



**VIT<sup>®</sup>**  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

**FALLSEM 2020**

Project Report

**Clustering and General Self-Organized Tree based Energy-Balance Routing  
Protocol for Wireless Sensor Network (CGSTEB)**

***ECE2006 – Digital Signal Processing***

***SLOT – L43+L44                      /                      Group No.: 28-29***

**Submitted**

To	From
Dr. J. Valarmathi,  Professor, SENSE	18BEC0967, Vikram Baruah  18BEC0949, Mohammed Zain Mehaboob

**B-Tech (ECE)**

**School of Electronics Engineering (SENSE)**

## **Objective**

- WSNs have become major area of research in computational theory due to its wide range of applications. But due to limited battery power the energy consumption has become major limitations of WSNs protocols.
- Though many protocols have been proposed so far to improve the energy efficiency further but still much enhancement can be done.
- Although GSTEB has shown quite significant results over available protocols but it can be further improved using clustering based mechanism.
- This paper has proposed a new clustering and tree based routing protocol for wireless sensor networks.
- The proposed technique utilizes the leach based clustering protocol and improves the GSTEB further by increasing the stability period.
- The experimental results have shown the significant improvement.
- The aim is to not only reduce the energy consumption but also balance the load in the network.

## **Introduction**

One big problem of Wireless Sensor Networks (WSN) is that they die very quickly if they are not energy efficient enough. The current protocol used for WSN's is commonly known as General Self-Organized Tree-Based Energy-Balance Routing Protocol (GSTEB). It uses basic hierarchical structuring to somewhat order the nodes and increase efficiency compared to a completely random arrangement.

## Literature Survey

Publication	Title	Conclusion
Proc.IEEE INFOCOM / 2000	Energy conserving routing in wireless ad hoc networks	In order to maximize the lifetime, the traffic should be routed such that energy is balanced according to energy reserves.
Int. Conf. SystemSci / 20000	Energy efficient communication protocols for wireless microsensor networks	Using LEACH protocol is nearly 8 times as efficient compared to random energy dissipation
WSNA'02 / 2002	Wireless Sensor Networks for Habitat Monitoring	This paper looks at a real world WSN. It states that one of the bigger challenges is to increase the lifetime of the WSN
Proc. IEEE Aerospace Conf. / 2002	Pegasis: Powerefficient gathering in sensor information systems	This paper looks at protocols than are better than LEACH. This paper states than PEGASIS is better than LEACH, proving that we can improve on LEACH
SIGMOD Rec / 2003	Power efficient data gathering and aggregation in wireless sensor networks	This paper looks at two other protocols, which can outperform LECH, but require a specific setup

However, WSN can still be made more efficient. This project tries to come up with a newer, more efficient protocol. Energy signal is used to perform a MATLAB simulation of CGSTEB. The performance is evaluated for the proposed CGSTEB scheme and compared against GSTEB. The sensor nodes are deployed at random and n number of energy signals are noted. These energy signals are used to find the minimum distance and compared with the initial energy signal ( $E_0$ ). If  $d$  is less than  $d_0$  then energy consumption is updated. After the creation of the clusters, nodes in the cluster start transmitting the data they currently have throughout their allocated transmission time to the cluster-head (cluster-head node keeps its receiver on all the time to receive the sent data). Once all the data (sent by nodes in the cluster) have been received by the

cluster-head node, it will perform signal processing functions to compress the data into a single signal (the steady-state operation of LEACH networks).

## Existing Protocols

- *LEACH (Low-Energy Adaptive clustering Hierarchy):*

Basic clusters of nodes are formed, with one node acting as the cluster head (CH). 3-8 times as efficient compared to unorganized nodes.

- *HEED (hybrid, energy-efficient, distributed clustering algorithm):*

Only one CH in a uniform range, so better distribution and efficiency. Requires nodes with different energy

- *PEGASIS (Power-Efficient Gathering in Sensor Information Systems):*

All nodes form a chain with one node as a leader that directly communicates with the base station. Reduces data needed for long distance communication

- *TBC (Tree-Based Clustering):*

Within the clusters, a tree model is formed. The nodes record the data of their neighbor to more effectively build topography.

- *PEACH (Proxy-Enable Adaptive Clustering Hierarchy):*

A proxy node is selected each round as the cluster head. It has lesser energy than the other nodes.

- *EDACH (Energy-Driven Adaptive Clustering Hierarchy):*

This protocol employs simulation-based fault injection method to select cluster heads. Extends lifetime by 50% compared to LEACH

- *DEEC (distributed energy-efficient clustering algorithm):*

Nodes having higher than usual initial or residual energy are selected as cluster heads.

## Methodology

- Energy signal is used to perform a MATLAB simulation of CGSTEB. The performance is evaluated for the proposed CGSTEB scheme and compared against GSTEB.
- The sensor nodes are deployed at random and n number of energy signals are noted.
- These energy signals are used to find the minimum distance and compared with the initial energy signal ( $E_0$ ). If  $d$  is less than  $d_0$  then energy consumption is updated.
- After the creation of the clusters, nodes in the cluster start transmitting the data they currently have throughout their allocated transmission time to the cluster-head (cluster-head node keeps its receiver on all the time to receive the sent data).
- Once all the data (sent by nodes in the cluster) have been received by the cluster-head node, it will perform signal processing functions to compress the data into a single signal. (the steady-state operation of LEACH networks)

## Algorithm

1. We deploy the sensor nodes randomly
2. We decided cluster-heads based on number of nodes
3. Associate the nodes with their respective cluster heads
4. Evaluate how much energy would be lost and check if the node is dead
5. Check the total number of dead nodes
6. If the number is over the threshold, we end and record the number of rounds

# MATLAB Code

## CGSTEB

```
Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m
LeachCodeOnline.m  x  gsteb.m  x  +

1
2 ~
3 ~
4 ~
5 ~
6 ~
7 ~
8 ~
9 ~
10 ~
11 ~
12 ~
13 ~
14 ~
15 ~
16 ~
17 ~
18 ~
19 ~
20 ~
21 ~
22 ~
23 ~
24 ~
25 ~
26 ~
27 ~
28 ~
29 ~
30 ~
31 ~
32 ~
33 ~
34 ~
35 ~
36 ~
37 ~
38 ~
39 ~
40 ~
41 ~
42 ~
43 ~

clc
clear all;
xm=100;
ym=100;

sink.x=0.5*xm;
sink.y=0.5*ym;

n=100;

p=0.1;
%Energy Model (all values in Joules)
%Initial Energy
Eo=0.01;
%Eelec=Extx*Erx
ETX=50*0.000000001;
ERX=50*0.000000001;
%Transmit Amplifier types
Efa=10*0.000000000001;
Emp=0.0013*0.000000000001;
%Data Aggregation Energy
EDA=5*0.000000001;
%Values for Heterogeneity
%Percentage of nodes than are advanced
m=0.1;
%alpha
a=1;
%maximum number of rounds
rmax=500
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% END OF PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Computation of do/
do=sqrt(Efa/Emp);
%Creation of the random Sensor Network
figure(1);
for i=1:n
    S(i).xd=rand(1,1)*xm;
    XR(i)=S(i).xd;
    S(i).yd=rand(1,1)*ym;
    YR(i)=S(i).yd;
    S(i).G=0;
    %Initially there are no cluster heads only nodes
    S(i).type='N';
end
```

```
Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m
LeachCodeOnline.m  x  gsteb.m  x  +

44
45 ~
46 ~
47 ~
48 ~
49 ~
50 ~
51 ~
52 ~
53 ~
54 ~
55 ~
56 ~
57 ~
58 ~
59 ~
60 ~
61 ~
62 ~
63 ~
64 ~
65 ~
66 ~
67 ~
68 ~
69 ~
70 ~
71 ~
72 ~
73 ~
74 ~
75 ~
76 ~
77 ~
78 ~
79 ~
80 ~
81 ~
82 ~
83 ~
84 ~
85 ~
86 ~
87 ~

temp_rnd=1;
%Random Election of Normal Nodes
if (temp_rnd>m*n+1)
    S(i).E=Eo;
    S(i).ENERGY=0;
    plot(S(i).xd,S(i).yd,'o');
    hold on;
end
%Random Election of Advanced Nodes
if (temp_rnd<m*n+1)
    S(i).E=Eo*(1+a);
    S(i).ENERGY=1;
    plot(S(i).xd,S(i).yd,'+');
    hold on;
end
end
S(n+1).xd=sink.x;
S(n+1).yd=sink.y;
plot(S(n+1).xd,S(n+1).yd,'x');

%First Iteration
figure(1);
%counter for CHs
countCHs=0;
%counter for CHs per round
rcountCHs=0;
cluster=1;
countCHs;
rcountCHs=rcountCHs+countCHs;
flag_first_dead=0;
for r=0:rmax
    %Operation for epoch
    if(mod(r, round(1/p))==0)
        for i=1:n
            S(i).G=0;
            S(i).cl=0;
        end
    end
end
hold off;
%Number of dead nodes
```

```

Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m
LeachCodeOnline.m  x  gsteb.m  x  +
100 %When the first node dies
101 %Number of dead nodes
102 dead=0;
103 %Number of dead Advanced Nodes
104 dead_a=0;
105 %Number of dead Normal Nodes
106 dead_n=0;
107 %counter for bit transmitted to Bases Station and to Cluster Heads
108 packets_TO_BS=0;
109 packets_TO_CH=0;
110 %counter for bit transmitted to Bases Station and to Cluster Heads
111 PACKETS_TO_CH(r+1)=0;
112 PACKETS_TO_BS(r+1)=0;
113 figure(1);
114 for i=1:l:n
115     %checking if there is a dead node
116     if (S(i).E<=0)
117         plot(S(i).xd,S(i).yd,'red .');
118         dead=dead+1;
119         if (S(i).ENERGY==1)
120             dead_a=dead_a+1;
121         end
122         if (S(i).ENERGY==0)
123             dead_n=dead_n+1;
124         end
125         hold on;
126     end
127     if S(i).E>0
128         S(i).type='N';
129         if (S(i).ENERGY==0)
130             plot(S(i).xd,S(i).yd,'o');
131         end
132         if (S(i).ENERGY==1)
133             plot(S(i).xd,S(i).yd,'+');
134         end
135         hold on;
136     end
137 end
138 plot(S(n+1).xd,S(n+1).yd,'x');
139 STATISTICS(r+1).DEAD=dead;
140 DEAD(r+1)=dead;
141 DEAD_N(r+1)=dead_n;
142 DEAD_A(r+1)=dead_a;
143 %When the first node dies

```

```

Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m
LeachCodeOnline.m  x  gsteb.m  x  +
129 %When the first node dies
130 if (dead==1)
131     if (flag_first_dead==0)
132         first_dead=r;
133         flag_first_dead=1;
134     end
135 end
136 countCHs=0;
137 cluster=1;
138 for i=1:l:n
139     if (S(i).E>0)
140         temp_rand=rand;
141         if ( (S(i).G)<=0)
142             %Election of Cluster Heads
143             if (temp_rand<= (p/(1-p*mod(r,round(1/p))))))
144                 countCHs=countCHs+1;
145                 packets_TO_BS=packets_TO_BS+1;
146                 PACKETS_TO_BS(r+1)=packets_TO_BS;
147
148                 S(i).type='C';
149                 S(i).G=round(1/p)-1;
150                 C(cluster).xd=S(i).xd;
151                 C(cluster).yd=S(i).yd;
152                 plot(S(i).xd,S(i).yd,'k*');
153
154                 distance=sqrt( (S(i).xd-(S(n+1).xd))^2 + (S(i).yd-(S(n+1).yd))^2 );
155                 C(cluster).distance=distance;
156                 C(cluster).id=i;
157                 X(cluster)=S(i).xd;
158                 Y(cluster)=S(i).yd;
159                 cluster=cluster+1;
160
161                 %Calculation of Energy dissipated
162                 distance;
163                 if (distance>do)
164                     S(i).E=S(i).E- ( (ETX+EDA)*(4000) + Emp*4000*( distance*distance*distance*distance ));
165                 end
166                 if (distance<=do)
167                     S(i).E=S(i).E- ( (ETX+EDA)*(4000) + Efs*4000*( distance * distance ));
168                 end
169             end
170         end
171     end

```

```

Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m
LeachCodeOnline.m  x  gsteb.m  x  +

170
171     end
172     end
173 -end
174 STATISTICS(r+1).CLUSTERHEADS=cluster-1;
175 CLUSTERHS(r+1)=cluster-1;
176 %Election of Associated Cluster Head for Normal Nodes
177 for i=1:l:n
178     if ( S(i).type=='N' && S(i).E>0 )
179         if (cluster-1>1)
180             min_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );
181             min_dis_cluster=1;
182             for c=1:1:cluster-1
183                 temp=min(min_dis,sqrt( (S(i).xd-C(c).xd)^2 + (S(i).yd-C(c).yd)^2 ) );
184                 if ( temp<min_dis )
185                     min_dis=temp;
186                     min_dis_cluster=c;
187                 end
188             end
189
190             %Energy dissipated by associated Cluster Head
191             min_dis;
192             if (min_dis>do)
193                 S(i).E=S(i).E- ( ETX*(4000) + Emp*4000*( min_dis * min_dis * min_dis * min_dis));
194             end
195             if (min_dis<=do)
196                 S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*( min_dis * min_dis));
197             end
198             %Energy dissipated
199             if (min_dis>0)
200                 S(C(min_dis_cluster).id).E = S(C(min_dis_cluster).id).E- ( ERX + EDA)*4000 );
201                 PACKETS_TO_CH(r+1)=n-dead-cluster+1;
202             end
203             S(i).min_dis=min_dis;
204             S(i).min_dis_cluster=min_dis_cluster;
205
206         end
207     end
208 -end
209 hold on;
210 countCHs;
211 rcountCHs=rcountCHs+countCHs
212 end

```

## GSTEB

```

Editor - C:\Users\vikra\Downloads\gsteb.m
LeachCodeOnline.m  x  gsteb.m*  x  +

1 - clc
2 - clear all;
3 - xm=100;
4 - ym=100;
5
6 - sink.x=0.5*xm;
7 - sink.y=0.5*ym;
8
9 - n=100
10
11 - p=0.1;
12 %Energy Model (all values in Joules)
13 %Initial Energy
14 Eo=0.01;
15 %Elec=ETx=Erx
16 ETX=50*0.000000001;
17 ERX=50*0.000000001;
18 %Transmit Amplifier types
19 Efs=10*0.000000000001;
20 Emp=0.0013*0.000000000001;
21 %Data Aggregation Energy
22 EDA=5*0.000000001;
23 %Values for Heterogeneity
24 %Percentage of nodes than are advanced
25 m=0.1;
26 %alpha
27 a=1;
28 %maximum number of rounds
29 rmax=400
30 %***** END OF PARAMETERS *****
31 %Computation of do/
32 do=sqrt(Efs/Emp);
33 %Creation of the random Sensor Network
34 figure(1);
35 for i=1:l:n
36     S(i).xd=rand(1,1)*xm;
37     XR(i)=S(i).xd;
38     S(i).yd=rand(1,1)*ym;
39     YR(i)=S(i).yd;
40     S(i).G=0;
41     %initially there are no cluster heads only nodes
42     S(i).type='N';
43

```



```
Editor - C:\Users\vikra\Downloads\gsteb.m
LeachCodeOnline.m x gsteb.m +
43 -
44 -     temp_rnd0=i;
45 -     %Random Election of Normal Nodes
46 -     if (temp_rnd0>=m*n+1)
47 -         S(i).E=Eo;
48 -         S(i).ENERGY=0;
49 -         plot(S(i).xd,S(i).yd,'o');
50 -         hold on;
51 -     end
52 -     %Random Election of Advanced Nodes
53 -     if (temp_rnd0<m*n+1)
54 -         S(i).E=Eo*(1+a)
55 -         S(i).ENERGY=1;
56 -         plot(S(i).xd,S(i).yd,'+');
57 -         hold on;
58 -     end
59 - end
60 -
61 - S(n+1).xd=sink.x;
62 - S(n+1).yd=sink.y;
63 - plot(S(n+1).xd,S(n+1).yd,'x');
64 -
65 -
66 - %First Iteration
67 - figure(1);
68 - %counter for CHs
69 - countCHs=0;
70 - %counter for CHs per round
71 - rcountCHs=0;
72 - cluster=1;
73 - countCHs;
74 - rcountCHs=rcountCHs+countCHs;
75 - flag_first_dead=0;
76 - for r=0:1:rmax
77 -     %
78 -     %Operation for epoch
79 -     if(mod(r, round(1/p))==0)
80 -         for i=1:1:n
81 -             S(i).G=0;
82 -             S(i).cl=0;
83 -         end
84 -     end
85 -     hold off;
```

```
Editor - C:\Users\vikra\Downloads\gsteb.m
LeachCodeOnline.m x gsteb.m +
85 -
86 -     hold off;
87 -     %Number of dead nodes
88 -     dead=0;
89 -     %Number of dead Advanced Nodes
90 -     dead_a=0;
91 -     %Number of dead Normal Nodes
92 -     dead_n=0;
93 -     %counter for bit transmitted to Bases Station and to Cluster Heads
94 -     packets_TO_BS=0;
95 -     packets_TO_CH=0;
96 -     %counter for bit transmitted to Bases Station and to Cluster Heads
97 -     PACKETS_TO_CH(r+1)=0;
98 -     PACKETS_TO_BS(r+1)=0;
99 -     figure(1);
100 -     for i=1:1:n
101 -         %checking if there is a dead node
102 -         if (S(i).E<=0)
103 -             plot(S(i).xd,S(i).yd,'red .');
104 -             dead=dead+1;
105 -             if (S(i).ENERGY==1)
106 -                 dead_a=dead_a+1;
107 -             end
108 -             if (S(i).ENERGY==0)
109 -                 dead_n=dead_n+1;
110 -             end
111 -             hold on;
112 -         end
113 -         if S(i).E>0
114 -             S(i).type='N';
115 -             if (S(i).ENERGY==0)
116 -                 plot(S(i).xd,S(i).yd,'o');
117 -             end
118 -             if (S(i).ENERGY==1)
119 -                 plot(S(i).xd,S(i).yd,'+');
120 -             end
121 -             hold on;
122 -         end
123 -     end
124 -     plot(S(n+1).xd,S(n+1).yd,'x');
125 -     STATISTICS(r+1).DEAD=dead;
126 -     DEAD(r+1)=dead;
127 -     DEAD_N(r+1)=dead_n;
128 -     DEAD_A(r+1)=dead_a;
```

```

Editor - C:\Users\vikra\Downloads\gsteb.m
LeachCodeOnline.m x gsteb.m x +
127 - DEAD_A(r+1)=dead_a;
128 - %When the first node dies
129 - if (dead==1)
130 -     if(flag_first_dead==0)
131 -         first_dead=r;
132 -         flag_first_dead=1;
133 -     end
134 - end
135 - countCHs=0;
136 - cluster=1;
137 - for i=1:l:n
138 -     if(S(i).E>0)
139 -         temp_rand=rand;
140 -         if ( (S(i).G)<=0)
141 -             %Election of Cluster Heads
142 -             if(temp_rand<= (p/(1-p)*mod(x,round(1/p))))
143 -                 countCHs=countCHs+1;
144 -                 packets_TO_BS=packets_TO_BS+1;
145 -                 PACKETS_TO_BS(r+1)=packets_TO_BS;
146 -
147 -                 S(i).type='C';
148 -                 S(i).G=round(1/p)-1;
149 -                 C(cluster).xd=S(i).xd;
150 -                 C(cluster).yd=S(i).yd;
151 -                 plot(S(i).xd,S(i).yd,'k');
152 -
153 -                 distance=sqrt( (S(i).xd-(S(n+1).xd))^2 + (S(i).yd-(S(n+1).yd))^2 );
154 -                 C(cluster).distance=distance;
155 -                 C(cluster).id=i;
156 -                 X(cluster)=S(i).xd;
157 -                 Y(cluster)=S(i).yd;
158 -                 cluster=cluster+1;
159 -
160 -                 %Calculation of Energy dissipated
161 -                 distance;
162 -                 if (distance>do)
163 -                     S(i).E=S(i).E- ( (ETX+EDA)*(4000) + Emp*4000*( distance*distance*distance*distance ));
164 -                 end
165 -                 if (distance<=do)
166 -                     S(i).E=S(i).E- ( (ETX+EDA)*(4000) + Efs*4000*( distance * distance ));
167 -                 end
168 -             end
169 -

```

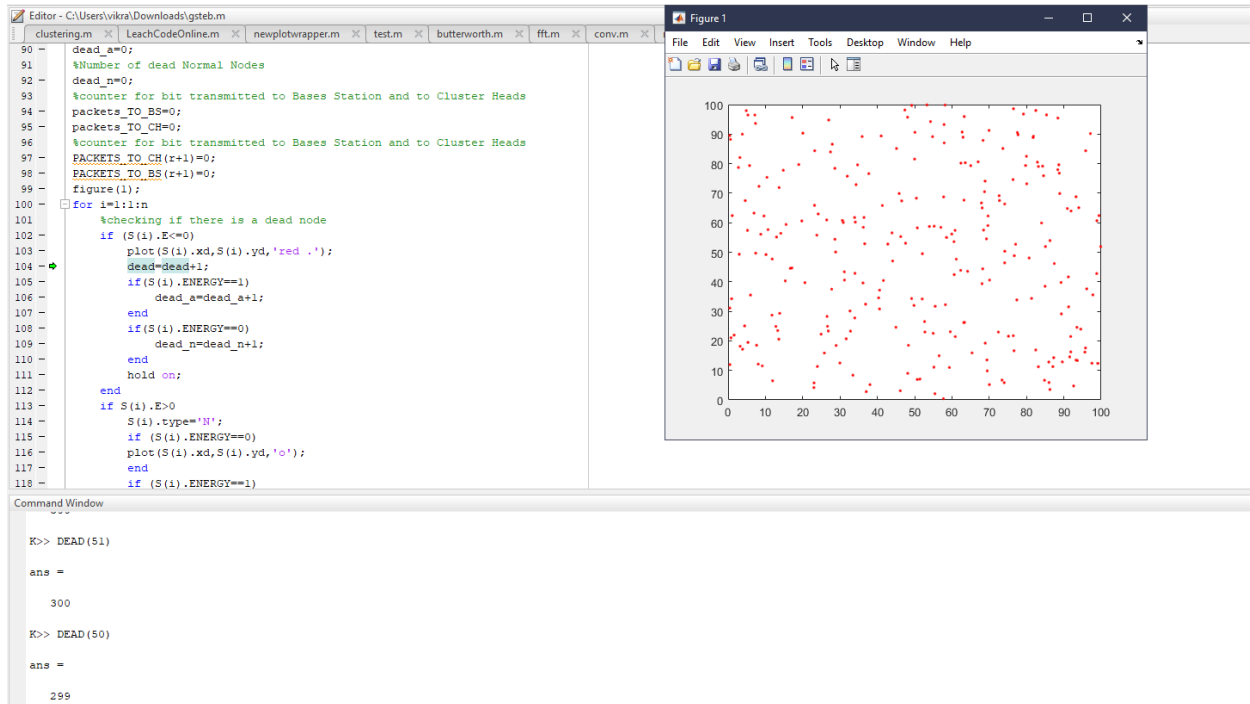
```

Editor - C:\Users\vikra\Downloads\gsteb.m
LeachCodeOnline.m x gsteb.m x +
157 -     Y(cluster)=S(i).yd;
158 -     cluster=cluster+1;
159 -
160 -     %Calculation of Energy dissipated
161 -     distance;
162 -     if (distance>do)
163 -         S(i).E=S(i).E- ( (ETX+EDA)*(4000) + Emp*4000*( distance*distance*distance*distance ));
164 -     end
165 -     if (distance<=do)
166 -         S(i).E=S(i).E- ( (ETX+EDA)*(4000) + Efs*4000*( distance * distance ));
167 -     end
168 - end
169 -
170 - end
171 - end
172 - end
173 - STATISTICS(r+1).CLUSTERHEADS=cluster-1;
174 - CLUSTERS(r+1)=cluster-1;
175 - %Election of Associated Cluster Head for Normal Nodes
176 - for i=1:l:n
177 -     if ( S(i).type=='N' && S(i).E>0 )
178 -         min_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );
179 -
180 -         %Energy dissipated by associated Cluster Head
181 -         min_dis;
182 -         if (min_dis>do)
183 -             S(i).E=S(i).E- ( ETX*(4000) + Emp*4000*( min_dis * min_dis * min_dis * min_dis ));
184 -         end
185 -         if (min_dis<=do)
186 -             S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*( min_dis * min_dis ));
187 -         end
188 -         %Energy dissipated
189 -         if(min_dis>0)
190 -             S(i).E = S(i).E- ( (ERX + EDA)*4000 );
191 -         end
192 -         S(i).min_dis=min_dis;
193 -
194 -     end
195 - end
196 - hold on;
197 -
198 -
199 - end

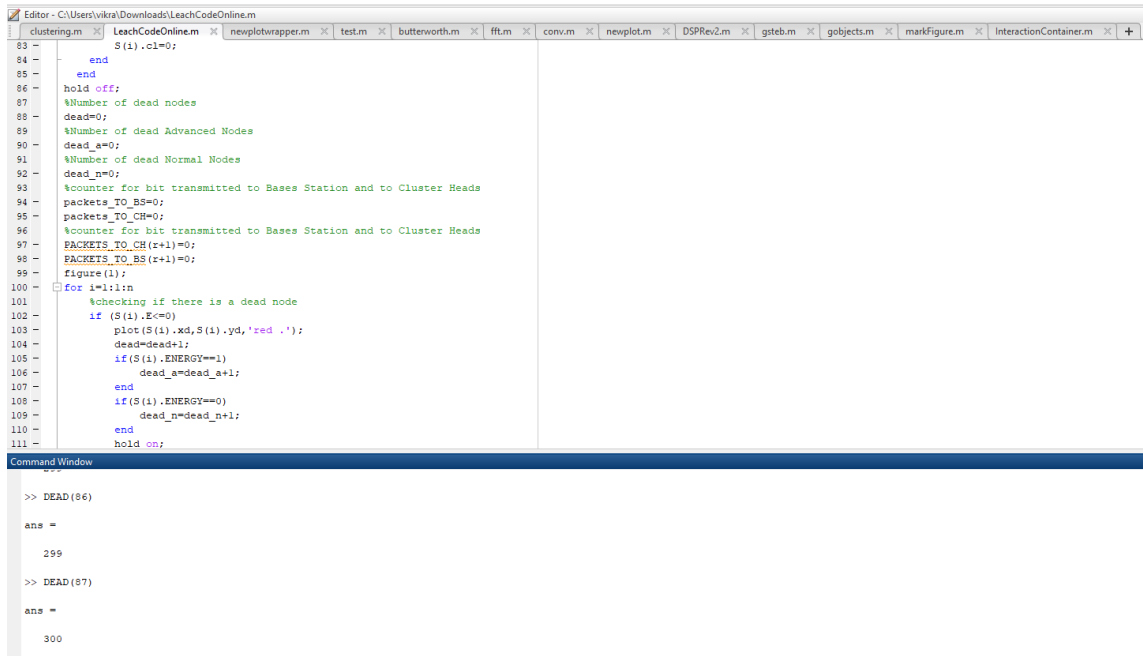
```

## Output & Results

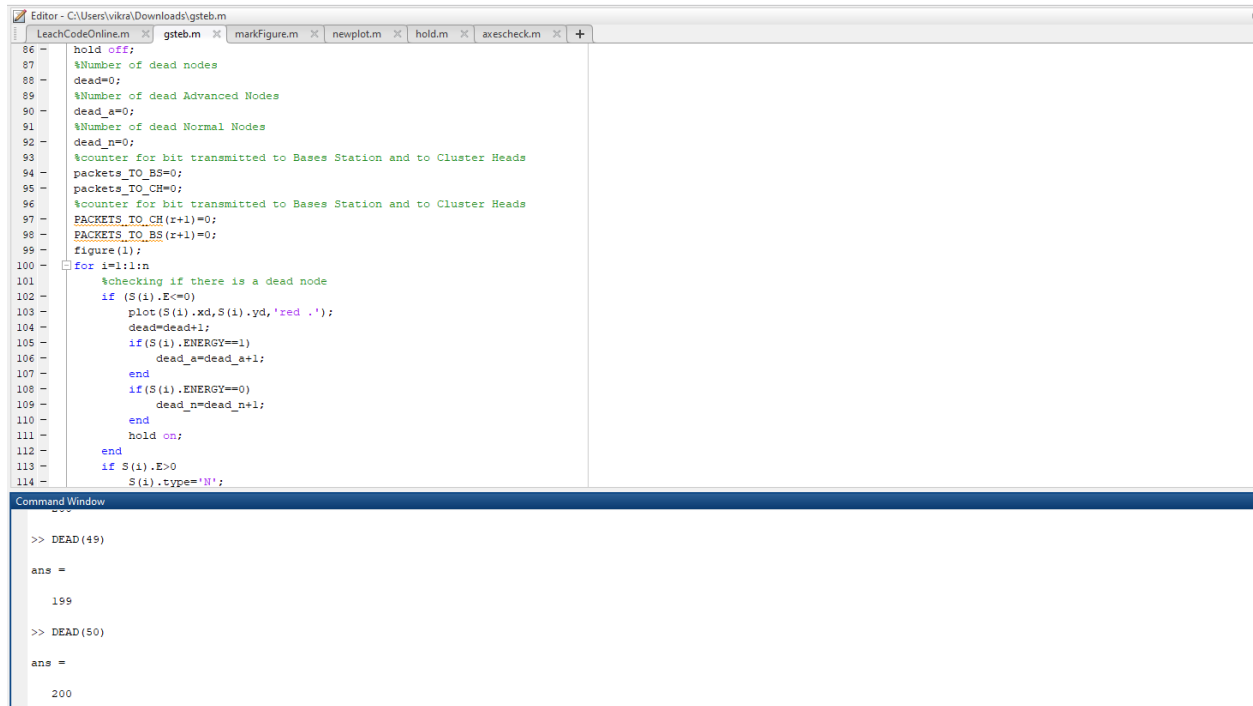
$N=300$ ,  $E_i = 0.01$ , GSTEB



$N=300$ ,  $E_i = 0.01$ , CGSTEB



$N=200$ ,  $E_i = 0.01$ , GSTEB



The image shows a MATLAB Editor window with a script named 'gsteb.m' and a Command Window. The script defines parameters for a simulation, including the number of nodes (N=200), energy (E<sub>i</sub>=0.01), and various counters for dead nodes and packets. It includes a loop to check for dead nodes and update the simulation state. The Command Window shows the results of the 'DEAD' function calls.

```
Editor - C:\Users\vikra\Downloads\gsteb.m
LeachCodeOnline.m x gsteb.m x markFigure.m x newplot.m x hold.m x axescheck.m x +

56 - hold off;
57 - %Number of dead nodes
58 - dead=0;
59 - %Number of dead Advanced Nodes
60 - dead_a=0;
61 - %Number of dead Normal Nodes
62 - dead_n=0;
63 - %counter for bit transmitted to Bases Station and to Cluster Heads
64 - packets_TO_BS=0;
65 - packets_TO_CH=0;
66 - %counter for bit transmitted to Bases Station and to Cluster Heads
67 - PACKETS_TO_CH(r+1)=0;
68 - PACKETS_TO_BS(r+1)=0;
69 - figure(1);
70 - for i=1:1:n
71 -     %checking if there is a dead node
72 -     if (S(i).E<=0)
73 -         plot(S(i).xd,S(i).yd,'red .');
74 -         dead=dead+1;
75 -         if(S(i).ENERGY==1)
76 -             dead_a=dead_a+1;
77 -         end
78 -         if(S(i).ENERGY==0)
79 -             dead_n=dead_n+1;
80 -         end
81 -     end
82 -     hold on;
83 - end
84 - if S(i).E>0
85 -     S(i).type='N';
86 - end

Command Window

>> DEAD(49)

ans =

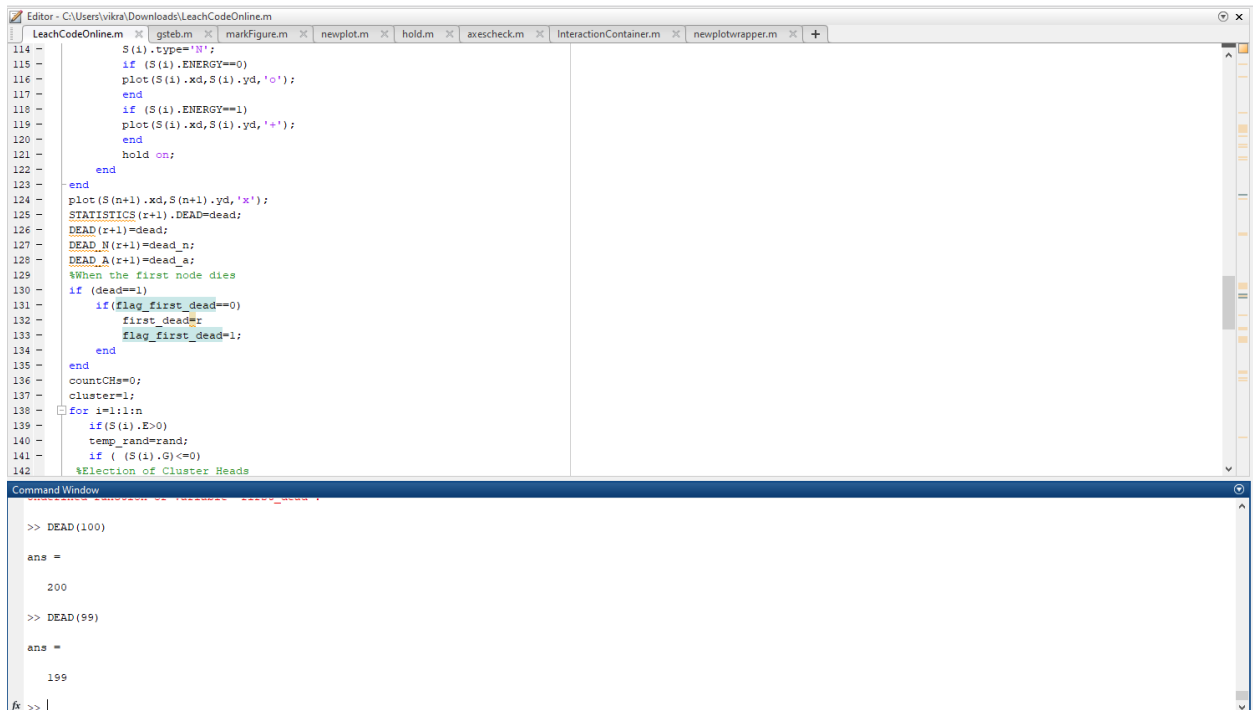
    199

>> DEAD(50)

ans =

    200
```

$N=200$ ,  $E_i = 0.01$ , CGSTEB



The image shows a MATLAB Editor window with a script named 'cgsteb.m' and a Command Window. The script defines parameters for a simulation, including the number of nodes (N=200), energy (E<sub>i</sub>=0.01), and various counters for dead nodes and packets. It includes a loop to check for dead nodes and update the simulation state. The Command Window shows the results of the 'DEAD' function calls.

```
Editor - C:\Users\vikra\Downloads\LeachCodeOnline\
LeachCodeOnline.m x cgsteb.m x markFigure.m x newplot.m x hold.m x axescheck.m x InteractionContainer.m x newplotwrapper.m x +

114 - S(i).type='N';
115 - if (S(i).ENERGY==0)
116 -     plot(S(i).xd,S(i).yd,'o');
117 - end
118 - if (S(i).ENERGY==1)
119 -     plot(S(i).xd,S(i).yd,'+');
120 - end
121 - hold on;
122 - end
123 - end
124 - plot(S(n+1).xd,S(n+1).yd,'x');
125 - STATISTICS(r+1).DEAD=dead;
126 - DEAD(r+1)=dead;
127 - DEAD_A(r+1)=dead_a;
128 - DEAD_N(r+1)=dead_n;
129 - %When the first node dies
130 - if (dead==1)
131 -     if(flag_first_dead==0)
132 -         first_dead=r;
133 -         flag_first_dead=1;
134 -     end
135 - end
136 - countCHs=0;
137 - cluster=1;
138 - for i=1:1:n
139 -     if(S(i).E>0)
140 -         temp_rand=rand;
141 -         if ( (S(i).G)<=0)
142 -             %Election of Cluster Heads
143 -         end
144 -     end
145 - end

Command Window

>> DEAD(100)

ans =

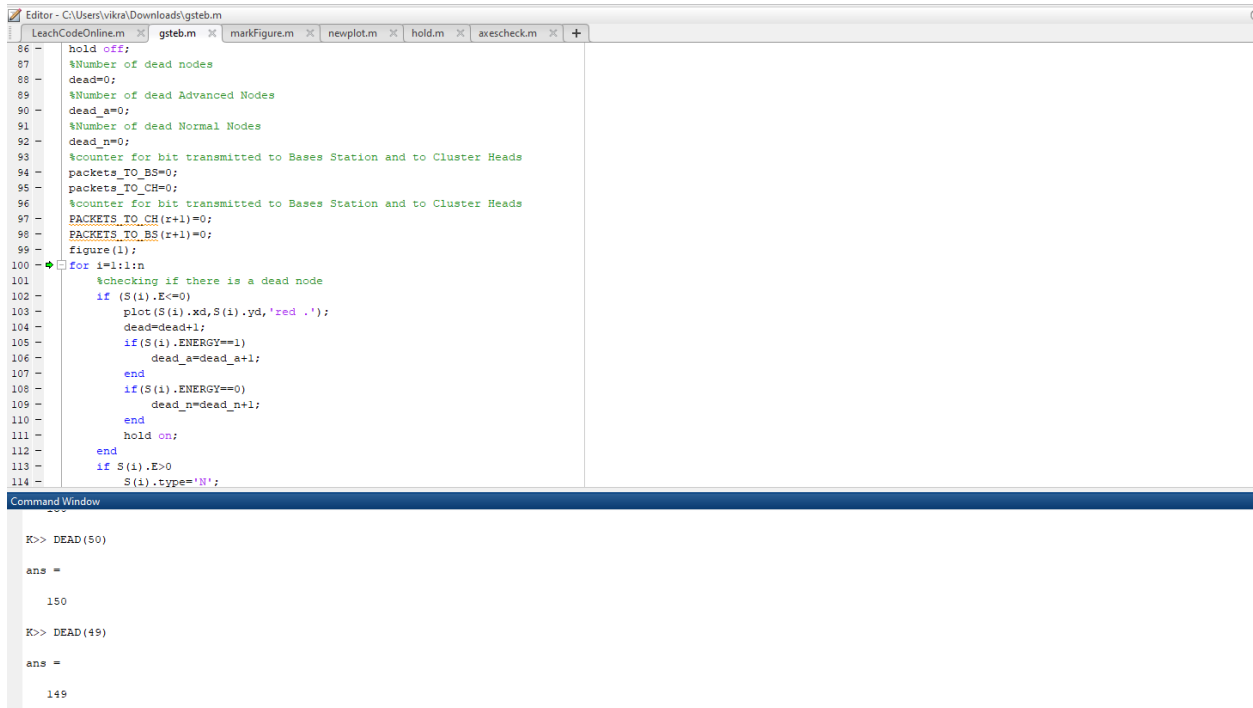
    200

>> DEAD(99)

ans =

    199
```

$N=150$ ,  $E_i = 0.01$ , GSTEB



The image shows a MATLAB Editor window with a script named 'gsteb.m' and a Command Window. The script contains a loop for  $i=1:1:n$  that checks for dead nodes based on energy levels. The Command Window shows the output of the 'DEAD' function.

```

66 hold off;
67 %Number of dead nodes
68 dead=0;
69 %Number of dead Advanced Nodes
70 dead_a=0;
71 %Number of dead Normal Nodes
72 dead_n=0;
73 %counter for bit transmitted to Bases Station and to Cluster Heads
74 packets_TO_BS=0;
75 packets_TO_CH=0;
76 %counter for bit transmitted to Bases Station and to Cluster Heads
77 PACKETS_TO_CH(r+1)=0;
78 PACKETS_TO_BS(r+1)=0;
79 figure(1);
80 for i=1:1:n
81     %checking if there is a dead node
82     if (S(i).E<=0)
83         plot(S(i).xd,S(i).yd,'red .');
84         dead=dead+1;
85         if (S(i).ENERGY==1)
86             dead_a=dead_a+1;
87         end
88         if (S(i).ENERGY==0)
89             dead_n=dead_n+1;
90         end
91     end
92     hold on;
93 end
94 if S(i).E>0
95     S(i).type='N';
96 end

```

```

K>> DEAD(50)

ans =

    150

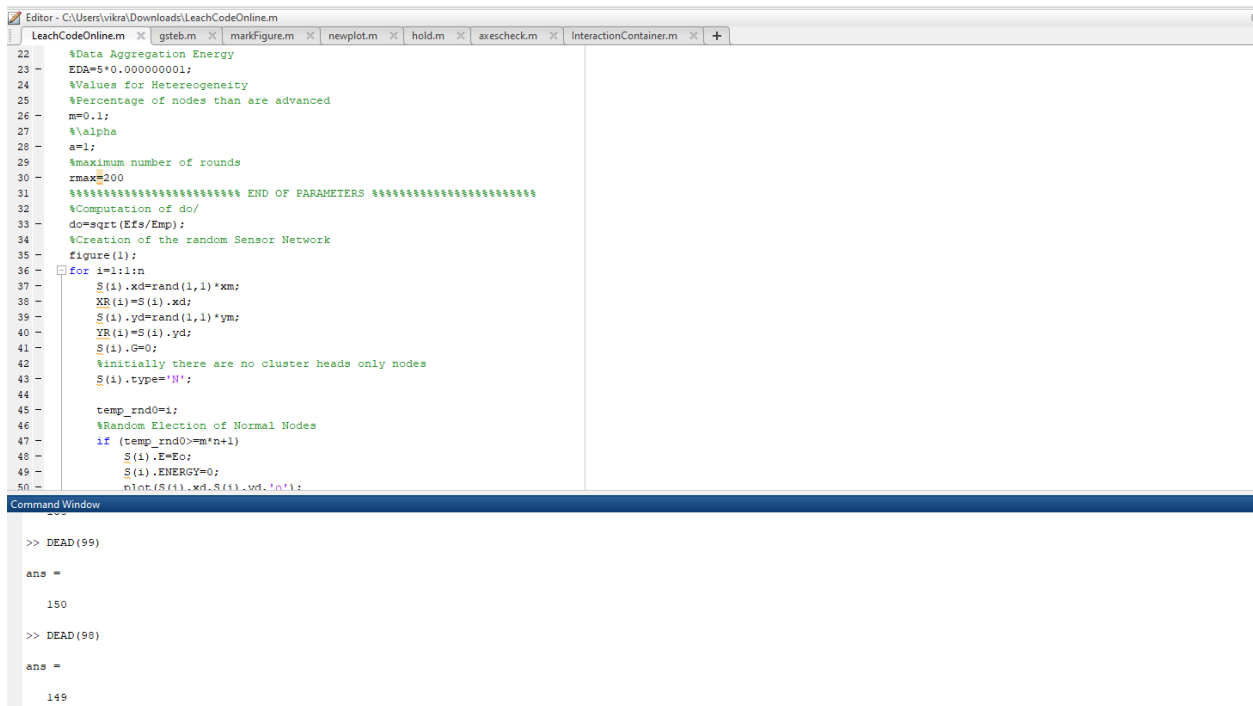
K>> DEAD(49)

ans =

    149

```

$N=150$ ,  $E_i = 0.01$ , CGSTEB



The image shows a MATLAB Editor window with a script named 'LeachCodeOnline.m' and a Command Window. The script contains parameters for data aggregation energy, heterogeneity, and node advancement. It also includes a loop for  $i=1:1:n$  that initializes nodes and checks for dead nodes. The Command Window shows the output of the 'DEAD' function.

```

22 %Data Aggregation Energy
23 EDA=5*0.000000001;
24 %Values for Heterogeneity
25 %Percentage of nodes than are advanced
26 m=0.1;
27 %\alpha
28 a=1;
29 %maximum number of rounds
30 rmax=200;
31 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% END OF PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32 %Computation of do/
33 do=sqrt(EDA/Emp);
34 %Creation of the random Sensor Network
35 figure(1);
36 for i=1:1:n
37     S(i).xd=rand(1,1)*xm;
38     XR(i)=S(i).xd;
39     S(i).yd=rand(1,1)*ym;
40     YR(i)=S(i).yd;
41     S(i).G=0;
42     %initially there are no cluster heads only nodes
43     S(i).type='N';
44
45     temp_rnd0=i;
46     %Random Election of Normal Nodes
47     if (temp_rnd0>=m*n+1)
48         S(i).E=Eo;
49         S(i).ENERGY=0;
50         plot(S(i).xd,S(i).yd,'o');

```

```

>> DEAD(99)

ans =

    150

>> DEAD(98)

ans =

    149

```

$N=100$ ,  $E_i = 0.01$ , GSTEB



The image shows a MATLAB Editor window with a script named 'gsteb.m' and a Command Window. The script defines parameters for a network simulation, including node count, energy model, and various energy thresholds. The Command Window shows the execution of the 'DEAD' function, which returns the number of dead nodes.

```
1 clc
2 clear all;
3 xm=100;
4 ym=100;
5
6
7 sink.x=0.5*xm;
8 sink.y=0.5*ym;
9
10 n=100;
11
12 p=0.1;
13 %Energy Model (all values in Joules)
14 %Initial Energy
15 Eo=0.01;
16 %Eleo=Etx=Erx
17 ETX=50*0.000000001;
18 ERX=50*0.000000001;
19 %Transmit Amplifier types
20 Efs=10*0.000000000001;
21 Emp=0.0013*0.000000000001;
22 %Data Aggregation Energy
23 EDA=5*0.000000001;
24 %Values for Heterogeneity
25 %Percentage of nodes than are advanced
26 m=0.1;
27 %\alpha
28 a=1;
29 %maximum number of rounds
```

```
>> DEAD(50)

ans =

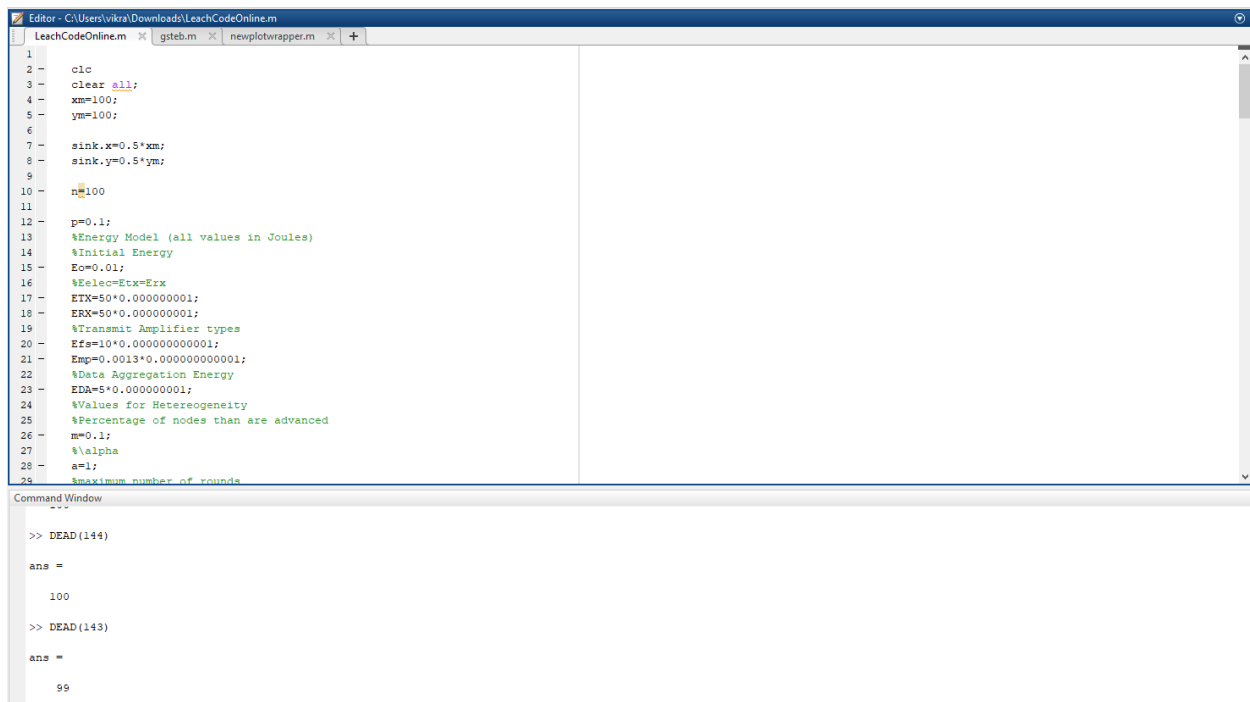
    98

>> DEAD(51)

ans =

   100
```

$N=100$ ,  $E_i = 0.01$ , CGSTEB



The image shows a MATLAB Editor window with a script named 'cgsteb.m' and a Command Window. The script defines parameters for a network simulation, including node count, energy model, and various energy thresholds. The Command Window shows the execution of the 'DEAD' function, which returns the number of dead nodes.

```
1 clc
2 clear all;
3 xm=100;
4 ym=100;
5
6
7 sink.x=0.5*xm;
8 sink.y=0.5*ym;
9
10 n=100;
11
12 p=0.1;
13 %Energy Model (all values in Joules)
14 %Initial Energy
15 Eo=0.01;
16 %Eleo=Etx=Erx
17 ETX=50*0.000000001;
18 ERX=50*0.000000001;
19 %Transmit Amplifier types
20 Efs=10*0.000000000001;
21 Emp=0.0013*0.000000000001;
22 %Data Aggregation Energy
23 EDA=5*0.000000001;
24 %Values for Heterogeneity
25 %Percentage of nodes than are advanced
26 m=0.1;
27 %\alpha
28 a=1;
29 %maximum number of rounds
```

```
>> DEAD(144)

ans =

   100

>> DEAD(143)

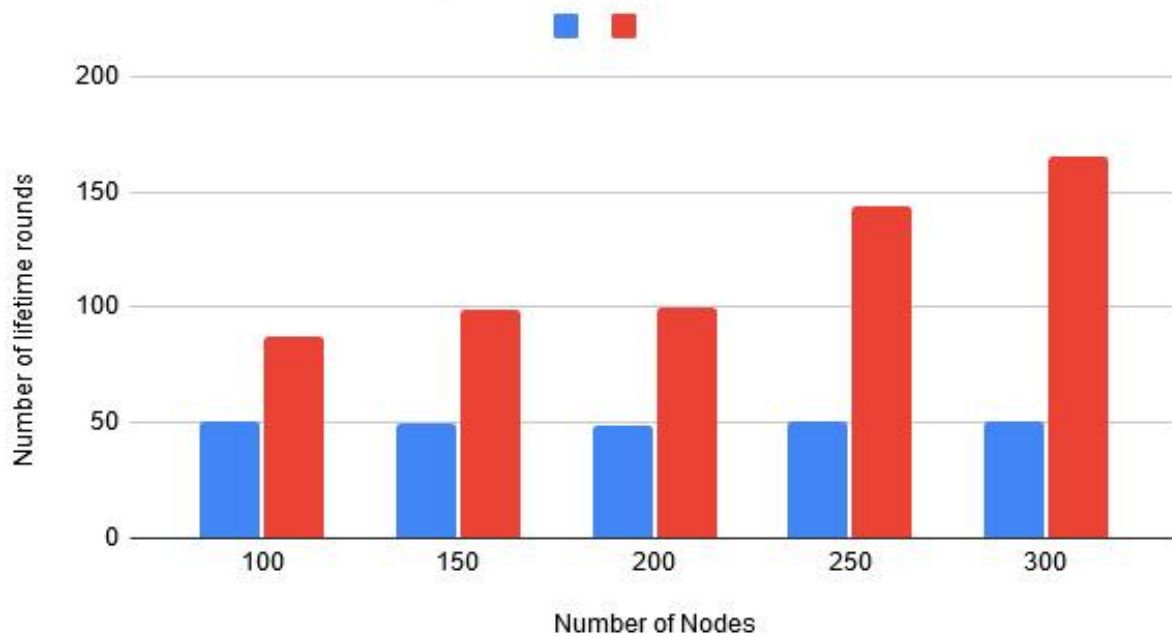
ans =

    99
```

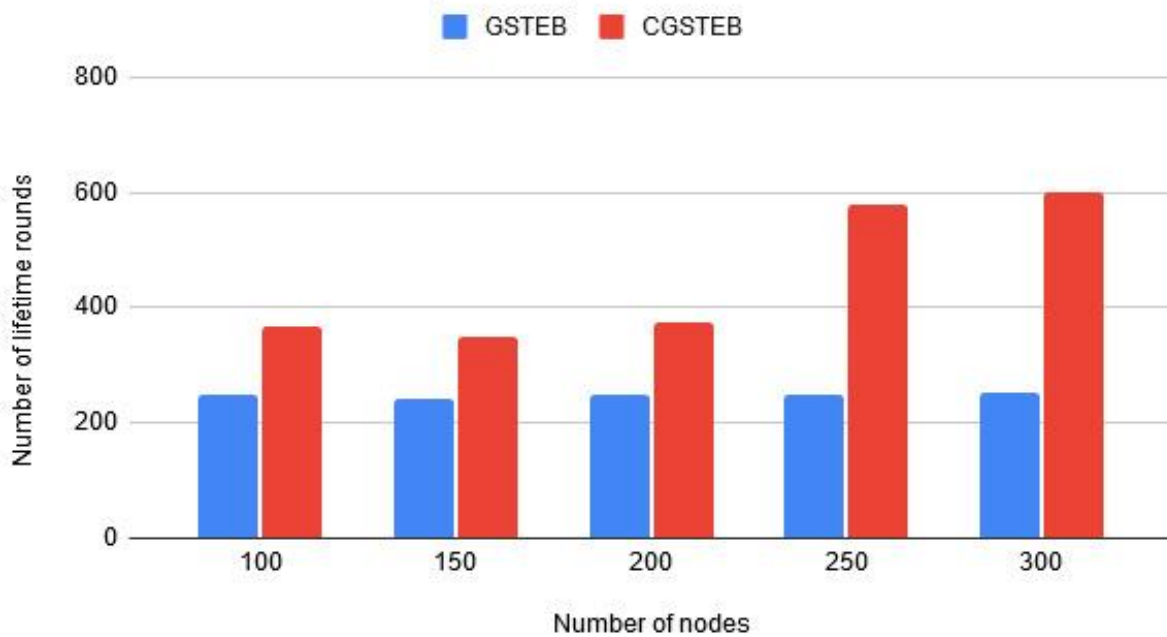
**Table (Number of rounds taken for all nodes to die)**

N	E <sub>i1</sub> (0.01)		E <sub>i2</sub> (0.05)	
	GSTEB	CGSTEB	GSTEB	CGSTEB
100	51 <sup>ST</sup> ROUND	144 <sup>TH</sup> ROUND	249 <sup>TH</sup> ROUND	367 <sup>ST</sup> ROUND
150	50 <sup>TH</sup> ROUND	99 <sup>TH</sup> ROUND	240 <sup>TH</sup> ROUND	349 <sup>TH</sup> ROUND
200	49 <sup>TH</sup> ROUND	100 <sup>TH</sup> ROUND	248 <sup>th</sup> ROUND	602 <sup>ND</sup> ROUND
250	51 <sup>ST</sup> ROUND	165 <sup>TH</sup> ROUND	248 <sup>TH</sup> ROUND	373 <sup>RD</sup> ROUND
300	51 <sup>ST</sup> ROUND	87 <sup>TH</sup> ROUND	251 <sup>ST</sup> ROUND	580 <sup>TH</sup> ROUND

GSTEB and CGSTEB ( $E=0.01$ )



GSTEB and CGSTEB ( $E=0.05$ )





## **Conclusion**

The number of rounds or the time taken for the nodes to die implementing the CGSTEB algorithm is greater compared to the traditional GSTEB algorithm. As the nodes survive for a longer time by clustering to effectively dissipate the energy within the nodes, the CGSTEB algorithm is far superior to GSTEB in terms of efficiency.

## **References**

- [1] J. H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks," in Proc. IEEE INFOCOM, vol. 1, pp. 22–31, 2000.
- [2] W. R. Heinzelman et al, "Energy efficient communication protocols for wireless microsensor networks," in Proc. 33rd Hawaii Int. Conf. SystemSci.,pp.3005–3014,Jan.2000.
- [3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," WSNA'02, Atlanta, Georgia, September 2002.
- [4] S. Lindsey and C. Raghavendra, "Pegasis: Powerefficient gathering in sensor information systems," in Proc. IEEE Aerospace Conf., vol. 3, pp. 1125–1130 , 2002.
- [5] H. O. Tan and I. Korpeoglu, "Power efficient data gathering and aggregation in wireless sensor networks," SIGMOD Rec., vol. 32, no. 4, pp. 66–71, 2003.