energy consumption, network connectivity, and effective movement techniques.

Due to the constrained power resources of sensor nodes, energy efficiency is another crucial challenge in wireless sensor networks (WSNs). To maximise the network's lifespan and guarantee optimal functioning, a number of energy efficiency challenges must be resolved. Sensing and communication processes use a lot of energy, which can quickly deplete the node's battery. Data transmission is another concern because it uses a lot of energy. Various energy-efficient protocols and algorithms have been developed to address these problems. In our proposed system the EQDC protocol divides the network into 4 equal quadrants, each having cluster heads(CH) which transmits the data collected to the base station. Information from sensor nodes is transmitted to the nearest cluster head. Later cluster heads are connected to base station by minimum spanning tree for data transmission.

Metrics such data dependability, ideal power consumption, memory constraints, and data latency in wireless sensor networks (WSNs) provide significant implementation issues. The implementation of an effective network for wireless sensor networks is hampered by factors such data reliability, optimal power consumption, storage constraints, and data latency

II. REVIEW OF RELATED LITERATURE

1) A Hybrid Fault-Tolerant Routing based on Gaussian Network for Wireless Sensor Network [1]

Using hybrid fault tolerant routing, the problem of fault tolerance in wireless sensor networks (WSNs) with hierarchical topology was tackled. The hierarchical topology is formed by grouping sensor nodes and representing them as Gaussian integers collectively. Because of this, the network region is split into small square grids, with a Gaussian integer acting as each grid's cluster head. A Gaussian network is built by connecting these cluster heads together. This study proposes a hybrid fault-tolerant clustering routing protocol for wireless sensor networks that utilises the shortest path in the Gaussian network, node symmetry, and the advantages of multi route routing (FCGW). For wireless sensor networks, FCGW aims to increase fault tolerance, boost data dependability, and cut down on energy use. The FCGW protocol provides great data reliability, according to experimental results of the suggested system. Additionally, the FCGW protocol uses 48% of the network's energy, whereas other protocols use 70% more energy . The network as a whole is separated into many clusters in this approach, which maximises WSN's energy efficiency. The network diameter and the average distance in between a pair of nodes may both be very slight.

2) Energy efficiency clustering based on Gaussian network for wireless sensor network [2]

In wireless sensor networks, the routing protocol is designed with the sensors' connection model and energy efficiency in mind. Accordingly, the study suggests a novel wireless sensor network connection model, in which the network area will be divided into some virtual square grids, based on the node-symmetric and four different directions to four nearby nodes of each node in the Gaussian network model. The authors propose a routing method that combines the shortest path routing protocol in the Gaussian network with clustering protocol to increase the routing effectiveness of the wireless sensor network. In the new wireless model, each virtual square grid is described as a node in the Gaussian network. The proposed routing is effective, according to simulations carried out in NS2. This routing approach is suggested in the design of the sensors' connection model and energy. which compiles the clustering protocol with the shortest path routing protocol in a gaussian network, hence increasing the WSN's routing effectiveness.

3) On the Utilization of Shortest Paths in Complex Networks [3]

The analysis of network architecture, connection patterns, and their impact on network dynamics has received a lot of attention. The notion that traffic takes the shortest routes linking pairs of distant vertices is common in network analysis models. Examples of algorithms that make use of this premise include community extraction techniques, core-periphery identification algorithms, and graph centrality metrics. However, since it places so much emphasis on shortest path communications, this viewpoint is quite constrained and can be deceptive. In this paper, how shortest paths are used in various data dissemination scenarios in complex networks is investigated. Also investigates whether there are any universal characteristics that might improve how well networks use the shortest pathways. Here, the use of shortest paths in complicated networks demonstrated may not be as widespread as expected by running simulations on a collection of real-world and artificial networks. This suggests that longer pathways may occasionally be just as crucial as the quickest ones. The findings demonstrate that at least two elements—network topology and data dissemination algorithm—clearly affect shortest path utilisation in a network. Additionally, it discovers that a network's nature is not a reliable predictor of its shortest path utilisation.

4) DORA: A Destination Oriented Routing Algorithm for Energy-Balanced Wireless Sensor Networks [4]

The destination-oriented routing algorithm (DORA), a new multi-chain routing technique, is used. The suggested approach makes advantage of the Power-Efficient Gathering in Sensor Information Systems (PEGASIS) chain routing feature, which chooses the preferred forwarding node by taking into account the possible transmission power and the path from the source node to the sink. To transfer packets for energy-balanced WSNs, the suggested solution creates a fresh multi-chain routing method. The mathematical analysis model obtains the ideal transmission span between any two nodes based on multi-chain routing. In order to build the multi-chain structures by precise direction, distance, and many paths, the design criterion for the best distance is to choose the farthest transmitting node within the communication vicinity and the path to the sink. Unnecessary energy loss can be decreased with the shorter chain routing to increase the lifespan of the global network. Testing results show that the suggested DORA enhances the RPC protocol's longevity by 60% and doubles the network lifespan when compared to the traditional PEGASIS. The optimum multichain routing strategy-based transmission distance between any two nodes.

5) Learning-based Adaptive Sensor Selection Framework for Multisensing WSN [5]

Multiple sensor wireless sensor nodes frequently have a low energy capacity. A unique adaptive sensor selection strategy is suggested in order to maximize the energy sustainability of such sensor hubs. Cross-correlation between several sensors measuring various characteristics in the same environment is frequently present, making the system predictive. In order to achieve this, a learning-based optimization strategy based on the Upper Confidence Bound algorithm is created in order to choose the best active sensor set in a measurement cycle based on the cross-correlations between the parameters, energy used by the sensors, and energy available at the node. The parameter values of inactive sensors are further predicted from the cross-correlated parameters of active sensors using a Gaussian process regressor-based prediction model. An air pollution monitoring sensor node with seven sensors has been installed on campus to test the performance of the suggested framework in practical applications. This sensor node gathers data at a high sampling rate by default. The effectiveness and efficiency of the suggested framework are confirmed by the simulation results. The proposed algorithm is 54% more energy efficient than the state-of-the-art. Utilizing reinforcement for a small sensor hub with numerous sensors measuring various environmental conditions.

6) Wireless-Sensor Network Topology Optimization in Complex Terrain: A Bayesian Approach [6]

Approaches for wireless sensor network (WSN) topology enhancement rely on the oversimplifying assumption that there is a fixed communication radius between network nodes, which makes them ineffective for IoT networks placed in complicated terrain. Using an in-situ connection quality assessment and a Bayesian link classifier that relies on LIDAR-derived topographical features, a data-driven approach is proposed for WSN topology optimisation. The classifier is trained to predict where strong network interconnections (packet delivery ratio, PDR>0.5) are most likely to occur given complex terrain factors. Using a large number of suggested wireless node placements across the domain, the classifier is then used to generate an undirected weighted network that represents the possible connection across the region. The probability that two nodes will form a strong link is inversely proportional with edge weights in the connectivity network. The revolutionary modified cycle-union (MCyU) approach is then used to the undirected weighted graph of potential network component placements to produce a 2-vertex linked, Steiner minimal network. This increases the likelihood of strong links in the final network while ensuring a resilient network design. The overall number and spatial distribution of network components produced by the algorithm are contrasted to an existing wireless sensor network used for regional environmental monitoring. Additionally, three graph test cases were used to assess the MCyU method in order to compare it to state-of-the-art remedies. MCyU outperformed the alternatives in terms of weight minimization and computing time. It creates and evaluates a technique for WSN topology optimization in difficult terrain.

7) Node position estimation for efficient coverage hole-detection in wireless sensor network [7]

In many applications, sensor nodes are often distributed at random. As a result, estimating node placement in WSNs is a real issue since uncovered areas might lead to coverage gaps in the network. The Global Positioning System (GPS) is one cutting-edge solution to this issue, although using GPS to localise a node may not be worthwhile due to the expense of additional hardware and power needs. Therefore, a computational geometry-based technique to localising a node may be a low-cost solution to this issue. In this study, we use RSSI profiling to determine the distance between the Anchor Node (AN) and Unknown Node (UN). Trilateration is then used to estimate the node's position. The Delaunay Triangle is then built using the knowledge of the node's position. Then, in order to determine if a coverage hole exists in the provided ROI or not, the property of an empty circle is utilised. Based on simulations and theoretical justifications, the algorithm's correctness is evaluated.

8) Enhanced LEACH protocol for increasing a lifetime of WSNs [8]

Numerous low-cost and low-power sensing devices are a part of wireless sensor networks, but local processing, the ability of wireless connection, and the network's energy efficiency all have issues. In order to reduce power consumption in cluster head nodes and throughout the entire network, the goal of this paper is to enhance the LEACH (low energy adaptive clustering hierarchy) protocol, which is the cluster routing protocol. This is done by identifying a cluster head according to the lowest degree of distance from the base station. Thus, the findings demonstrate LEACH's capacity to increase network lifespan and lowering power usage.

9) Improving lifetime of wireless sensor networks based on nodes' distribution using Gaussian mixture model in multi-mobile sink approach [9]

Energy conservation in Wireless Sensor Networks (WSNs) is essential for a variety of applications, including maintaining human consciousness and monitoring the environment. Numerous studies have examined energy usage and found that lowering energy consumption increased the lifespan longevity of WSNs. However, proposed methods ignore a crucial aspect of a WSN—the nodes' distribution function in the energy model and routing protocol. In this work, a novel method called GDECA is presented, that operates under the real-world presumption that nodes' distributions are mashups of Gaussian distributions. Therefore, in order to fit the Gaussian Mixture Model (GMM) to the nodes and determine the parameters for these distributions, GDECA uses a distribution estimation technique that it adopted from machine learning (ML). Next, the Cluster Head CH selection strategy uses the calculated parameters. Additionally, the dispersion of nodes is used to calculate sinks routing. Results indicated that energy usage has improved by about 40–50%. Another result of GDECA is that all nodes remain active during the experiment. Observations also show that this method is best for calculating sinks' paths, and that altering the number of sinks at random makes the system use more energy.

10) Enhancing the Lifetime of Wireless Sensor Networks Using Fuzzy Logic LEACH Technique-Based Particle Swarm Optimization [10]

To extend the use of WSNs in more applications, they must have a longer lifespan. Clustering with the optimum cluster head (CH) is one of the most efficient techniques for enhancing the lifetime of the network. This paper suggests a particle swarm optimization (PSO) method based on fuzzy logic (FL) and the low-energy adaptive clustering hierarchy (LEACH) technology. For cluster generation, it uses a hybrid PSO and the K-means clustering method. FL is used to choose the primary CH (PCH) and secondary CH (SCH). A simulation program was used to run several simulations in order to verify the effectiveness of the suggested methodology.

11) APTEEN routing protocol optimization in wireless sensor networks based on combination of genetic algorithms and fruit fly optimization algorithm [11]

The APTEEN routing protocol has issues with unequal network energy consumption, node early mortality, consuming excessive amounts of energy needlessly, and insufficiently effectively covering the entire network. In order to address these issues, this study combines a genetic algorithm with a fruit fly optimisation technique to optimise the APTEEN routing protocol. The genetic algorithm and fruit fly optimisation algorithm are used to choose cluster heads for the first time, and the second time is based on density adaptive algorithm, adding residual energy, distance from node to base station, distance from node to geometric centre of the entire network, node degree, and other selection factors to cluster heads selection. Based on the position and number of nodes, certain nodes are chosen to sleep. When nodes join clusters, the left-over energy of the cluster head, the distance between the node and the cluster head, and the total number of cluster members are all taken into consideration. The Dijkstra algorithm determines the best path for energy transmission from cluster heads to base stations. The GA-APTEEN routing protocol is achieved by the aforementioned optimisation by adding the rule of rotating cluster heads when the energy used for data transmission is too high. According to the simulation findings, GA-APTEEN increases the network's 50% lifespan, 10% coverage, and resilience, lowers the system's total energy consumption, and prevents the hot zone of energy phenomena.

III. PROPOSED SYSTEM

A. NODE PARTICIPATION:

Node participation is a fundamental aspect of wireless sensor networks (WSNs), ensuring efficient data collection and transmission. In a WSN, nodes play a crucial role in sensing and monitoring the surrounding environment. Each node is equipped with sensors that measure various parameters such as temperature, humidity, pressure, or light intensity. Through active participation, these nodes continuously gather data from their respective locations, providing valuable insights into the monitored area.

In our proposed system, nodes are distributed randomly in a predefined network area. In a randomly distributed wireless sensor network (WSN), nodes are typically scattered across the deployment area without any predetermined structure or pattern. As a result, certain regions within the network may exhibit varying degrees of node density. These regions can be classified into sparse and dense regions based on the distribution of participating nodes. Sparse regions in a randomly distributed WSN refer to areas where the node density is relatively low. In these regions, the distance between neighboring nodes tends to be greater, resulting in a sparser coverage of the environment. On the other hand, dense regions in a randomly distributed WSN exhibit higher node density. These regions are characterized by nodes being closely located to one another, resulting in a more comprehensive sensing coverage. In dense regions, data collection and transmission are generally more efficient due to the shorter distances between nodes.

In order to attain complete coverage throughout our proposed system, Delaunay triangulation is used to find the coverage holes. Delaunay triangulation, a geometric method that connects sensor nodes through triangles, can reveal areas where the network lacks sufficient coverage. Fig. 2 shows the Delaunay triangulation done before mobile node placement. By leveraging this information, mobile nodes can be strategically placed within these coverage holes to fill the gaps and enhance the overall network performance.

The placement of mobile nodes in coverage holes aims to optimize network coverage and data collection efficiency. These mobile nodes can possess additional sensing capabilities or serve as relays to facilitate data transmission between nearby stationary nodes. By deploying mobile nodes in coverage holes, the WSN can extend its coverage range, ensuring that important areas are adequately monitored and reducing the chances of data loss due to insufficient coverage. Mobile nodes are placed on the midpoint of the longest edge of each triangle where coverage hole is found, represented in Fig. 3. Fig. 4 shows the retriangulated network after mobile node placement. This approach enhances the connectivity within the network and contributes to a more robust and reliable wireless sensor network infrastructure.

Delaunay Triangulation Algorithm

1: Create an empty triangulation T.
T = EmptyTriangulation()

2: Compute the convex hull of the points in P and add the boundary edges to T.
convex_hull_edges = ComputeConvexHull(P)
T.add_edges(convex_hull_edges)

3: Sort the points in P based on their x-coordinate in ascending order.
sorted_points = SortPoints(P)

4: For each point pi in P for pi in sorted_points:
a) Find the neighboring points of pi to the right in the sorted order.

neighboring_points=FindNeighboringPoints(pi, sorted_points)
b) For each point pj in neighboring_points
for pj in neighboring_points:
b(i): Check if the circumcircle of (pi, pj, pk) contains no other points in P
circumcircle_empty = CheckCircumcircleEmpty(pi,pj,P)
b(ii): If circumcircle is empty, create a new triangle and add it to T.
if circumcircle_empty:
pk = FindAnyOtherPoint(P)
triangle = CreateTriangle(pi, pj, pk)
T.add_triangle(triangle)

5: Output the triangulation T, which represents the Delaunay triangulation of the input points P.
Output(T)

Coverage Hole Identification and Patching Algorithm

1: Create an empty list of coverage holes.
coverage_holes = []
2: For each edge e in the triangulation T
for e in T.edges:
a) Calculate the length of the edge |e|.
edge_length = CalculateEdgeLength(e)

b) Calculate the radius of the sensor node r as half of the communication range.
r = CalculateSensorNodeRadius()

*c) If |e| > 2 * r, a coverage hole is detected.*
if edge_length > 2 * r:
coverage_holes.append(e)
 # Add the edge e to the list of coverage holes.

3: For each coverage hole in the list
for coverage_hole in coverage_holes:
a) Calculate the midpoint of the edge.
midpoint = CalculateMidpoint(coverage_hole)

b)Place a mobile node at the midpoint to patch the coverage hole.
PlaceMobileNode(midpoint)

4: Output the patched sensor network with no coverage holes.
OutputPatchedSensorNetwork()

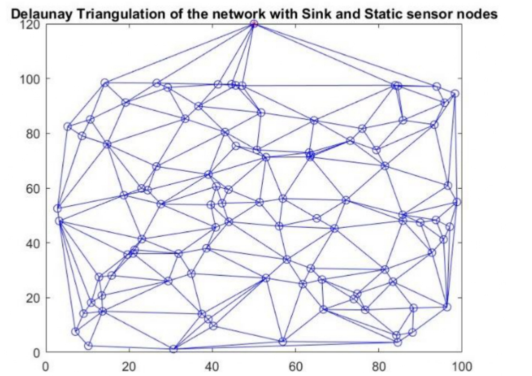

Fig.2 Delaunay triangulation before placement of mobile node

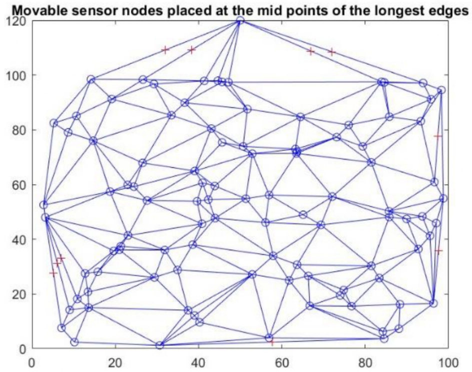

Fig. 3 Mobile nodes are placed on the midpoint of longest edge

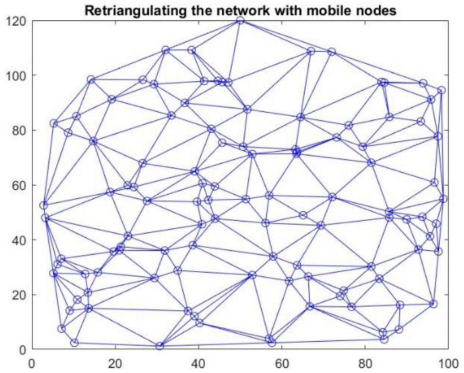

Fig. 4 Delaunay triangulation after placing mobile node (improved participation)

B. EQDC PROTOCOL:

Our proposed system introduces the EQDC (Equi-Quadrant Division Clustering) protocol, an optimized clustering algorithm specifically designed for wireless sensor networks (WSNs). The primary objective of our system is to achieve efficient data transmission distribution while minimizing energy consumption. In the EQDC protocol, the entire network area is divided into four equal quadrants, with each quadrant being assigned a dedicated cluster head. These cluster heads play a vital role in aggregating and forwarding data within their respective quadrants, ensuring effective data management and communication. Through our proposed system, we aim to enhance the performance and energy efficiency of WSNs by leveraging the benefits of the EQDC protocol.

In our proposed network model, the first step involves dividing the network area into equal quadrants to facilitate efficient clustering. Each quadrant follows a set of rules to determine its cluster head (CH). If a node in a quadrant satisfies the election factor and has an energy level above a predefined threshold, it becomes the CH. However, if the energy level of a CH falls below the given threshold, a CH re-election process takes place to ensure that the network remains stable.

The regular nodes in the network connect to the CH based on their distance to the CH and the base station. If a node's distance to the CH is greater than its distance to the base station, it directly connects to the base station. This approach helps to reduce the overhead on the CH, as certain nodes can bypass the CH and transmit their data directly to the base station.

To establish an optimal path for data transmission to the base station, we employ a minimum spanning tree (MST) to connect the CH and the base station. The MST algorithm constructs a tree that covers all the nodes while minimizing the total path cost. By utilizing this MST-based connection, we ensure an efficient and optimized data routing path from the CHs to the base station. Fig. 5 shows the implementation of EQDC protocol in a network.

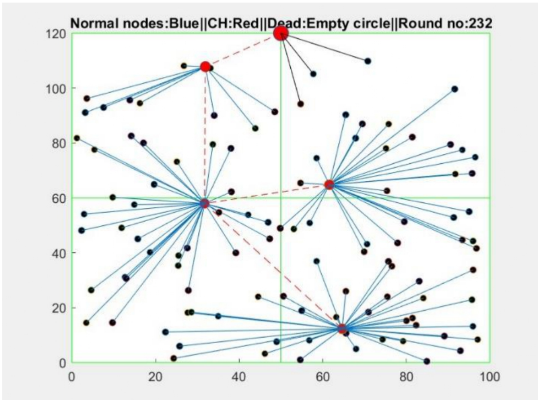

Fig. 5 EQDC protocol in node participation improved network

By dividing the network into quadrants and leveraging the concept of nearest cluster heads, EQDC enables efficient data transmission and minimizes energy consumption in WSNs. EQDC achieves improved energy efficiency and network lifetime also by effectively reducing the distance travelled for data transmission. The EQDC protocol has demonstrated superior performance and yielded better results compared to other existing methods in wireless sensor networks (WSNs).

Algorithm for EQDC Protocol

Input: nodes after placing mobile nodes(mn)
1: area(a)
Divide area into equal quadrants
number of quadrants - Q
number of nodes - n
set election factor(e)
Initial energy(E_o)
CH threshold(E_t)
2: for a given quadrant
{

```
  CH=0
  for 1:n
  if (election condition is true & CH=0 & E>Et )
  {
  node type=cluster head
  CH++
  }
3: for 1:n
   if(Node type = normal node)
   {
    if(distance to BS< Distance to CH)
   send to BS
   else
   send to CH
   }
5: for 1:number of CH
   {
    path=mst(cluster heads)
   }
6: send to BS from CH
```

IV. SIMULATION AND ANALYSIS

The performance of our proposed system is evaluated by simulating the wireless sensor network in MATLAB. In this simulation, an area of 100 x 120 is predefined. 100 static sensor nodes are randomly deployed in this area. 10 mobile nodes are deployed and moved to the midpoint of the longest edge of each triangle where coverage hole is found. Base station is fixed at the coordinate (50,60). We evaluated the performance of our proposed system against some existing protocols like FL-LEACH & GA-APTEEN. The results of evaluation are given below.

1) *Enhanced node involvement:*

Mobile nodes deployed strategically within coverage holes play a crucial role in enhancing node participation in wireless sensor networks (WSNs). These mobile nodes are strategically positioned to identify areas lacking sufficient coverage and actively fill those gaps. By dynamically adjusting their locations, they effectively bridge the coverage holes, resulting in improved node participation and data transmission. The presence of mobile nodes not only extends the reach of the network but also facilitates seamless communication and collaboration among previously isolated sensor nodes.

2) *Throughput:*

It represents the capacity and efficiency of the network to successfully deliver data from source nodes to destination nodes over a specific time period. Throughput is influenced by various factors, including the network topology, communication protocols, packet size, channel conditions, and interference levels. Our proposed solution demonstrates superior throughput performance compared to existing methods in wireless sensor networks. Comparison of throughput results are shown in the fig. 6.

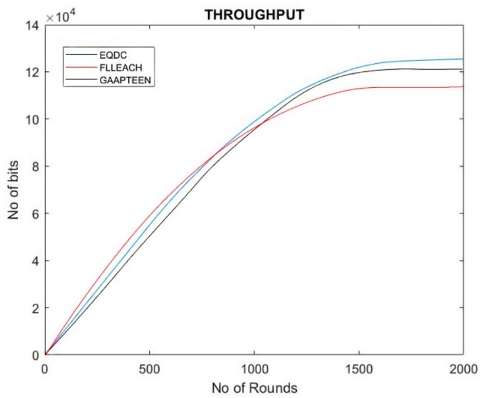

Fig. 6 Comparison of throughput result in proposed system with FL-LEACH and GA-APTEEN

3) *Residual Energy:*

Residual energy in a wireless sensor network (WSN) refers to the remaining amount of energy within the sensor nodes after a specific period of operation. This information helps in implementing strategies such as energy-efficient protocols, node sleep scheduling, or dynamic power management techniques to extend the network's lifespan and ensure reliable operation of the WSN. Residual energy in proposed system and other techniques are shown in the fig. 7. Effectively managing residual energy ensures the sustainability and efficiency of the WSN in various applications.

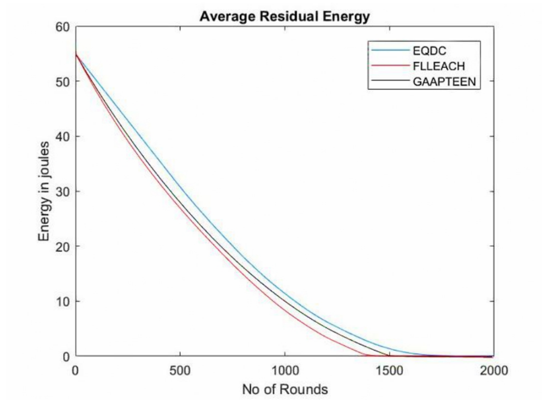

Fig. 7 Average residual energy of sensors in EQDC, FL-LEACH & GA-APTEEN protocols

V. CONCLUSION

In the mentioned paper, a novel clustering protocol called EQDC is implemented in a wireless sensor network to improve node participation. The protocol divides the network into four equal quadrants and utilizes mobile nodes placed in coverage holes to enhance participation. Data transmission occurs based on the shortest distance, either directly from nodes to sinks or through cluster heads to the base station. The proposed system demonstrates higher energy efficiency and reliability compared to traditional clustering protocols. However, in certain scenarios where quadrants fail to meet cluster head detection conditions, nodes in those areas connect to the nearest cluster head in the adjacent quadrant, potentially overloading the connected cluster head. Additionally, there may be nodes with residual energy in the final rounds that cannot transfer data due to cluster formation conditions, leading to packet loss. In future research, our focus will be on addressing the aforementioned challenges and improving the performance of our WSN network model.

VI. ACKNOWLEDGEMENTS

We would like to express our sincere gratitude and appreciation to all those who have played significant role in the successful completion and publication of our project titled "An Energy Efficient Wireless Sensor Network with EQDC Protocol and Improved Node Participation."

REFERENCES

[1] D. N. Quoc, N. Liu and D. Guo, "A hybrid fault-tolerant routing based on Gaussian network for wireless sensor network," in *Journal of Communications and Networks*, vol. 24, no. 1, pp. 37-46, Feb. 2022, doi: 10.23919/JCN.2021.000028.

[2] Nguyen Quoc, D., Bi, L., Wu, Y., He, S., Li, L. and Guo, D. (2019), Energy efficiency clustering based on Gaussian network for wireless sensor network. IET Communications, 13: 741-747. doi.org/10.1049/iet-com.2018.5398

[3] H. Alrasheed, "On the Utilization of Shortest Paths in Complex Networks," in *IEEE Access*, vol. 9, pp. 110989-111004, 2021, doi: 10.1109/ACCESS.2021.3101176.

[4] K. Wang, C. -M. Yu and L. -C. Wang, "DORA: A Destination-Oriented Routing Algorithm for Energy-Balanced Wireless Sensor Networks," in *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 2080-2081, 1 Feb.1, 2021

[5] S. Ghosh, S. De, S. Chatterjee and M. Portmann, "Learning-Based Adaptive Sensor Selection Framework for Multi-Sensing WSN," in *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13551-13563, 15 June15, 2021, doi: 10.1109/JSEN.2021.3069264.

[6] Carlos A. Oroza; Jairo A. Giraldo; Masood Parvania; Thomas Watteyne C. A. Oroza, J. A. Giraldo, M. Parvania and T. Watteyne, "Wireless-Sensor Network Topology Optimization in Complex Terrain: A Bayesian Approach," in IEEE Internet of Things Journal, vol. 8, no. 24, pp. 17429-17435, 15 Dec.15, 2021,

[7] Das, Smita, and Mrinal Kanti DebBarma. "Node position estimation for efficient coverage hole-detection in wireless sensor network." *Computación y Sistemas* 23.1 (2019): 185-195.

[8] Abu Salem, Amer O., and Noor Shudifat. "Enhanced LEACH protocol for increasing a lifetime of WSNs." *Personal and Ubiquitous Computing* 23 (2019): 901-907.

[9] Hojjatinia, Houriya, Mohsen Jahanshahi, and Saeedreza Shehnepoor. "Improving lifetime of wireless sensor networks based on nodes' distribution using Gaussian mixture model in multi-mobile sink approach." *Telecommunication Systems* 77 (2021): 255-268.

[10] Gamal, Marwa, et al. "Enhancing the lifetime of wireless sensor networks using fuzzy logic LEACH technique-based particle swarm optimization." *IEEE Access* 10 (2022): 36935-36948.

[11] Wang, Minghao, Shubin Wang, and Bowen Zhang. "APTEEN routing protocol optimization in wireless sensor networks based on combination of genetic algorithms and fruit fly optimization algorithm." *Ad Hoc Networks* 102 (2020): 102138.

BIBLIOGRAPHY

[1] D. N. Quoc, N. Liu, and D. Guo, “A hybrid fault-tolerant routing based on gaussian network for wireless sensor network,” *Journal of Communications and Networks*, vol. 24, no. 1, pp. 37–46, 2022.

[2] D. Q. Nguyen, L. Bi, Y. Wu, S. He, L. Li, and D. Guo, “Energy efficiency clustering based on gaussian network for wireless sensor network,” *IET Journals*, vol. 13, no. 11, pp. 1625–1632, 2019.

[3] H. Alrasheed, “On the utilization of shortest paths in complex networks,” *IEEE Access*, vol. 9, pp. 110 989–111 004, 2021.

[4] K. Wang, C.-M. Yu, and L.-C. Wang, “Dora: a destination-oriented routing algorithm for energy-balanced wireless sensor networks,” *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 2080–2081, 2020.

[5] S. Ghosh, S. De, S. Chatterjee, and M. Portmann, “Learning-based adaptive sensor selection framework for multi-sensing wsn,” *IEEE Sensors Journal*, vol. 21, no. 12, pp. 13 551–13 563, 2021.

[6] C. A. Oroza, J. A. Giraldo, M. Parvania, and T. Watteyne, “Wireless-sensor network topology optimization in complex terrain: a bayesian approach,” *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 429–17 435, 2021.

[7] S. Das and M. Kanti DebBarma, “Node position estimation for efficient coverage hole-detection in wireless sensor network,” *Computación y Sistemas*, vol. 23, no. 1, pp. 185–195, 2019.

[8] D. Thomas, R. Shankaran, M. A. Orgun, and S. C. Mukhopadhyay, “Sec 2: A secure and energy efficient barrier coverage scheduling for wsn-based iot applications,” *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 622–634, 2021.

[9] H. Hojjatinia, M. Jahanshahi, and S. Shehnepoor, "Improving lifetime of wireless sensor networks based on nodes' distribution using gaussian mixture model in multi-mobile sink approach," *Telecommunication Systems*, vol. 77, pp. 255–268, 2021.

[10] P. Sharma, R. P. Singh, M. A. Mohammed, R. Shah, and J. Nedoma, "A survey on holes problem in wireless underground sensor networks," *IEEE Access*, vol. 10, pp. 7852–7880, 2021.

[11] A. O. Abu Salem and N. Shudifat, "Enhanced leach protocol for increasing a lifetime of wsns," *Personal and Ubiquitous Computing*, vol. 23, pp. 901–907, 2019.

[12] M. Gamal, N. E. Mekky, H. Soliman, and N. A. Hikal, "Enhancing the lifetime of wireless sensor networks using fuzzy logic leach technique-based particle swarm optimization," *IEEE Access*, vol. 10, pp. 36 935–36 948, 2022.