**A Project Report**

On

# CLUSTERING IN WIRELESS SENSOR NETWORK USING GENETIC ALGORITHM TO ENHANCE NETWORK LIFETIME

Submitted in partial fulfilment of the requirements for the award of the Degree of

**BACHELOR OF ENGINEERING**

IN

**COMPUTER SCIENCE AND ENGINEERING**

BY


**Ms. Mathari Pavani (16241A0597)**

**Ms. Somasri Harshitha (16241A05A7)**

**Ms. Srujana Sabbani (16241A05A9)**

**Ms. Nalli Suvarna (17245A0520)**


Under the Esteemed guidance of


**T.V. Suneetha**

**Assistant Professor**

**Department of CSE**

**GRIET, Hyderabad.**




**Department of Computer Science and Engineering**

**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous) Bachupally, Kukatpally, Hyderabad- 5000902019-2020.**

## Department of Computer Science and Engineering

## GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

## (Autonomous) Bachupally, Kukatpally, Hyderabad- 5000902019-2020.


# CERTIFICATE

This is to certify that the project work entitled **"Clustering in Wireless Sensor Network using Genetic Algorithm to enhance network lifetime"** is a bonafide work carried out by **Ms. Mathari Pavani (16241A0595), Ms. Somasri Harshitha (16241A05A7), Ms. Srujana Sabbani (16241A05A9), Ms. Nalli Suvarna (17245A0520)** in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING** from Gokaraju Rangaraju Institute Of Engineering & Technology College (Autonomous), Bachupally, Kukatpally, Hyderabad, under our guidance and supervision**.**

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

| | |
|---|---|
| Internal Guide | Head of the Department |
| **T.V.Suneetha** | **Dr.K.Madhavi** |
| **Assistant Professor** | **Professor and Head** |
| **Department of CSE** | **Department of CSE** |
| **GRIET, Hyderabad.** | **GRIET, Hyderabad.** |

# DECLARATION

This is to certify that the work reported in the present project entitled **"Clustering in Wireless Sensor Network using Genetic Algorithm to enhance network lifetime."** is a record of bonafide work done by us/ me in the Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The reports are based on the project work done entirely by us and not copied from any other source.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our/ my knowledge and belief.

| Signature | Signature | Signature | Signature |
|---|---|---|---|
| Mathari Pavani | Somasri Harshitha | Srujana Sabbani | Nalli Suvarana |
| (16241A0597) | (16241A05A7) | (16241A05A9) | (17245A0520) |

## ACKNOWLEDGEMENT

We / I would like to express my sincere gratitude and indebtedness to my project guide T.V. Suneetha for her valuable suggestions and interest throughout the course of this project.

We are / I am also thankful to our principal Dr. J. Praveen and Dr. K. Madhavi, Professor and Head, Department of Computer Science and Engineering, GRIET Engineering College, Hyderabad for providing excellent infrastructure and a nice atmosphere for completing this project successfully as a part of our B.E. Degree (CSE).

We / I convey our/my heartfelt thanks to the lab staff for allowing me to use the required equipment whenever needed.

Finally, we/ I would like to take this opportunity to thank my family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

**Mathari Pavani (16241A0595)**

**Somasri Harshit(16241A05A7)**

**Srujana Sabbani (16241A05A9)**

**Nalli Suvarna (17245A0520)**

# LIST OF FIGURES

# LIST OF TABLES



# INDEX TERMS

| | |
|---|---|
| **BS** | **Base Station** |
| **CH** | **Cluster Head** |
| **WSN** | **Wireless Sensor Network** |
| **GA** | **Genetic Algorithm** |
| **LEACH** | **Low Energy Adaptive Clustering Hierarchy** |

# ABSTRACT

WSN is one of the most common communication tools used in many areas at the life, in both civilians and militaries. These wireless networks consistent of a huge number of very small and powerful devices called **sensor nodes**. The sensor nodes communicate using wireless strategies. These sensor node are powered using batteries. So they have huge effect on communication strategies due to energy issues making them un-reliable. In this network, CH's are burdened with congestion than any other component of the network. This puts a lot of work load on a cluster heads that are nearest to the base station. This forms the main reason to elect an efficient CH using in a clustered based routing protocol.

Our project works by applying the evolutionary algorithm such as **"Genetic Algorithm"** to solve these energy specific problems in **WSN**. The Proposed protocol is implemented using **MATLAB(a simulating environment),** to show performance analysis between **GA**, **LEACH** and **K-Means Clustering.**

In this project, we also worked on **LEACH** with multiple stations to increase the lifetime of the network. CH will transmit the data signals to the nearest BS, thus reducing the overall energy consumption.

**CONTENTS** **Page No**

.

# 1. INTRODUCTION

## 1.1 Wireless sensor network

The original motivation for WSN analysis stemmed from the vision of Smart Dust in the late 90's. This vision entailed mainly on integrated computing, communication, and sensing platform consisting of many small devices, resulting in the applications like as dense environmental monitoring and sensible home/office.

Whenever sensor nodes connect with controllers and process stations directly (e.g., using LAN), these large number of sensors communicate the collected data wirelessly to a centralized process station. This is often necessary since several network applications need hundreds or thousands of sensing nodes, usually deployed in remote and inaccessible areas.

With these enhancements, a sensing node does information assortment, in-network analysis, correlation, and fusion of its own perceived information and information from remaining nodes. These several sensors nodes together monitor the massive physical environments, typically known as Wireless Sensor Network (WSN). Sensor nodes communicate with other nodes as well as with a base station (BS) with the help of their wireless transreceiver unit, permitting them to disseminate their sensor information to remote processing, virtualization , analysis, and storage systems.

Wireless networks is the trending technology that may enable the users to get access to the services and information electronically, without depending on the location of nodes. The sensor nodes have considerably lower communication and computation capabilities than the fulyl-featured computers collaborating in circumstantial networks.

Due to their scattered layout model, the energy supply of the device node is taken into account non-renewable (although some device nodes may be able to scavenge resources from their environment). Routing protocols deployed in these networks to contemplate the matter of efficient use of power resources.

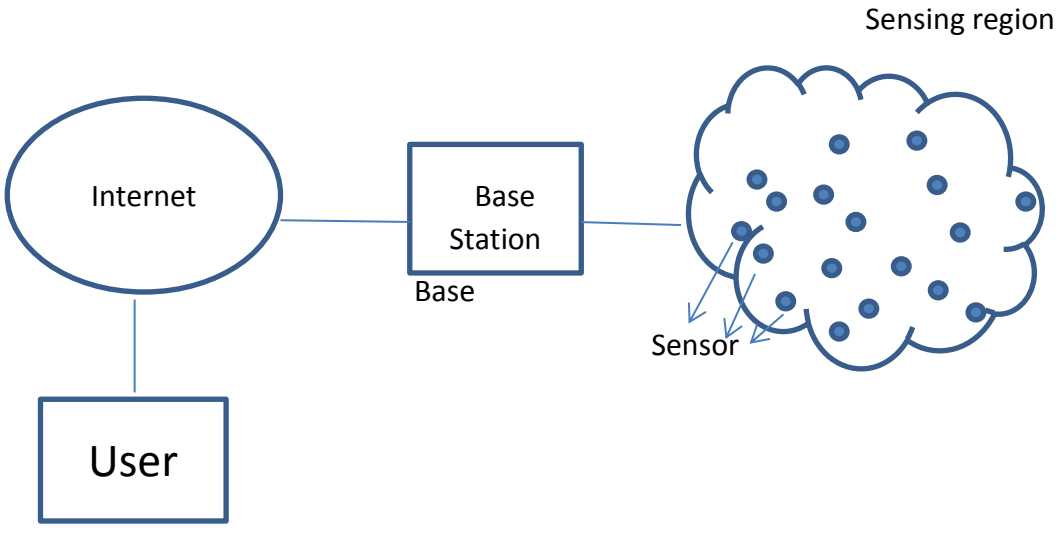


**Figure: 1.1**

## 1.1.1 Components of WSN

The WSN consists of

- sensor nodes and
- sink (Base Station).

**Base Station (Sink)**

The sink is intermediatary between the external world (or user) and computational world (sensor network). It is normally a resourceful node having un-interrupted computational capabilities and a plethora of energy supply. These WSN can have single or multiple base stations. In our project, we have made use of four base stations instead of one base station which decreased the network delay and proved the increased performance of the WSN.

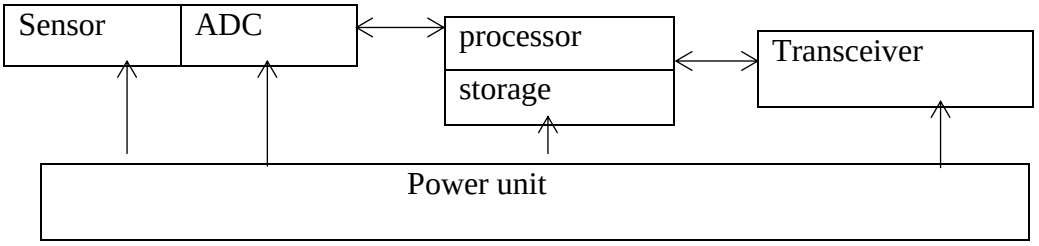


**Figure: 1.2**

**Sensor Nodes**

As shown in the above figure 1.2, a sensor node is comprised of 4 sub-units sensing unit, a transmitting and receiving unit, a power unit and a process unit.

The sensing units are usually composed of two sub-units: Sensing part and ADC (analog-to-digital converters). The ADC converts the analog signals perceived by the sensor into digital signals. These digital signals are then processed by processing unit. The processing unit consists of a storage device. This sensor node collaborates with other node to transmit the data to the base station.

### 1.1.2 Applications of Wireless Sensor Networks

- Area monitoring
- Health care monitoring
- Environmental/Earth sensing
- Air pollution monitoring
- Forest fire detection

## 1.2 LEACH

In unsupervised learning algorithm initially used for data mining is k-means clustering. It was proposed by Mac Queen, 1967. This algorithm works by partitioning the objects into clusters such that intra-cluster similarities are maximized and inter-clustering similarities are minimized using Euclidean distance. Initially, it creates k enters of the cluster where k is the number of clusters formed. Now each object or data point that is closest to the centers is assigned to that data center. The center of the cluster is changed to the mean of all the objects in its cluster. The Euclidean formula is given below to calculate the distance between the objects.

- $T(n)= \begin{cases} \frac{p}{1-p*(r \bmod \frac{1}{p})} & \text{if } n \in G \\ 0 \quad otherwise \end{cases}$ (1.1)

Where r is the round number, p is the probability of a node to becoming a CH, and G is the set of nodes that have not been chosen as CH yet. Even though, Leach works efficiently, it has some disadvantages.

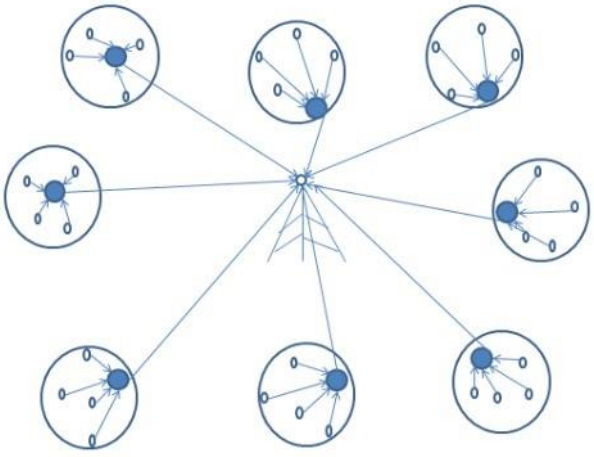

**Figure: 1.3**

## 1.3 K-Means Clustering

K-Means is an unsupervised learning algorithm, proposed by Mac Queen in 1967, initially was meant for data mining. It partitions the data set into certain number of clusters using the Euclidian distance mean that maximizes the intra-cluster similarities and minimizes inter-cluster similarities. Typically, it generates randomly k points (centre of the clusters), k being the expected number of clusters. The distance between each of the data points to each of the centres is calculated and assigned each point to the closest centre. The centre of the new cluster is

calculated by the mean value of all data points in the respective cluster. The distance between the data points is calculated using Euclidean distance defined by equation.

$$E = \sum_{i-1}^{k} \sum_{x \in ci} \left| x - \bar{x} \right| \qquad (1.2)$$

### 1.4 Clustering based on GA

Many alternative approaches were used earlier for clustering or to reduce energy consumption in wireless sensor network. This project deals with clustering the sensor nodes using Genetic Algorithm. This algorithm works to improve the lifetime of network by optimally consuming energy. Similarly others like an optimal traffic distribution technique is used to enhance the network life time of multiple sensored network, flat routing protocol is used to find optimal routes in WSN. This project shows a step by step working of genetic algorithm for clustering the nodes and handling the problems of WSN. The efficient of GA in WSN is proved by the simulation results at the end.

This works involves the use of genetic algorithm for clustering and choose a proper CH using fitness function.

### 1.5 Existing System

Enormous studies discuss that clustering has enough potential to reduce energy consumption and extends the network lifetime. The existing system of wireless sensor network uses leach protocol.

#### 1.5.1 Limitations of Existing System

Enormous studies show that clustering can reduce energy usage by the network components and helps in increasing network lifetime

### 1.6 Proposed system

Our objective is to perform clustering using genetic algorithm to reduce energy using of the network components. This done by selecting efficient cluster head using fitness function and selection operation.

# 2. SYSTEM REQUIREMENTS

## 2.1 Software Requirements

Supported Operating System:

- Windows 8(32 or 64 bit), Windows 10(32 or 64 bit)
- Supported Development Environment: MATLAB R2015a 8.5 version

## 2.2 Hardware Requirements:

- Processor: 1.2 GHZ
- RAM: 4 GB
- ROM: 1 TB

# 3. TECHNOLOGY

## 3.1 GENETIC ALGORITHM

The objective to perform clustering using genetic algorithm to reduce energy usage of(same as proposed system). The below diagram is the flow char of the genetic algorithm model. The three major steps that make GA most effective for problem solving in terms of system performance is (i) Creation of initial population, (ii) fitness function, (iii) GA parameters.

* **Encoding**: Encoding step involves finding the right choice of encoding from the pool of many coding schemes that are decimal coding, binary coding, octal coding, hexadecimal coding, tree coding and permutating so on. This choice of encoding is decided by the problem we are working with.

*** Creation of initial population:** The generation of initial population involves creating a chromosome of encoded string. The chromosome is a String of genes which contain all the genetic data of the individual.

*** Selection of fitness function:** The fitness function is evaluated on all individual in initial population to find the best individual to pass on its properties to next generation. This function brings variations.

* **Selection Operator:** The selection operation selects the best individuals based on fitness value of all the individuals. The common methods of selection operation are roulette wheel selection, elitism selection, tournament selection, steady state so on.

* **Crossover:** Crossover is the important step in the GA . The best individuals are mated by choosing a crossover point at random in the chromosomes. This is mainly swapping the strings of bits before and after crossover point of two individuals.

*** Mutation:** The probability of applying this step is very rare but this is an important step as it brings variation in the population. This step is done by flipping some of the bits in the string.

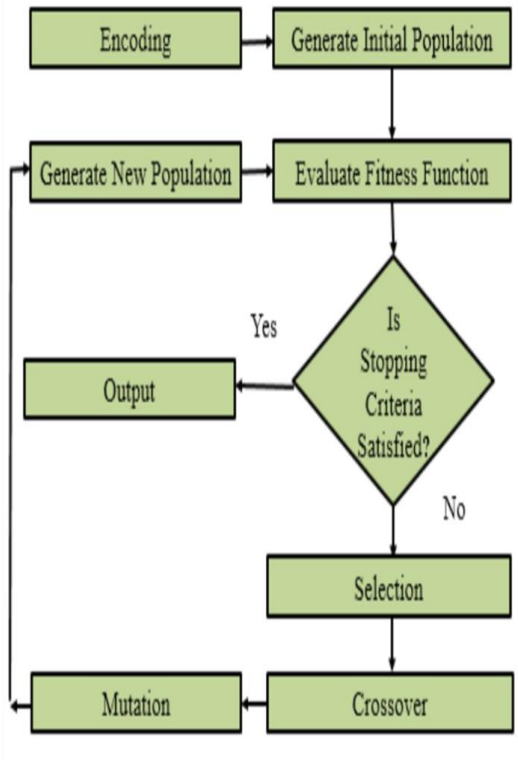


**Figure: 3.1**

### 3.1.1 General Description of GA

The work involves implementation genetic algorithm in WSN to choose a proper cluster head based depending on the fitness values of all sensor nodes. The node with highest value is selected as CH. The initial population is created by binary encoding. The length of chromosome created is 6. The fitness value is evaluated on every node using the formula given in equation 3.4. The node with highest fitness value is becomes CH. Now using a selection operation given in equation 3.5 we choose a node with highest probability as CH.

Now we perform crossover and mutation operation to create next generation. This process goes on until it arrives at initial population or the energy of all sensor nodes is depleted entirely.

### 3.1.2 Energy Model Analysis

The energy used for transmission of a single bits over a distance d from node(sender) to base station or another node(receiver)

Where,

- $E_{elec}$ is the energy consumed per bit for the working of transmitter or receiver unit of

node. The energy consumed depends on factors like filtering and spreading of signal,

digital coding etc.

- $\in_{fs}$and $\in_{mp}$are the sender amplifier energy constants where $\in_{fs}$ 0 is used free path and $\in_{mp}$is used for multi-path.

The free space path model ($d^2$power loss) is used when the distance between

the sender and receiver is less than the boundary value d 0 or else the multipath is used ($d^4$power loss). This loss can be prevented by using power amplifier. The energy used to receive

a signal bit of data signal and boundary value d 0 ratio is given as $\in_{fs}$and $\in_{mp}$:

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d) \quad = \begin{cases} l * E_{elec} + l * \varepsilon_{fs} * d^2 \ \ if\ d < d_0; \\ l * E_{elec} + l * \varepsilon_{mp} * d^4 \ if\ d \geq\ d_0; \end{cases} \qquad (3.1)$$

* $E_{elec}$ is the energy dissipated per bit to run the transmitter or the receiver circuit. It mostly depends on the parameters like digital coding, modulation, filtering and spreading of the signal.
*$\varepsilon_{fs}$&$\varepsilon_{mp}$are the transmitter amplifier characteristics, where $\in_{fs}$0 is used for free space and $\varepsilon_{mp}$used for multipath.

When the distance between transmitter and receiver is less than the threshold value$d_0$, the free space model ($d^2$power loss) is used. Otherwise, the multi-path fading channel model ($d^4$ power loss) is used. Power amplifier can be adjusted appropriately to invert this loss. The amount of energy consumption to receive l bit of data and the threshold value which is the ratio of $\in_{fs}$ &$\varepsilon_{mp}$is given.

$$E_{RX}(l) = E_{elec} * l \qquad (3.2)$$
$$d_0 = \sqrt{\epsilon_{fs}}/\varepsilon_{mp} \qquad (3.3)$$

### 3.1.3 The step by step GA operations incorporated in our work

**Encoding :** Our work involved creating initial population by encoding the nodes as binary strings of length of 6. The binary string is made up of energy, distance and number of neighbours.

For example, here we created a two-bit representation of each parameter to make 6-bit chromosome.

<Energy, Distance, No.of neighbours>

If suppose initial energy is 2J, then it is represented as 10

The distance calculated between node to the base station say 3 units is represents as 11

We considered the no of neighboures as 3 on an average, so it is represented as 11

So finally the chromosome is 101111

1. **Creation of Initial Population**: The binary encodes chromosomes of all the nodes the starating generation.

2. **Fitness function calculation:** In our work, we have calculated the fitness value of all the nodes in the population by using the following equation 3 4.

$$f_t(x) = D * \frac{d_{s-ch}}{d_{T(s-bs)} - d_{T(s-ch)}} \qquad (3.4)$$

Where,

$d_{s-ch}$ is the distance of each sensor node to the CH

$d_{T(s-bs)}$ is sum of distances of all the sensor nodes to the base station.

$d_{T(s-ch)}$ is the sum of distances of all the sensor nodes to cluster head.

For example, say node 1, fitness value is

$$f_{t(Node\ 1)} = 1.1 * \frac{2.5}{75-55.75} = 0.04$$

Where, $d_{s-ch}$= 2.5 cm

$d_{T(s-bs)}$ = 75cm

$d_{T(s-ch)}$ =55.75

In the same way, the fitness value of all other nodes is calculated and shown in Table 3.1

$$P_{Select} = \frac{f_{t(x)}}{\sum_0^5 f_t(x)} \qquad (3.5)$$

3. **Selection Operation:** Roulette wheel selection method works by calculating the percentage of a node has the chance to become cluster head among all the nodes using a probability density function given below as equation 3.5.
   From Table 3.2, we can say that Node 5 is having a greater chance to CH.

4. **Crossover Operation:** For example, we have taken 5 chromosomes randomly in cluster. Table 3.2 shows the initial population and fitness function value of the chromosomes using the equation given in 3.6. Table 3.3 shows the new population created after the crossover operation. The node with highest fitness values has high probability of the node to become CH.

5. **Mutation**: In our work, the mutation rate r is 0.001 which is very low. Mutation brings the variation to the population. Its done by flipping a bits from 1 to 0 or 0 to 1. Finally this process end if there are no alive nodes.

| Node ID | Cr NO | Initial Population | Decode value(D) in 1 Decimal point | $f_t$(x) |
|---|---|---|---|---|
| 1 | 1 | 0 0 1 0 1 1 | 1.1 | 0.04 |
| 2 | 2 | 0 1 0 0 1 0 | 1.8 | 0.09 |
| 3 | 3 | 0 0 1 1 0 0 | 1.2 | 0.08 |
| 4 | 4 | 0 0 1 1 1 1 | 1.5 | 0.12 |
| 5 | 5 | 1 1 0 1 0 1 | 5.3 | 0.47 |
| | | | | Sum=0.8 |

**Table: 3.1**

| Initial population | $p_{\text{select}}$ | Actual count | New Population |
|---|---|---|---|
| 0 0 1 0 1 1 | 0.05=5% | 0 | **Discard** |
| 0 1 0 0 1 0 | 0.11=11% | 1 | 0 1 0 0 1 0 |
| 0 0 1 1 0 0 | 0.1=10% | 0 | **Discard** |
| 0 0 1 1 1 1 | 0.15=15% | 1 | 0 0 1 1 1 1 |
| 1 1 0 1 0 1 | 0.58=58% | 3 | 1 1 0 1 0 1 |

**Table: 3.2**

| Initial Population | Crossover Point | New Population | Mutation |
|---|---|---|---|
| 0 0 1 0 11 | 0 1 0 0 1 0 | 0 1 1 1 1 1 | |
| 0 1 0 0 1 0 | 0 0 1 1 1 1 | 0 0 0 0 1 0 | |
| 0 0 1 1 0 0 | 1 1 0 1 0 1 | 1 1 0 1 0 1 | |
| 0 0 1 1 1 1 | 1 1 0 1 0 1 | 1 1 0 1 0 1 | |
| 1 1 0 1 0 1 | 1 1 0 1 0 1 | 1 1 0 1 1 1 | |

**Table: 3.3**

# 4. SYSTEM DESIGN

## 4.1 UML DIAGRAM

### 4.1.1 Class Diagram

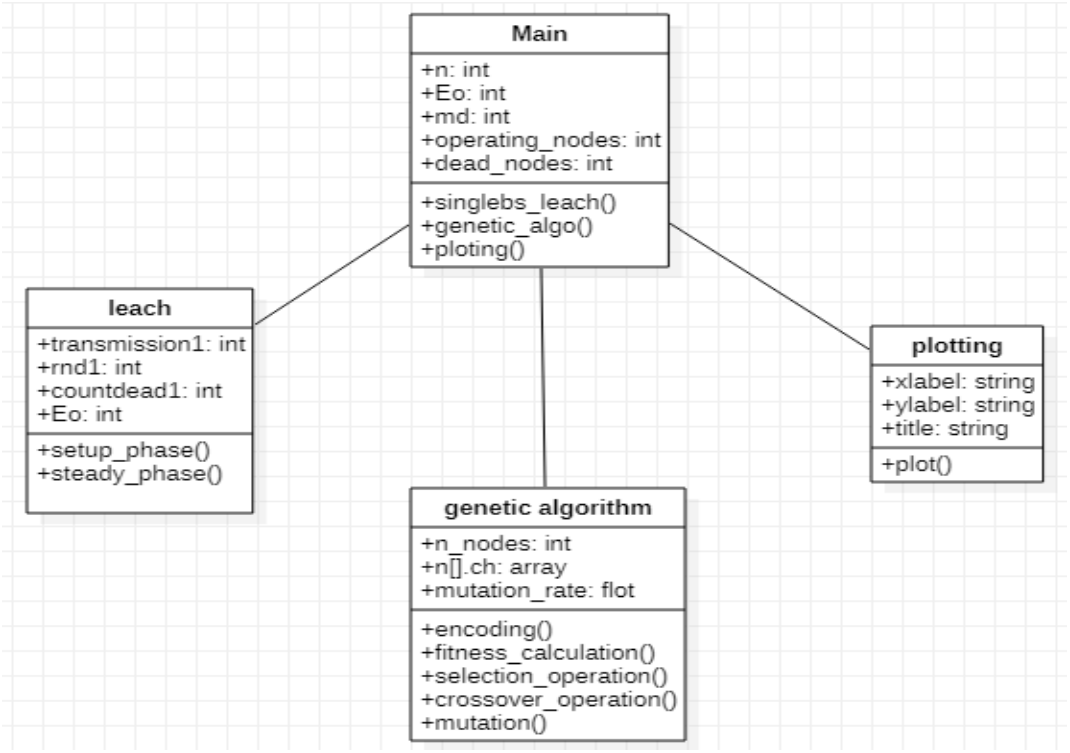


**Figure: 4.1**

### 4.1.2 Sequence Diagram

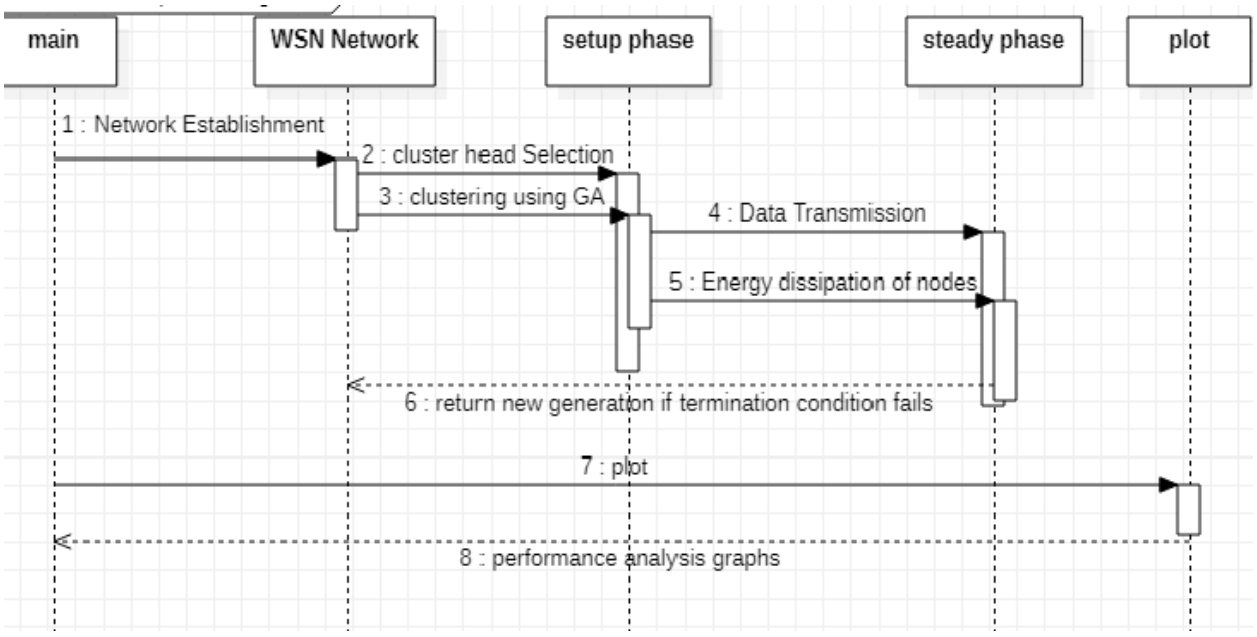


**Figure: 4.2**

### 4.1.3 Collaboration Diagram

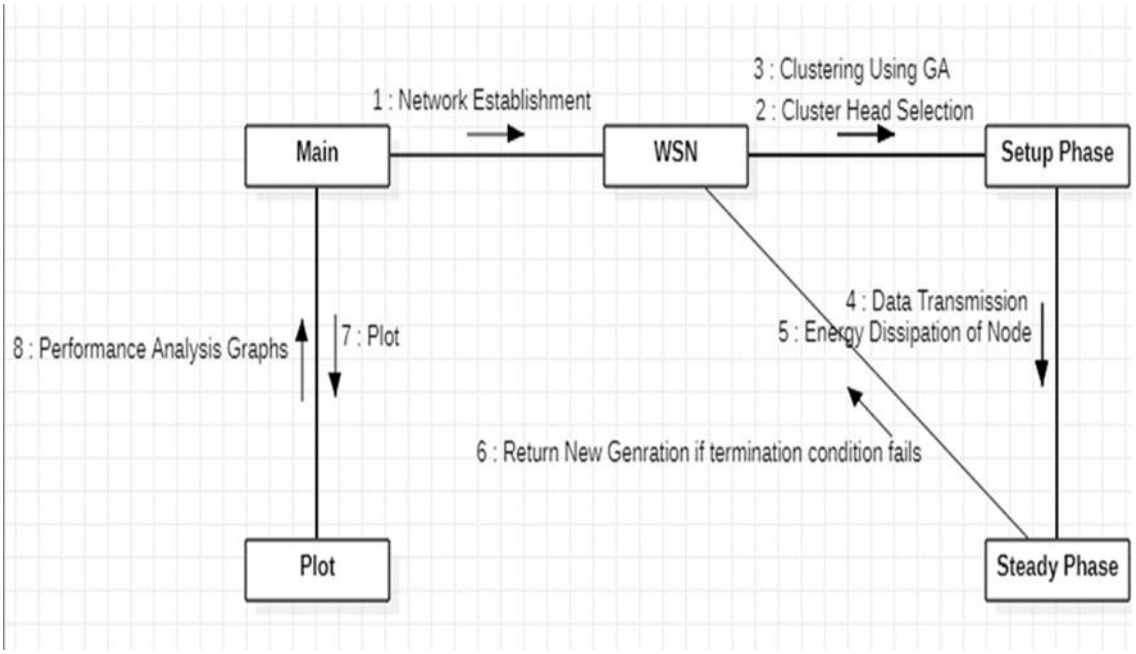


**Figure: 4.3**

### 4.1.4 Data Flow Diagram

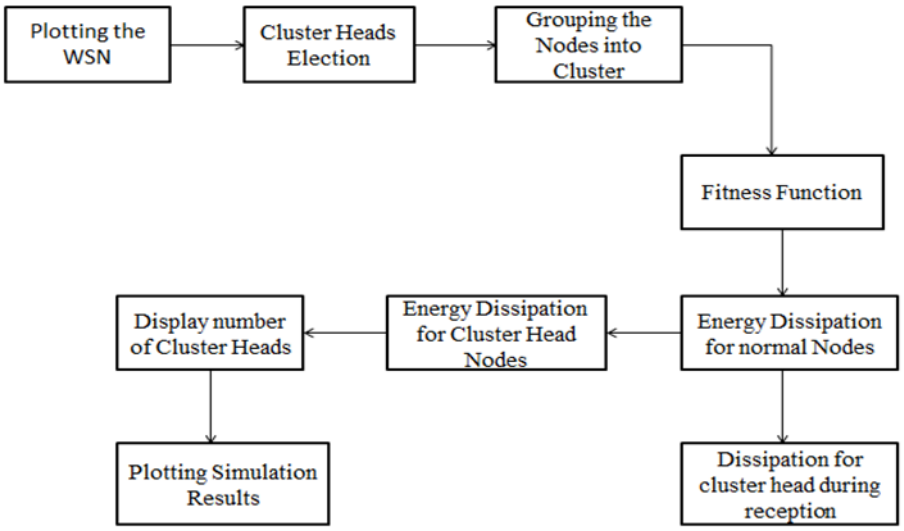


**Figure: 4.4**

### 4.1.5 Activity Diagram

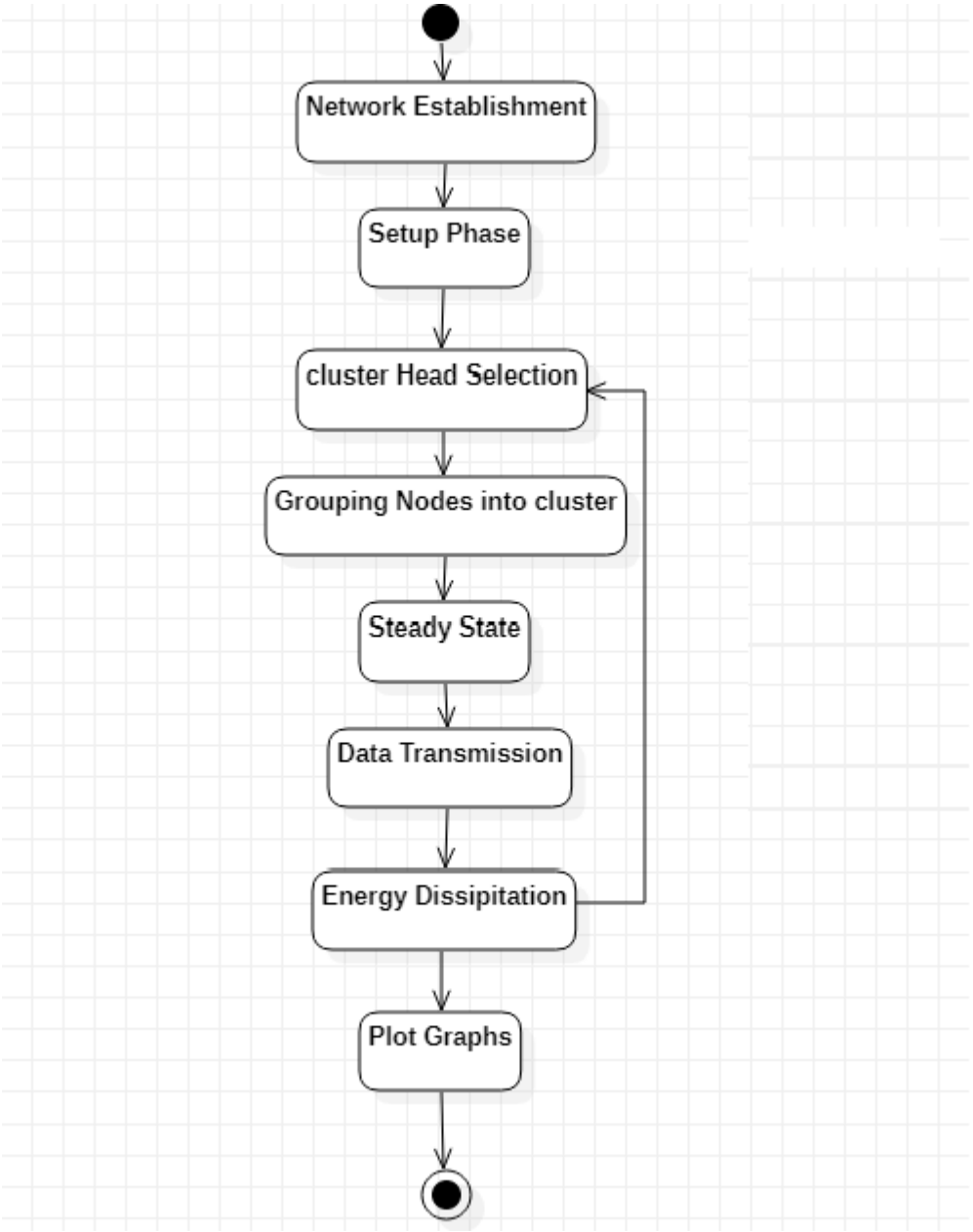


**Figure: 4.5**

### 4.1.6 State Chart Diagram

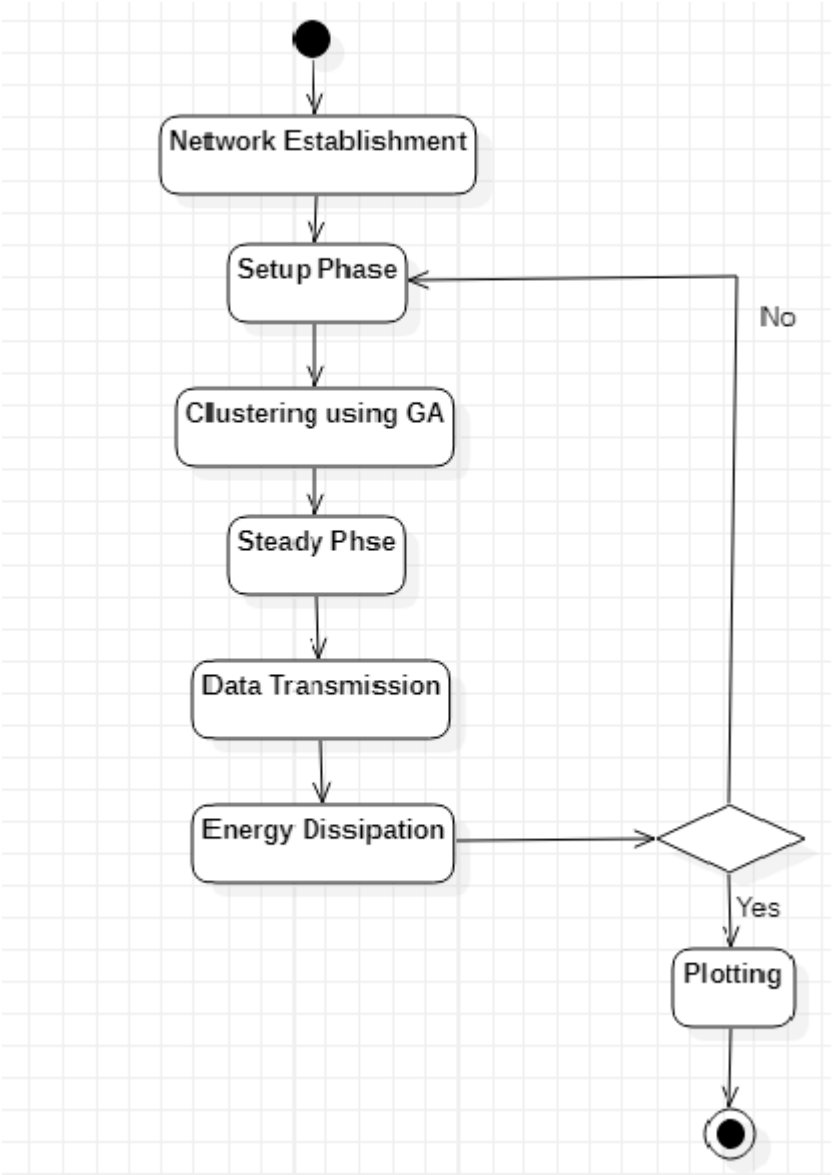


**Figure: 4.6**

### 4.1.7 Deployment Diagram

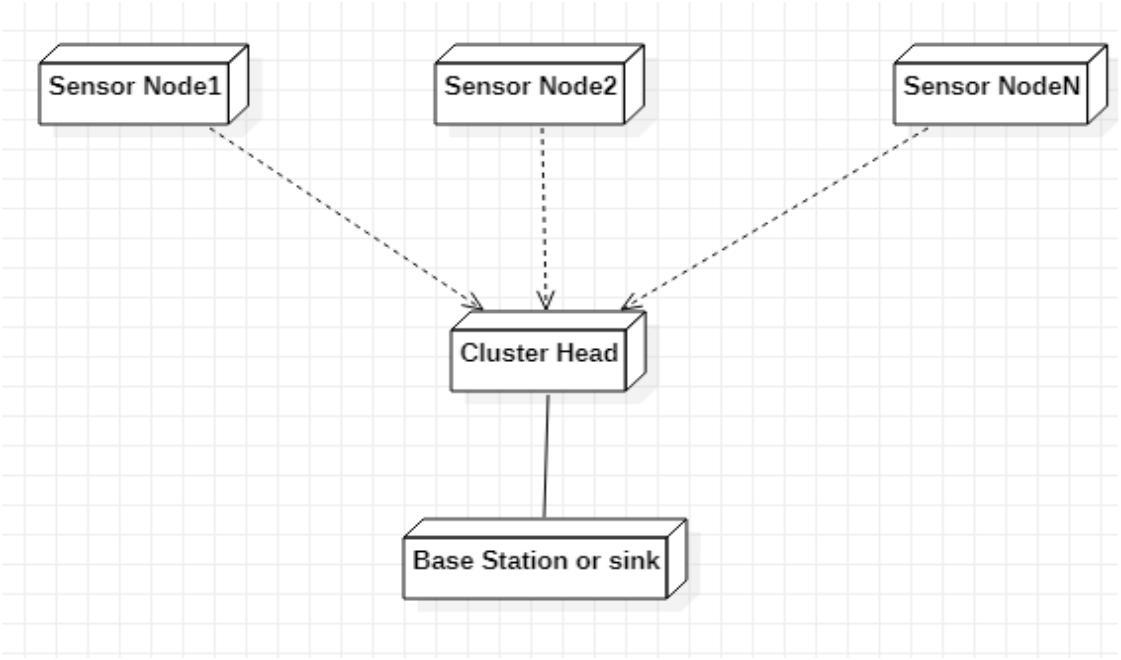


**Figure: 4.7**

# 5 IMPLEMENTATION

## 5.1 MODULES

### 5.1.1 Creation of Wireless Sensor Network

- Plotting of wireless sensor network includes plotting of sensor nodes and base station using rand() function.

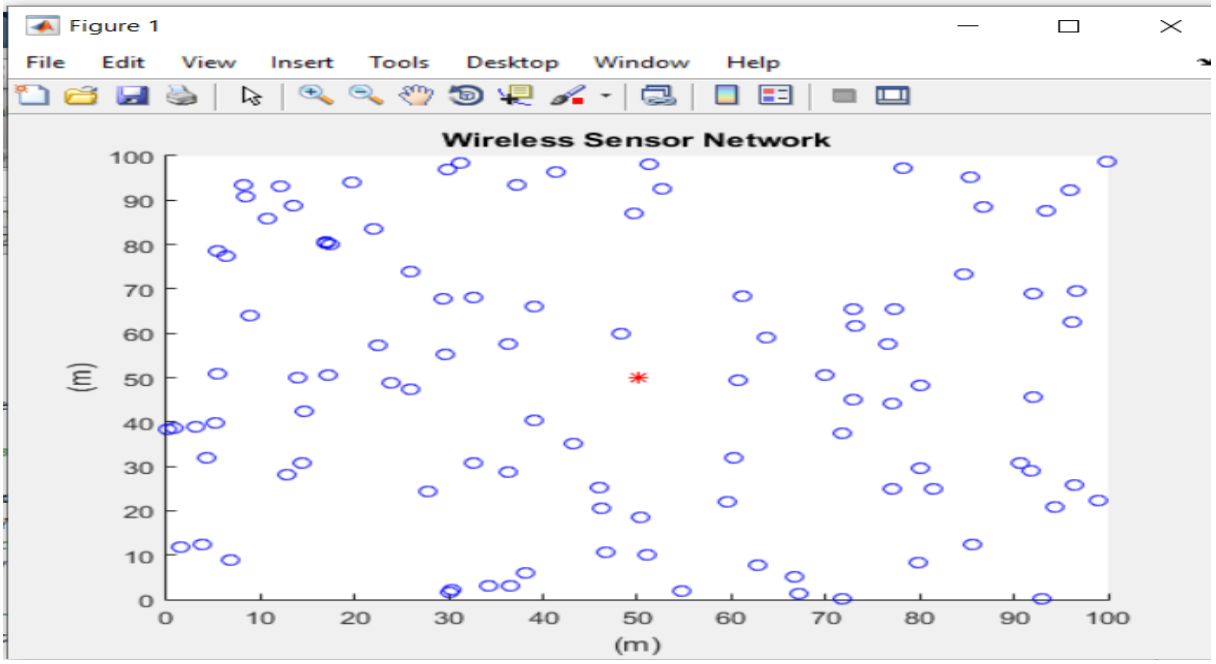


**Figure: 5.1**

### 5.1.2 Set-Up Phase

Cluster-heads are selected stochastically or randomly using this algorithm:

- $T(n)=\begin{cases} \frac{p}{1-p*(r \bmod \frac{1}{p})} & \\ 0 \quad otherwise \end{cases}$ (5.1)

This algorithm works by choosing each node as cluster head at least once.

**Algorithm for Setup phase**

- Every node that is selected as cluster head sends an advertisement message to other nodes.
- The transmission of advertisement message by cluster heads requires same transmit energy.
- During this transmission, the other nodes should turn on their receivers to listen to the advertisement message.
- The non-cluster head nodes now decide to cluster with cluster-head such that the node requires minimum communication energy during transmission.
- In our project, we used genetic algorithm for efficient clustering.

### 5.1.3 Algorithm for Steady phase

- After node picks cluster, it must inform cluster head
- Cluster heads will now know the number of nodes in its cluster.
- Nodes in the cluster will send the data according to the TDMA schedule created by its cluster head.
- When a node sends the data during its TDMA slot ,the other should turn off their transmitter thus reducing the energy consumption of individual nodes.
- Finally cluster head will now have all the data from the nodes in its cluster, and aggregates data.
- All the cluster heads will now sends the data to base station in CDMA fashion.

# 6 CODING

## 6.1 CLUSTERING USING GENETIC ALGORITHM

```
% GA
% pre defined values as per considered paper
ip = [001011,010010,001100,001111,110101];
d  = [1.1,1.8,1.2,1.5,5.3];
x = [0.036,0.050,0.067,0.080,0.087];
r = 0.001;


% intial population creation
n_nodes = 5;     % cosidered 5 nodes
for i = 1:n_nodes


    % encoding
    n(i).e = dec2bin(randperm(4,1)-1,2);     % node energy parameter
    n(i).d = dec2bin(randperm(4,1)-1,2);     % node distance parameter
    n(i).n = dec2bin(randperm(4,1)-1,2);     % node no. of neighbours paramete


    % chromosomes creation for initial population
    n(i).ch = append(n(i).e,n(i).d,n(i).n);
end


% fitness function
for i = 1:n_nodes
    f(i) = d(i)*x(i);
end

% probability density function
sum_f = sum(f);
for i = 1:n_nodes
    pdf_n(i) = f(i)/sum_f;
end
```

```
% selection operator
for i = 1:n_nodes
    p_pdf_n(i) = pdf_n(i)*100;  % node 5 has highest pdf
    if p_pdf_n(i) <= 10
        ip(i) = 0;
    else
        ip(i) = ip(i);
    end
end


% crossover operation
for i = 1:n_nodes
    Ip = [ip(2),ip(4),ip(5),ip(5),ip(5)];
end


% mutation


m_p = r*(2^6);
Ip1 = Ip*m_p;


Np = [011111,000010,ip(5),ip(5),110111];  % as node 5 is cluster head

%%


% Analysis and graphs


rounds_no = 2:2:20;     % number of rounds
pdr_values =  [42 33; 33 27; 29 25; 26 30; 25 31; 25 36; 24 36; 23 37; 22 39; 21 41];
figure(3)
bar(rounds_no,pdr_values)
xlabel('Number of Rounds')
ylabel('Packet delivery Ratio %')
title('Packet delivery ratio vs number of Rounds')
```

## 6.2 CLUSTERING USING LEACH FOR MULTIPLE BASE STATIONS

In this project, we also worked on LEACH with multiple stations to enhance the network lifetime. In this multi-model network the CH will send the data to the nearest BS, this reducing the overall energy consumption

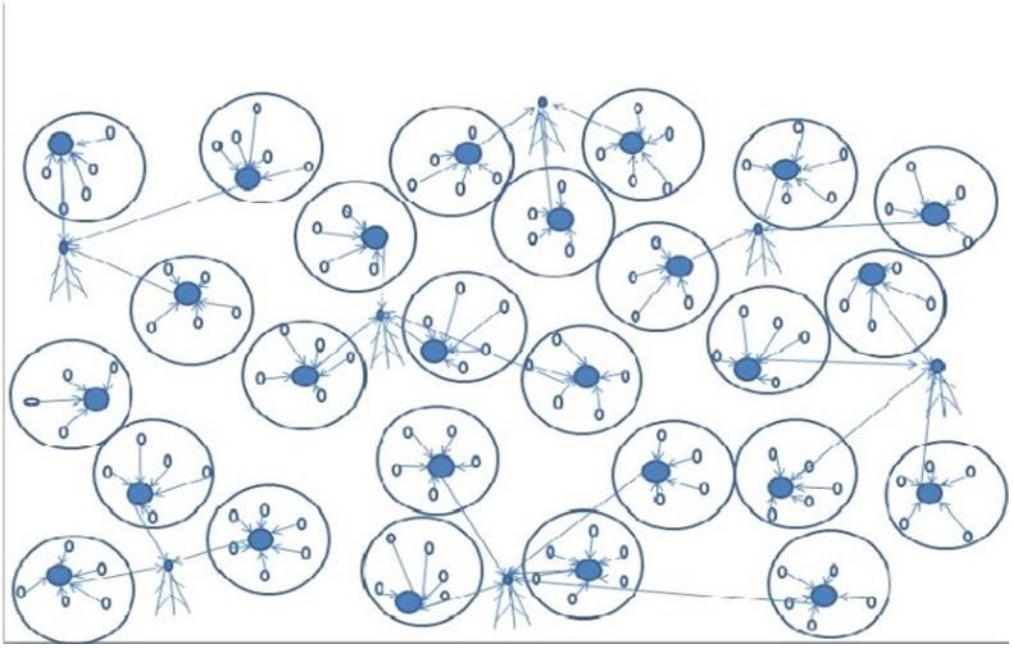

**Figure: 6.1**

main.m

```
close all;
clear;
clc;

%%%%%%%%%%%%%%%%%%%% Network Establishment Parameters %%%%%%%%%%%%%%%%%%%%

%%% Area of Operation %%%

% Field Dimensions in meters %
xm=100;
ym=100;
x=0; % added for better display results of the plot
y=0; % added for better display results of the plot
% Number of Nodes in the field %
n=100;
% Number of Dead Nodes in the beggining %
dead_nodes=0;
% Coordinates of the Sink (location is predetermined in this simulation) %
sinkx1=50;
sinky1=50;
sinkx2=[25 75 75 25];
sinky2=[25 25 75 75];
%%% Energy Values %%%
% Initial Energy of a Node (in Joules) %
Eo=2; % units in Joules
% Energy required to run circuity (both for transmitter and receiver) %
Eelec=50*10^(-9); % units in Joules/bit
ETx=50*10^(-9); % units in Joules/bit
ERx=50*10^(-9); % units in Joules/bit
% Transmit Amplifier Types %
Eamp=100*10^(-12); % units in Joules/bit/m^2 (amount of energy spent by the amplifier to transmit the bits)
% Data Aggregation Energy %
EDA=5*10^(-9); % units in Joules/bit
% Size of data package %
k=4000; % units in bits
% Suggested percentage of cluster head %
p=0.05; % a 5 percent of the total amount of nodes used in the network is proposed to give good results
% Number of Clusters %
No=p*n;
% Round of Operation %
rnd1=0;
rnd2=0;
% Current Number of operating Nodes %
operating_nodes=n;
transmissions1=0;
transmissions2=0;
temp_val=0;
flag1stdead=0;
flag2stdead=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%% End of Parameters %%%%%%%%%%%%%%%%%%%%%%%%%%%%


            %%% Creation of the Wireless Sensor Network %%%

% Plotting the WSN %
for i=1:n

    SN(i).id=i; % sensor's ID number
    SN(i).x=rand(1,1)*xm;   % X-axis coordinates of sensor node
    SN(i).y=rand(1,1)*ym;   % Y-axis coordinates of sensor node
```

```
SN(i).E=Eo;      % nodes energy levels (initially set to be equal to "Eo"
SN(i).role=0;    % node acts as normal if the value is '0', if elected as a cluster head it  gets the value '1' (initially all nodes are normal)
SN(i).cluster=0;     % the cluster which a node belongs to
SN(i).cond=1;    % States the current condition of the node. when the node is operational its value is =1 and when dead =0
SN(i).rop=0;     % number of rounds node was operational
SN(i).rleft=0;   % rounds left for node to become available for Cluster Head election
SN(i).dtch=0;    % nodes distance from the cluster head of the cluster in which he belongs
SN(i).dts=0;     % nodes distance from the sink
SN(i).tel=0;     % states how many times the node was elected as a Cluster Head
SN(i).rn=0;      % round node got elected as cluster head
SN(i).chid=0;    % node ID of the cluster head which the "i" normal node belongs to


hold on;
figure(1)
plot(x,y,xm,ym,SN(i).x,SN(i).y,'ob',sinkx1,sinky1,'*r');
title 'Wireless Sensor Network';
xlabel '(m)';
ylabel '(m)';

 hold on;
    figure(2)
    plot(x,y,xm,ym,SN(i).x,SN(i).y,'ob',sinkx2,sinky2,'*r');
    title 'Wireless Sensor Network';
    xlabel '(m)';
    ylabel '(m)';

end
[transmissions1,rnd1,op1,tr1,flag1stdead,nrg1,avg_node1,count1,countdead1]=singlebs_leach(SN,xm,ym,x,y,n,dead_nodes,sinkx1,sinky1,Eo,Eelec,ETx,ERx,Eamp,EDA,k,p,No,rnd1,trans
[transmissions2,rnd2,op2,tr2,flag2stdead,nrg2,avg_node2,count2,countdead2]=multibs_leach(SN,xm,ym,x,y,n,dead_nodes,sinkx2,sinky2,Eo,Eelec,ETx,ERx,Eamp,EDA,k,p,No,rnd2,transr

% Plotting Simulation Results "Operating Nodes per Round" %
    figure(3)
    plot(op1(1:rnd1),1:rnd1,'-r',op2(1:rnd2),1:rnd2,'-b','LineWidth',4);
    title ({'LEACH'; 'Operating Nodes per Round';})
    xlabel 'Operational Nodes';
    ylabel 'Rounds';
    hold on;

    % Plotting Simulation Results  %
    figure(4)
    plot(tr1(1:transmissions1),1:transmissions1,'-r',tr2(1:transmissions2),1:transmissions2,'-b','LineWidth',4);
    title ({'LEACH'; 'Operational Nodes per Transmission';})
    xlabel 'Operational Nodes';
    ylabel 'Transmissions';
    hold on;

% Plotting Simulation Results  %
figure(5)
plot(1:flag1stdead,nrg1(1:flag1stdead),'-r',1:flag2stdead,nrg2(1:flag2stdead),'-b','LineWidth',4);
title ({'LEACH'; 'Energy consumed per Transmission';})
xlabel 'Transmission';
ylabel 'Energy ( J )';
hold on;


% Plotting Simulation Results  %
figure(6)
plot(1:flag1stdead,avg_node1(1:flag1stdead),'-r',1:flag2stdead,avg_node2(1:flag2stdead),'-b','LineWidth',4);
title ({'LEACH'; 'Average Energy consumed by a Node per Transmission';})
xlabel 'Transmissions';
ylabel 'Energy(J)';
hold on;


% No of dead nodes per round %
figure(7)
plot(countdead1(1:rnd1),1:rnd1,'-r',countdead2(1:rnd2),1:rnd2,'-b','LineWidth',4);
title ({'No of dead nodes per round';})
xlabel 'Dead Nodes';
ylabel 'Rounds';
hold on;
```

## Singlebs_leach.m

```
function [transmissions1,rnd1,op1,tr1,flag1stdead,nrg1,avg_node1,count1,countdead1]=singlebs_leach(SN,xm,ym,x,y,n,dead_nodes,sinkx1,sinky1,Eo,Eelec,ETx,ERx,Eamp,EDA,k,p,No,r

%%%%%% Set-Up Phase %%%%%%
operating_nodes=n;

while operating_nodes>0

    % Displays Current Round %
    rnd1

    % Threshold Value %
    t=(p/(1-p*(mod(rnd1,1/p))));

    % Re-election Value %
    tleft=mod(rnd1,1/p);

    % Reseting Previous Amount Of Cluster Heads In the Network %
    CLheads=0;

    % Reseting Previous Amount Of Energy Consumed In the Network on the Previous Round %
    energy=0;


 Cluster Heads Election %

       for i=1:n
           SN(i).cluster=0;    % reseting cluster in which the node belongs to
           SN(i).role=0;       % reseting node role
           SN(i).chid=0;       % reseting cluster head id
           if SN(i).rleft>0
              SN(i).rleft=SN(i).rleft-1;
           end

           if (SN(i).E>0) && (SN(i).rleft==0)
               generate=rand;
                   if generate< t
                   SN(i).role=1;   % assigns the node role of acluster head
                   SN(i).rn=rnd1;  % Assigns the round that the cluster head was elected to the data table
                   SN(i).tel=SN(i).tel + 1;
                   SN(i).rleft=1/p-tleft;    % rounds for which the node will be unable to become a CH
                   SN(i).dts=sqrt((sinkx1-SN(i).x)^2 + (sinky1-SN(i).y)^2); % calculates the distance between the sink and the cluster hea
                   CLheads=CLheads+1;  % sum of cluster heads that have been elected
                   SN(i).cluster=CLheads; % cluster of which the node got elected to be cluster head
                   CL(CLheads).x=SN(i).x; % X-axis coordinates of elected cluster head
                   CL(CLheads).y=SN(i).y; % Y-axis coordinates of elected cluster head
                   CL(CLheads).id=i; % Assigns the node ID of the newly elected cluster head to an array
                   end

           end
       end

   % Fixing the size of "CL" array %
   CL=CL(1:CLheads);
% Grouping the Nodes into Clusters & caclulating the distance between node and cluster head %

       for i=1:n
        if  (SN(i).role==0) && (SN(i).E>0) && (CLheads>0) % if node is normal
            for m=1:CLheads
            d(m)=sqrt((CL(m).x-SN(i).x)^2 + (CL(m).y-SN(i).y)^2);
            % we calculate the distance 'd' between the sensor node that is
            % transmitting and the cluster head that is receiving with the following equation+
            % d=sqrt((x2-x1)^2 + (y2-y1)^2) where x2 and y2 the coordinates of
            % the cluster head and x1 and y1 the coordinates of the transmitting node
            end
        d=d(1:CLheads); % fixing the size of "d" array
        [M,I]=min(d(:)); % finds the minimum distance of node to CH
        [Row, Col] = ind2sub(size(d),I); % displays the Cluster Number in which this node belongs too
        SN(i).cluster=Col; % assigns node to the cluster
        SN(i).dtch= d(Col); % assigns the distance of node to CH
        SN(i).chid=CL(Col).id;
        end
       end
```

```
                        %%%%%% Steady-State Phase %%%%%%

% Energy Dissipation for normal nodes %

    for i=1:n
       if (SN(i).cond==1) && (SN(i).role==0) && (CLheads>0)
        if SN(i).E>0
            ETx= Eelec*k + Eamp * k * SN(i).dtch^2;
            SN(i).E=SN(i).E - ETx;
            energy=energy+ETx;

        % Dissipation for cluster head during reception
        if SN(SN(i).chid).E>0 && SN(SN(i).chid).cond==1 && SN(SN(i).chid).role==1
            ERx=(Eelec+EDA)*k;
            energy=energy+ERx;
            SN(SN(i).chid).E=SN(SN(i).chid).E - ERx;
             if SN(SN(i).chid).E<=0  % if cluster heads energy depletes with reception
                SN(SN(i).chid).cond=0;
                SN(SN(i).chid).rop=rnd1;
                dead_nodes=dead_nodes +1;
                operating_nodes=operating_nodes - 1
             end
        end
        end


        if SN(i).E<=0       % if nodes energy depletes with transmission
        dead_nodes=dead_nodes +1;
        operating_nodes= operating_nodes - 1
       SN(i).cond=0;
       SN(i).chid=0;
       SN(i).rop=rnd1;
       end

     end
  end



Energy Dissipation for cluster head nodes %

 for i=1:n
   if (SN(i).cond==1)  && (SN(i).role==1)
        if SN(i).E>0
           ETx= (Eelec+EDA)*k + Eamp * k * SN(i).dts^2;
           SN(i).E=SN(i).E - ETx;
           energy=energy+ETx;
        end
        if  SN(i).E<=0     % if cluster heads energy depletes with transmission
        dead_nodes=dead_nodes +1;
        operating_nodes= operating_nodes - 1
        SN(i).cond=0;
        SN(i).rop=rnd1;
        end
   end
 end
```

```
    if operating_nodes<n && temp_val==0
        temp_val=1;
        flaglstdead=rnd1;
    end
    % Display Number of Cluster Heads of this round %
    %CLheads;


    transmissions1=transmissions1+1;
    if CLheads==0
    transmissions1=transmissions1-1;
    end


    % Next Round %
    rnd1= rnd1 +1;

    tr1(transmissions1)=operating_nodes;
    op1(rnd1)=operating_nodes;
    count1(rnd1)=transmissions1;
    countdead1(rnd1)=dead_nodes;


    if energy>0
    nrg1(transmissions1)=energy;
    end



end
    % Next Round %
    rnd1= rnd1 +1;

    tr1(transmissions1)=operating_nodes;
    op1(rnd1)=operating_nodes;
    count1(rnd1)=transmissions1;
    countdead1(rnd1)=dead_nodes;


    if energy>0
    nrg1(transmissions1)=energy;
    end


end


sum=0;
for i=1:flaglstdead
    sum=nrg1(i) + sum;
end

temp1=sum/flaglstdead;
temp2=temp1/n;

for i=1:flaglstdead
avg_node1(i)=temp2;
end
```

## multibs_leach.m

```
function [transmissions2,rnd2,op2,tr2,flag2stdead,nrg2,avg_node2,count2,countdead2]=miltibs_leach(SN,xm,ym,x,y,n,dead_nodes,sinkx2,sinky2,Eo,Eelec,ETx,ERx,Eamp,EDA,k,p,No,r

    %%%%%% Set-Up Phase %%%%%%

operating_nodes=n;

while operating_nodes>0

    % Displays Current Round %
    rnd2

    % Threshold Value %

    t=(p/(1-p*(mod(rnd2,1/p))));

    % Re-election Value %
    tleft=mod(rnd2,1/p);

    % Reseting Previous Amount Of Cluster Heads In the Network %
    CLheads=0;

    % Reseting Previous Amount Of Energy Consumed In the Network on the Previous Round %
    energy=0;


% Cluster Heads Election %

        for i=1:n
            SN(i).cluster=0;    % reseting cluster in which the node belongs to
   SN(i).role=0;        % reseting node role
   SN(i).chid=0;        % reseting cluster head id
   if SN(i).rleft>0
      SN(i).rleft=SN(i).rleft-1;
   end
   if (SN(i).E>0) && (SN(i).rleft==0)
       generate=rand;
           if generate< t
           SN(i).role=1;   % assigns the node role of acluster head
           SN(i).rn=rnd2;  % Assigns the round that the cluster head was elected to the data table
           SN(i).tel=SN(i).tel + 1;
           SN(i).rleft=1/p-tleft;    % rounds for which the node will be unable to become a CH
   min_dist=100;
   for j=1:1:4
   temp=sqrt((sinkx2(j)-SN(i).x)^2 + (sinky2(j)-SN(i).y)^2); % calculates the distance between the sink and the cluster head
   if(temp<min_dist)
       min_dist=temp;
   end
   end
   SN(i).dts=min_dist;
           CLheads=CLheads+1;  % sum of cluster heads that have been elected
           SN(i).cluster=CLheads; % cluster of which the node got elected to be cluster head
           CL(CLheads).x=SN(i).x; % X-axis coordinates of elected cluster head
           CL(CLheads).y=SN(i).y; % Y-axis coordinates of elected cluster head
           CL(CLheads).id=i; % Assigns the node ID of the newly elected cluster head to an array
           end
        end

   % Fixing the size of "CL" array %
   CL=CL(1:CLheads);



 Grouping the Nodes into Clusters & caclulating the distance between node and cluster head %

      for i=1:n
       if  (SN(i).role==0) && (SN(i).E>0) && (CLheads>0) % if node is normal
           for m=1:CLheads
           d(m)=sqrt((CL(m).x-SN(i).x)^2 + (CL(m).y-SN(i).y)^2);
           % we calculate the distance 'd' between the sensor node that is
           % transmitting and the cluster head that is receiving with the following equation+
           % d=sqrt((x2-x1)^2 + (y2-y1)^2) where x2 and y2 the coordinates of
           % the cluster head and x1 and y1 the coordinates of the transmitting node
           end
       d=d(1:CLheads); % fixing the size of "d" array
       [M,I]=min(d(:)); % finds the minimum distance of node to CH
       [Row, Col] = ind2sub(size(d),I); % displays the Cluster Number in which this node belongs too
       SN(i).cluster=Col; % assigns node to the cluster
       SN(i).dtch= d(Col); % assigns the distance of node to CH
       SN(i).chid=CL(Col).id;
       end
      end
```

```
                        %%%%%% Steady-State Phase %%%%%%

% Energy Dissipation for normal nodes %

    for i=1:n
       if (SN(i).cond==1) && (SN(i).role==0) && (CLheads>0)
        if SN(i).E>0
            ETx= Eelec*k + Eamp * k * SN(i).dtch^2;
            SN(i).E=SN(i).E - ETx;
            energy=energy+ETx;

        % Dissipation for cluster head during reception
        if SN(SN(i).chid).E>0 && SN(SN(i).chid).cond==1 && SN(SN(i).chid).role==1
            ERx=(Eelec+EDA)*k;
            energy=energy+ERx;
            SN(SN(i).chid).E=SN(SN(i).chid).E - ERx;
             if SN(SN(i).chid).E<=0  % if cluster heads energy depletes with reception
                SN(SN(i).chid).cond=0;
                SN(SN(i).chid).rop=rnd2;
                dead_nodes=dead_nodes +1;
                operating_nodes= operating_nodes - 1
             end
        end
        end


        if SN(i).E<=0       % if nodes energy depletes with transmission
        dead_nodes=dead_nodes +1;
        operating_nodes= operating_nodes - 1
        SN(i).cond=0;
for i=1:n
  if (SN(i).cond==1)  && (SN(i).role==1)
      if SN(i).E>0
         ETx= (Eelec+EDA)*k + Eamp * k * SN(i).dts^2;
         SN(i).E=SN(i).E - ETx;
         energy=energy+ETx;
      end
      if  SN(i).E<=0     % if cluster heads energy depletes with transmission
      dead_nodes=dead_nodes +1;
      operating_nodes= operating_nodes - 1
      SN(i).cond=0;
      SN(i).rop=rnd2;
      end
  end
end



 if operating_nodes<n && temp_val==0
     temp_val=1;
     flag2stdead=rnd2
 end
 % Display Number of Cluster Heads of this round %
 %CLheads;


 transmissions2=transmissions2+1;
 if CLheads==0
 transmissions2=transmissions2-1;
 end
```

```
    % Next Round %
    rnd2= rnd2 +1;

    tr2(transmissions2)=operating_nodes;
    op2(rnd2)=operating_nodes;
    count2(rnd2)=transmissions2;
    countdead2(rnd2)=dead_nodes;
    if energy>0
    nrg2(transmissions2)=energy;
    end


end

sum=0;
for i=1:flag2stdead
    sum=nrg2(i) + sum;
end

temp1=sum/flag2stdead;
temp2=temp1/n;

for i=1:flag2stdead
avg_node2(i)=temp2;
end
```

## 6.3 Results

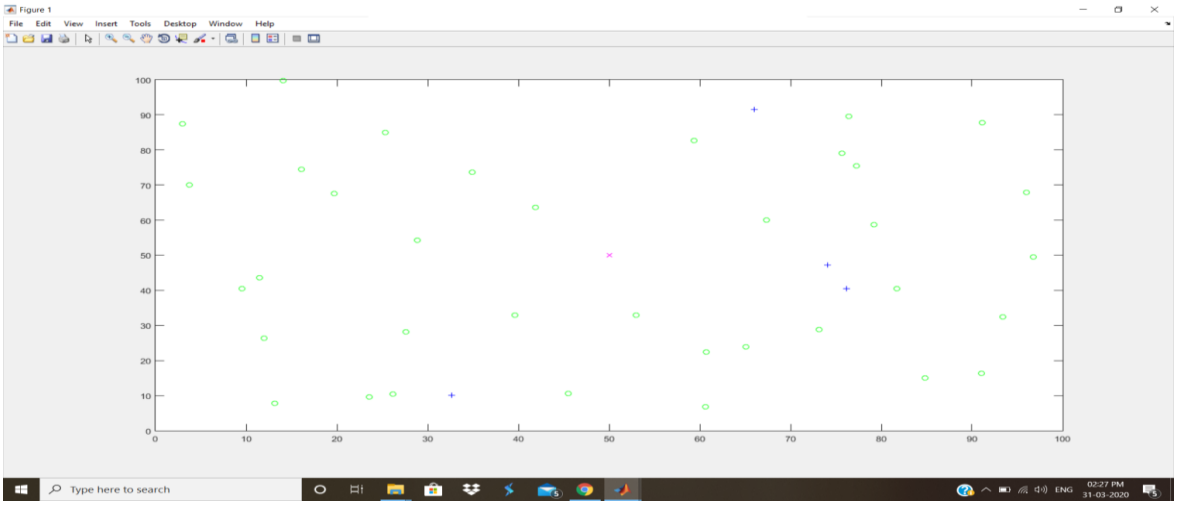


**Photo: 6.1**

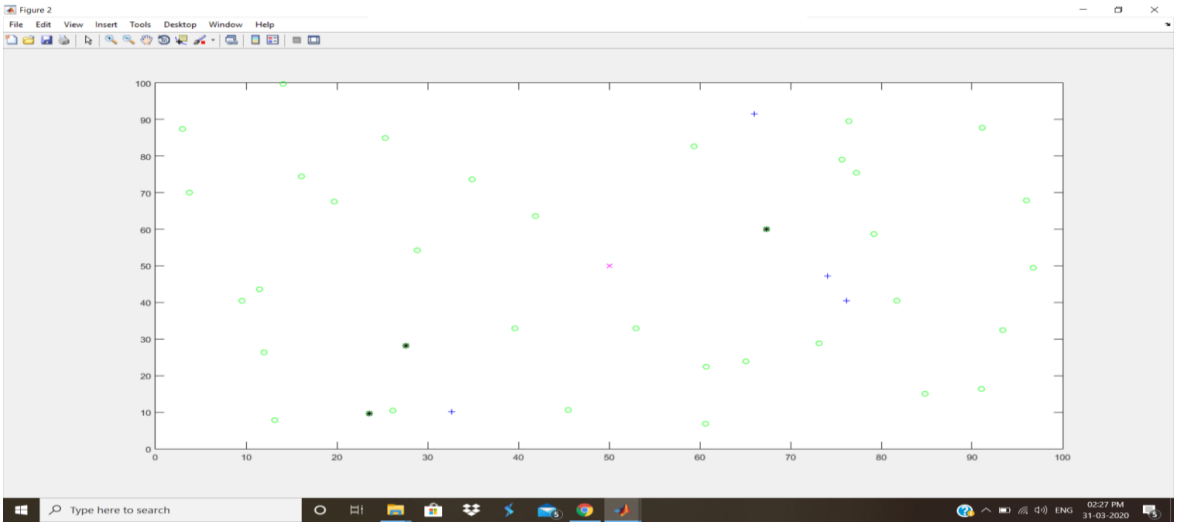


**Photo: 6.2**

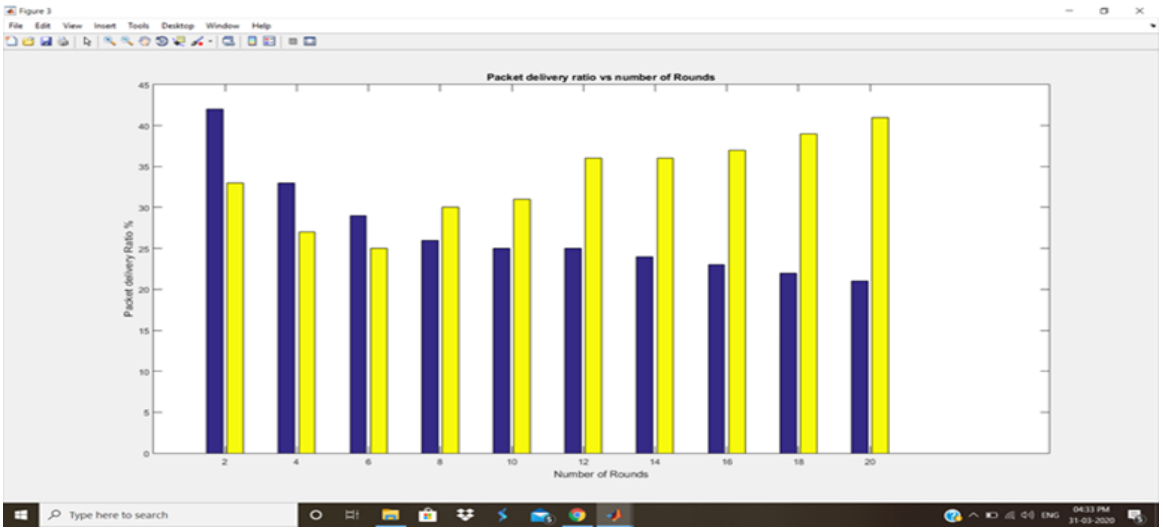


**Photo: 6.3**

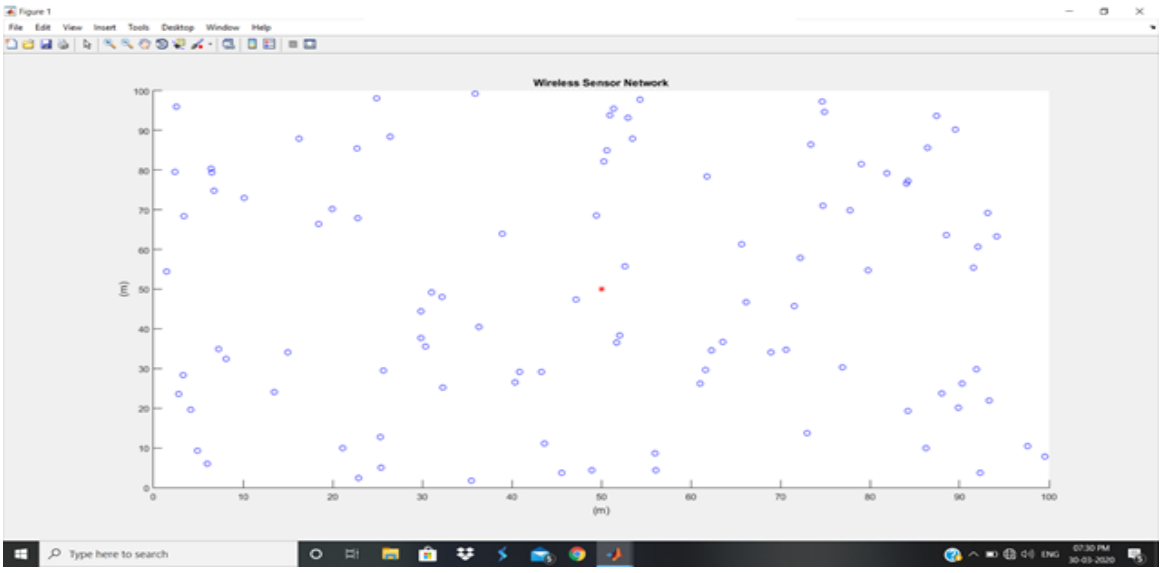


**Photo: 6.4**

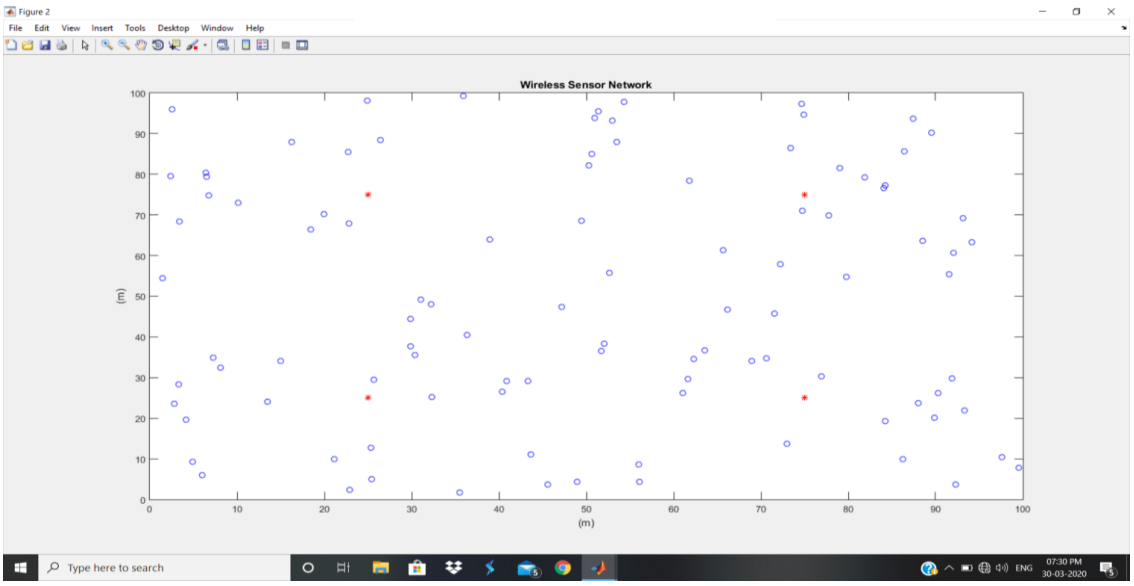


**Photo 6.5**

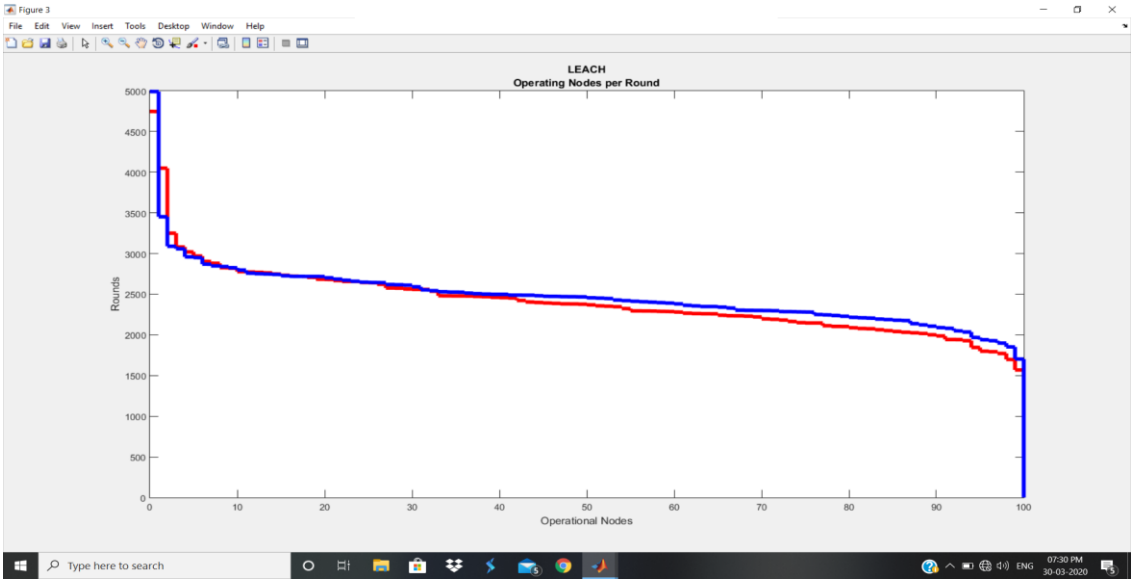


**Photo 6.6**

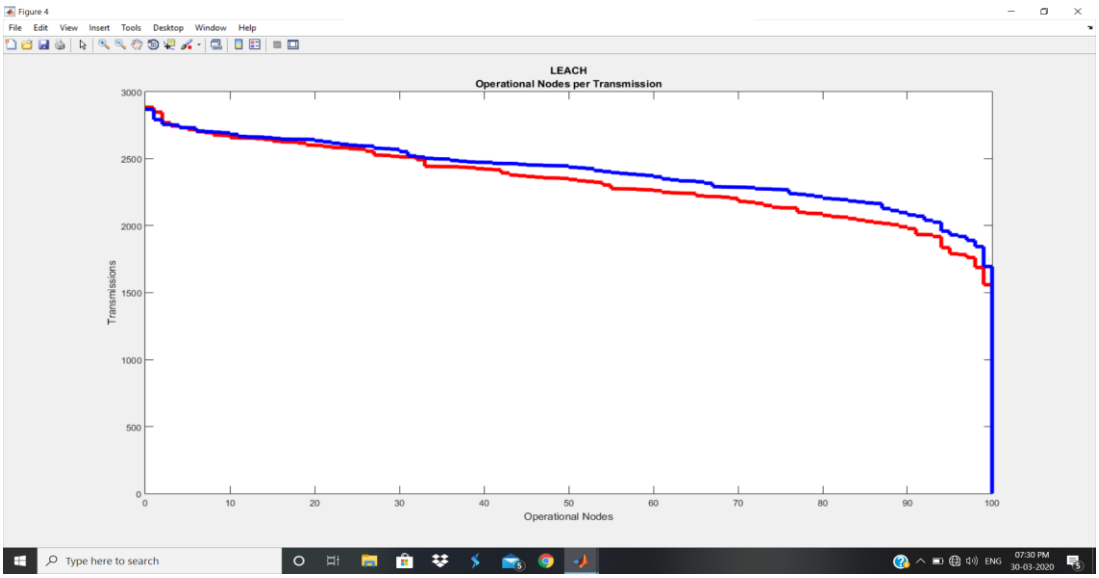


**Photo: 6.7**

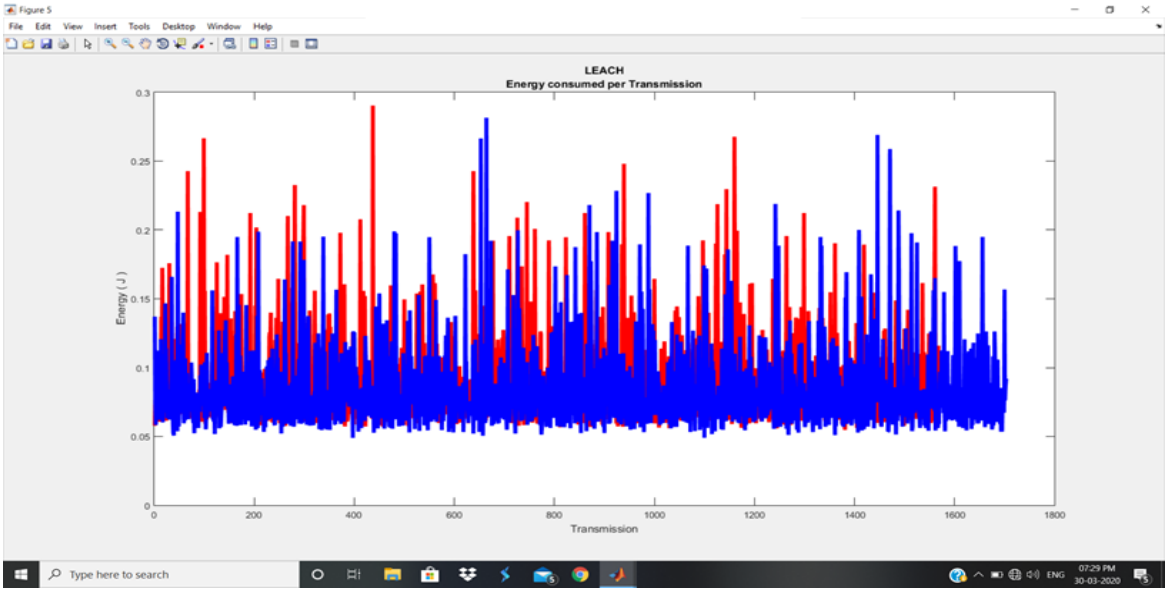


**Photo: 6.8**

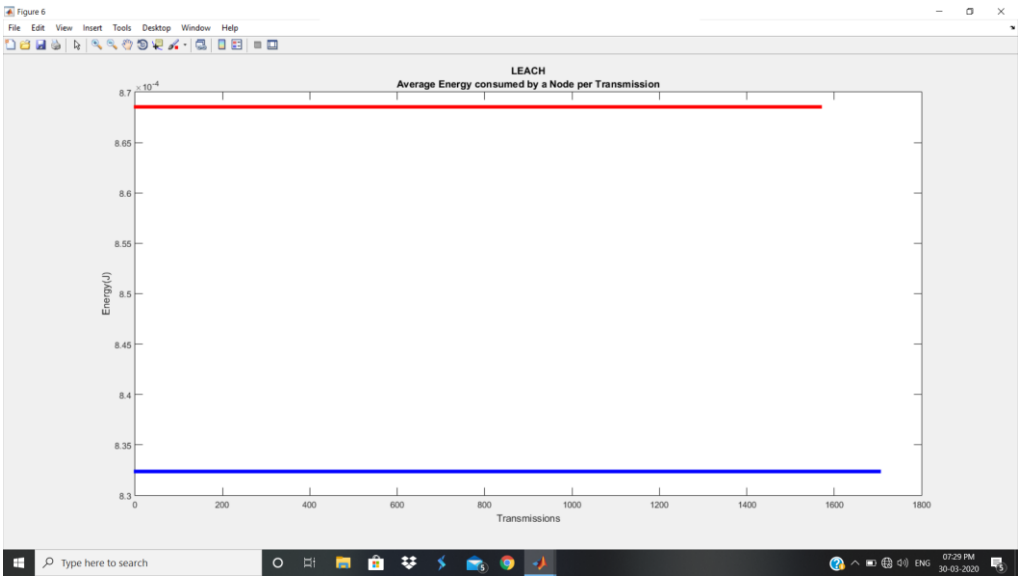


**Photo: 6.9**

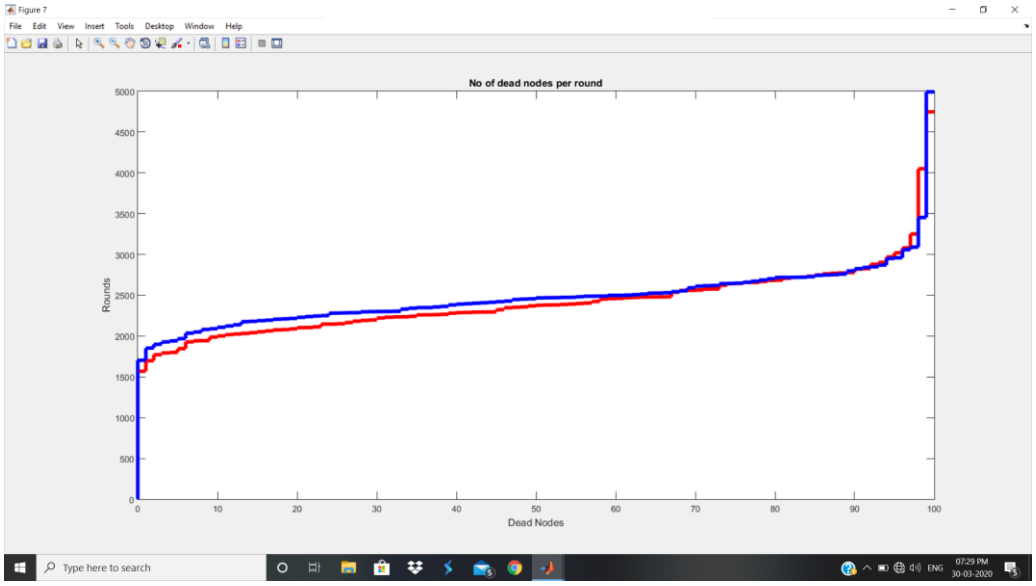


**Photo: 6.10**

# 7. CONCLUSION

- The aim of the project is to improve the network lifetime by maximizing the data delivery of data signals to BS. The network lifetime is directly proportional to the energy consumption. We know that energy consumption is less in LEACH but to further improve the performance we infused bio-inspired “Genetic Algorithm” in clustering.
- As we can see that energy consumption is less using GA as compared to LEACH. We also worked on multiple base stations in the network. The graphs resulted in decrease in average energy consumption when compared to that with the single base station network.
- Our work has resulted in the most desired output that is to sustain the network for longer period. We have measured it with respect to number alive nodes and dead nodes in each round. This research will put a great impact on monitoring the remote areas or restricted areas.

# 8. REFERENCES


- https://www.researchgate.net/publication/338422869_TENCON_2019
- https://www.mathworks.com/matlabcentral/fileexchange/44073-low-energy-adaptive-clustering-hierarchy-protocol-leach
- https://en.m.wikipedia.org/wiki/wireless_sensor_network
- https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3?gi=42d3f1e819c5
- https://towardsdatascience.com/introduction-to-optimization-with-genetic-algorithm-2f5001d9964b
- https://en.m.wikipedia.org/wiki/Genetic_algorithm
- https://youtu.be/uq0HF7yt_wl
- https://www.researchgate.net/publication/2617069505_Survey_of_routing_protocols_in_wireless_sensor_networks
- https://www.researchgate.net/publication/309770246_A_Study_on_genetic_Algorithm_and_its_Applications
- https://www.mathworks.com/matlabcentral/fileexchange/71378-yarpiz-evolutionary-algorithms-toolbox-ypea