VIT®

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 (E0). If d is less than d0 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 ($E0$). If d is less than $d0$ 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 | gsteb.m

```matlab

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=Etx=Erx
ETX=50*0.000000001;
ERX=50*0.000000001;
%Transmit Amplifier types
Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;
%Data Aggregation Energy
EDA=5*0.000000001;
%Values for Hetereogeneity
%Percentage of nodes than are advanced
m=0.1;
%\alpha
a=1;
%maximum number of rounds
rmax=800
%%%%%%%%%%%%%%%%%%%%%%%%% END OF PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%
%Computation of do/
do=sqrt(Efs/Emp);
%Creation of the random Sensor Network
figure(1);
for i=1:1: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';
```

Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m

LeachCodeOnline.m | gsteb.m

```matlab
    temp_rnd0=i;
    %Random Election of Normal Nodes
    if (temp_rnd0>=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_rnd0<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=0:1:rmax
    r
  %Operation for epoch
  if(mod(r, round(1/p) )==0)
    for i=1:1:n
        S(i).G=0;
        S(i).cl=0;
    end
  end
hold off;
%Number of dead nodes
```

Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m

LeachCodeOnline.m | gsteb.m

```
hold off;
%Number of dead nodes
dead=0;
%Number of dead Advanced Nodes
dead_a=0;
%Number of dead Normal Nodes
dead_n=0;
%counter for bit transmitted to Bases Station and to Cluster Heads
packets_TO_BS=0;
packets_TO_CH=0;
%counter for bit transmitted to Bases Station and to Cluster Heads
PACKETS_TO_CH(r+1)=0;
PACKETS_TO_BS(r+1)=0;
figure(1);
for i=1:1:n
    %checking if there is a dead node
    if (S(i).E<=0)
        plot(S(i).xd,S(i).yd,'red .');
        dead=dead+1;
        if(S(i).ENERGY==1)
            dead_a=dead_a+1;
        end
        if(S(i).ENERGY==0)
            dead_n=dead_n+1;
        end
        hold on;
    end
    if S(i).E>0
        S(i).type='N';
        if (S(i).ENERGY==0)
        plot(S(i).xd,S(i).yd,'o');
        end
        if (S(i).ENERGY==1)
        plot(S(i).xd,S(i).yd,'+');
        end
        hold on;
    end
end
plot(S(n+1).xd,S(n+1).yd,'x');
STATISTICS(r+1).DEAD=dead;
DEAD(r+1)=dead;
DEAD_N(r+1)=dead_n;
DEAD_A(r+1)=dead_a;
%When the first node dies
```

Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m

LeachCodeOnline.m | gsteb.m

```
DEAD_A(r+1)=dead_a;
%When the first node dies
if (dead==1)
    if(flag_first_dead==0)
        first_dead=r
        flag_first_dead=1;
    end
end
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
 if(temp_rand<= (p/(1-p*mod(r,round(1/p)))))
            countCHs=countCHs+1;
            packets_TO_BS=packets_TO_BS+1;
            PACKETS_TO_BS(r+1)=packets_TO_BS;

            S(i).type='C';
            S(i).G=round(1/p)-1;
            C(cluster).xd=S(i).xd;
            C(cluster).yd=S(i).yd;
            plot(S(i).xd,S(i).yd,'k*');

            distance=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );
            C(cluster).distance=distance;
            C(cluster).id=i;
            X(cluster)=S(i).xd;
            Y(cluster)=S(i).yd;
            cluster=cluster+1;

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

    end
```

Editor - C:\Users\vikra\Downloads\LeachCodeOnline.m

LeachCodeOnline.m | gsteb.m

```
        end
      end
  end
  STATISTICS(r+1).CLUSTERHEADS=cluster-1;
  CLUSTERHS(r+1)=cluster-1;
  %Election of Associated Cluster Head for Normal Nodes
  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=1;
         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

         %Energy dissipated by associated Cluster Head
              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
          %Energy dissipated
          if(min_dis>0)
            S(C(min_dis_cluster).id).E = S(C(min_dis_cluster).id).E- ( (ERX + EDA)*4000 );
           PACKETS_TO_CH(r+1)=n-dead-cluster+1;
          end
         S(i).min_dis=min_dis;
         S(i).min_dis_cluster=min_dis_cluster;

      end
    end
  end
  hold on;
  countCHs;
  rcountCHs=rcountCHs+countCHs
  end
```

GSTEB

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

LeachCodeOnline.m | gsteb.m*

```
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=Etx=Erx
ETX=50*0.000000001;
ERX=50*0.000000001;
%Transmit Amplifier types
Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;
%Data Aggregation Energy
EDA=5*0.000000001;
%Values for Hetereogeneity
%Percentage of nodes than are advanced
m=0.1;
%\alpha
a=1;
%maximum number of rounds
rmax=400
%%%%%%%%%%%%%%%%%%%%%%%% END OF PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%
%Computation of do/
do=sqrt(Efs/Emp);
%Creation of the random Sensor Network
figure(1);
for i=1:1: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';

```

Editor - C:\Users\vikra\Downloads\gsteb.m

LeachCodeOnline.m | gsteb.m

```
        temp_rnd0=i;
        %Random Election of Normal Nodes
        if (temp_rnd0>=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_rnd0<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=0:1:rmax
        r
      %Operation for epoch
      if(mod(r, round(1/p) )==0)
        for i=1:1:n
            S(i).G=0;
            S(i).cl=0;
        end
      end
    hold off;
```

Editor - C:\Users\vikra\Downloads\gsteb.m

LeachCodeOnline.m | gsteb.m

```
    hold off;
    %Number of dead nodes
    dead=0;
    %Number of dead Advanced Nodes
    dead_a=0;
    %Number of dead Normal Nodes
    dead_n=0;
    %counter for bit transmitted to Bases Station and to Cluster Heads
    packets_TO_BS=0;
    packets_TO_CH=0;
    %counter for bit transmitted to Bases Station and to Cluster Heads
    PACKETS_TO_CH(r+1)=0;
    PACKETS_TO_BS(r+1)=0;
    figure(1);
    for i=1:1:n
        %checking if there is a dead node
        if (S(i).E<=0)
            plot(S(i).xd,S(i).yd,'red .');
            dead=dead+1;
            if(S(i).ENERGY==1)
                dead_a=dead_a+1;
            end
            if(S(i).ENERGY==0)
                dead_n=dead_n+1;
            end
            hold on;
        end
        if S(i).E>0
            S(i).type='N';
            if (S(i).ENERGY==0)
            plot(S(i).xd,S(i).yd,'o');
            end
            if (S(i).ENERGY==1)
            plot(S(i).xd,S(i).yd,'+');
            end
            hold on;
        end
    end
    plot(S(n+1).xd,S(n+1).yd,'x');
    STATISTICS(r+1).DEAD=dead;
    DEAD(r+1)=dead;
    DEAD_N(r+1)=dead_n;
    DEAD_A(r+1)=dead_a;
```

Editor - C:\Users\vikra\Downloads\gsteb.m

LeachCodeOnline.m | gsteb.m

```
DEAD_A(r+1)=dead_a;
%When the first node dies
if (dead==1)
    if(flag_first_dead==0)
        first_dead=r
        flag_first_dead=1;
    end
end
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
 if(temp_rand<= (p/(1-p*mod(r,round(1/p)))))
            countCHs=countCHs+1;
            packets_TO_BS=packets_TO_BS+1;
            PACKETS_TO_BS(r+1)=packets_TO_BS;

            S(i).type='C';
            S(i).G=round(1/p)-1;
            C(cluster).xd=S(i).xd;
            C(cluster).yd=S(i).yd;
            plot(S(i).xd,S(i).yd,'k*');

            distance=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );
            C(cluster).distance=distance;
            C(cluster).id=i;
            X(cluster)=S(i).xd;
            Y(cluster)=S(i).yd;
            cluster=cluster+1;

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

```

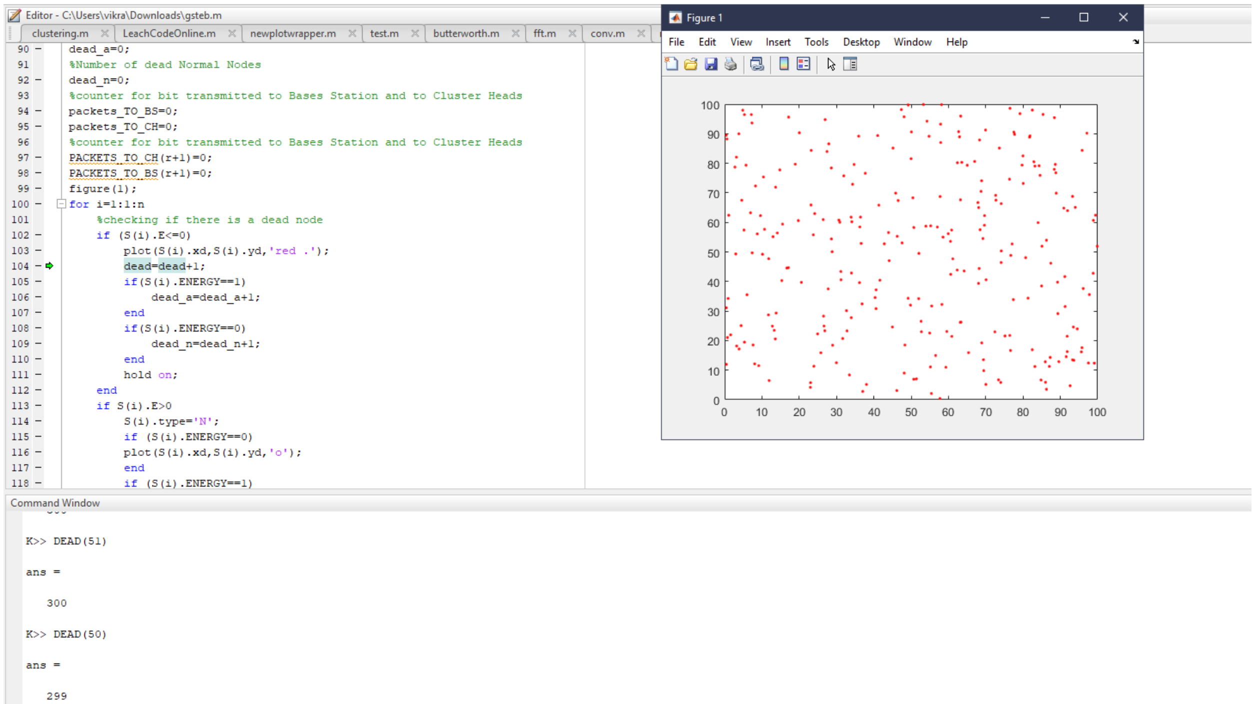
Editor - C:\Users\vikra\Downloads\gsteb.m

LeachCodeOnline.m | gsteb.m

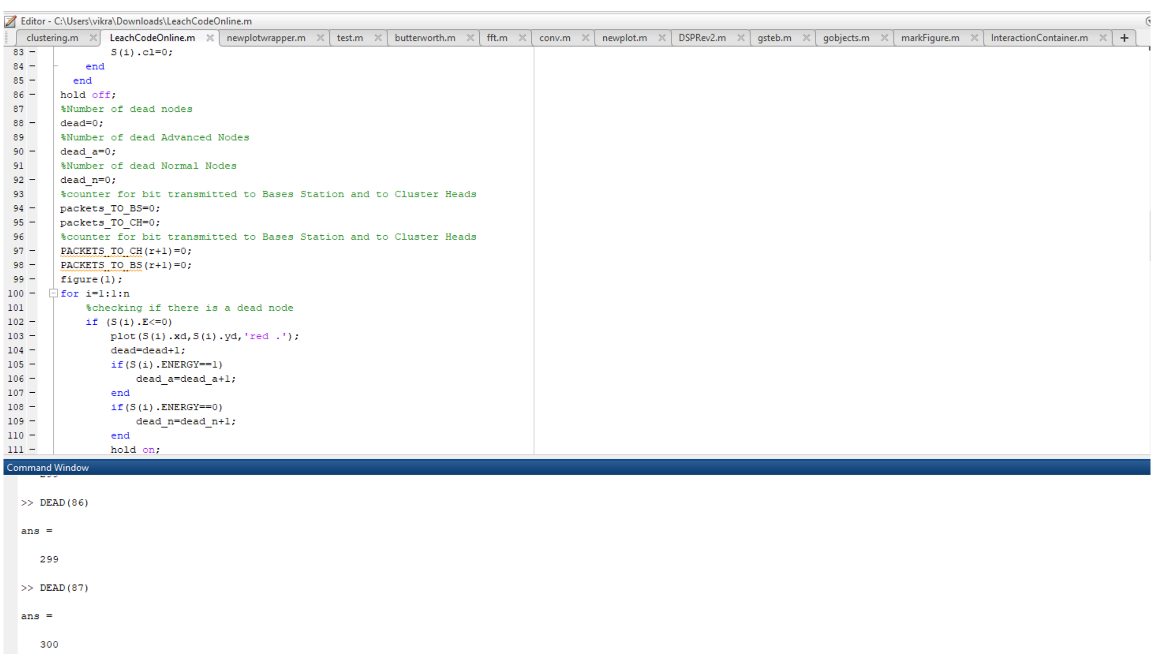
```
            Y(cluster)=S(i).yd;
            cluster=cluster+1;

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

    end
  end
end
STATISTICS(r+1).CLUSTERHEADS=cluster-1;
CLUSTERHS(r+1)=cluster-1;
%Election of Associated Cluster Head for Normal Nodes
for i=1:1:n
   if ( S(i).type=='N' && S(i).E>0 )
       min_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );

       %Energy dissipated by associated Cluster Head
            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
        %Energy dissipated
        if(min_dis>0)
          S(i).E = S(i).E- ( (ERX + EDA)*4000 );

        end
       S(i).min_dis=min_dis;

   end
end
hold on;

end
```

Output & Results

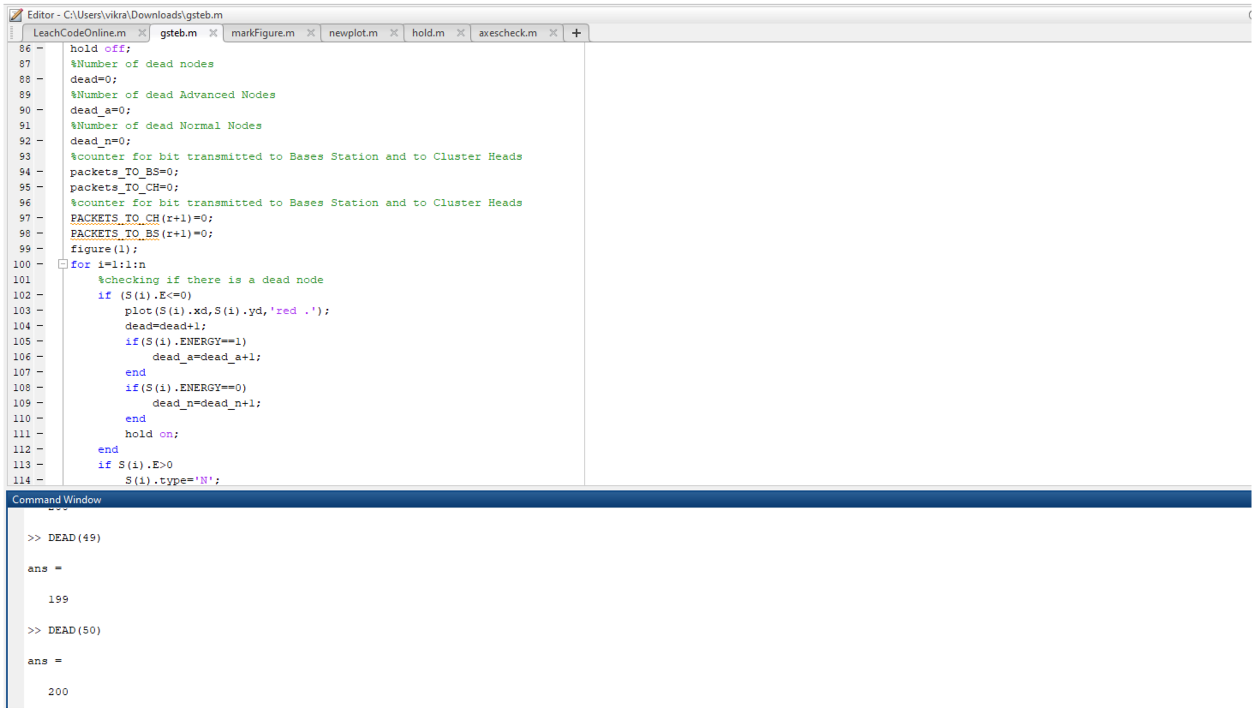
N=300, E_i = 0.01, GSTEB



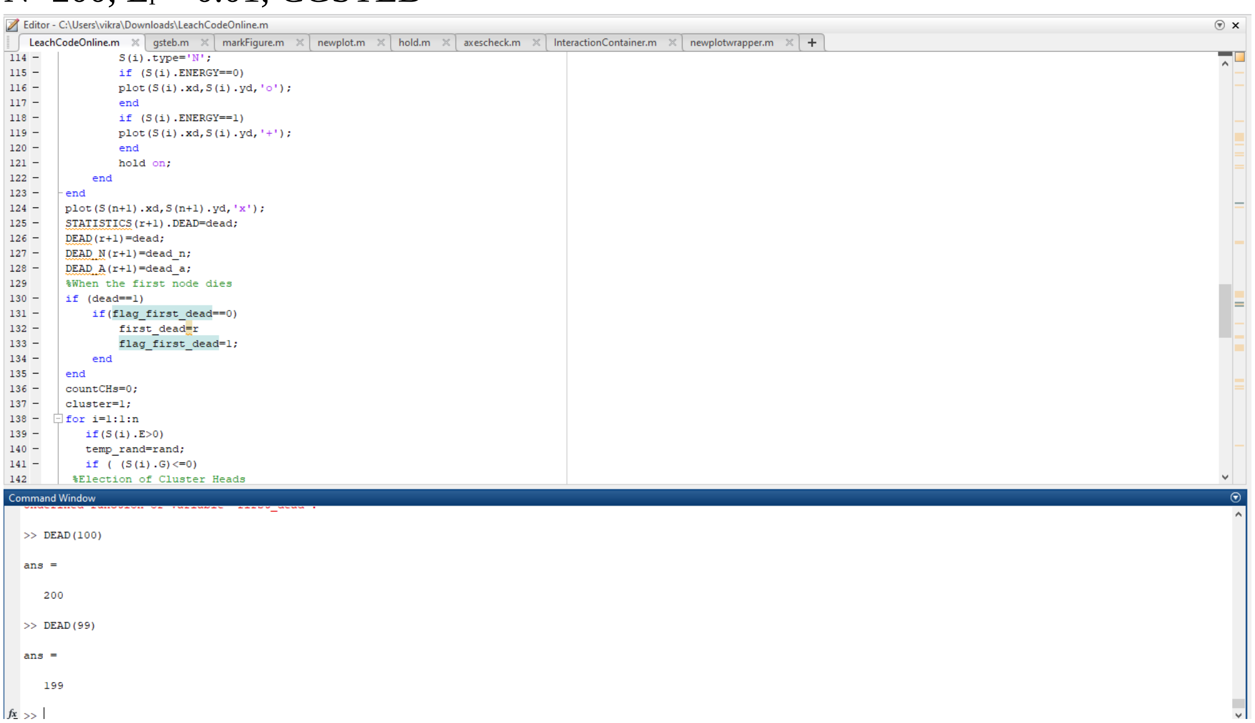
N=300, E_i = 0.01, CGSTEB



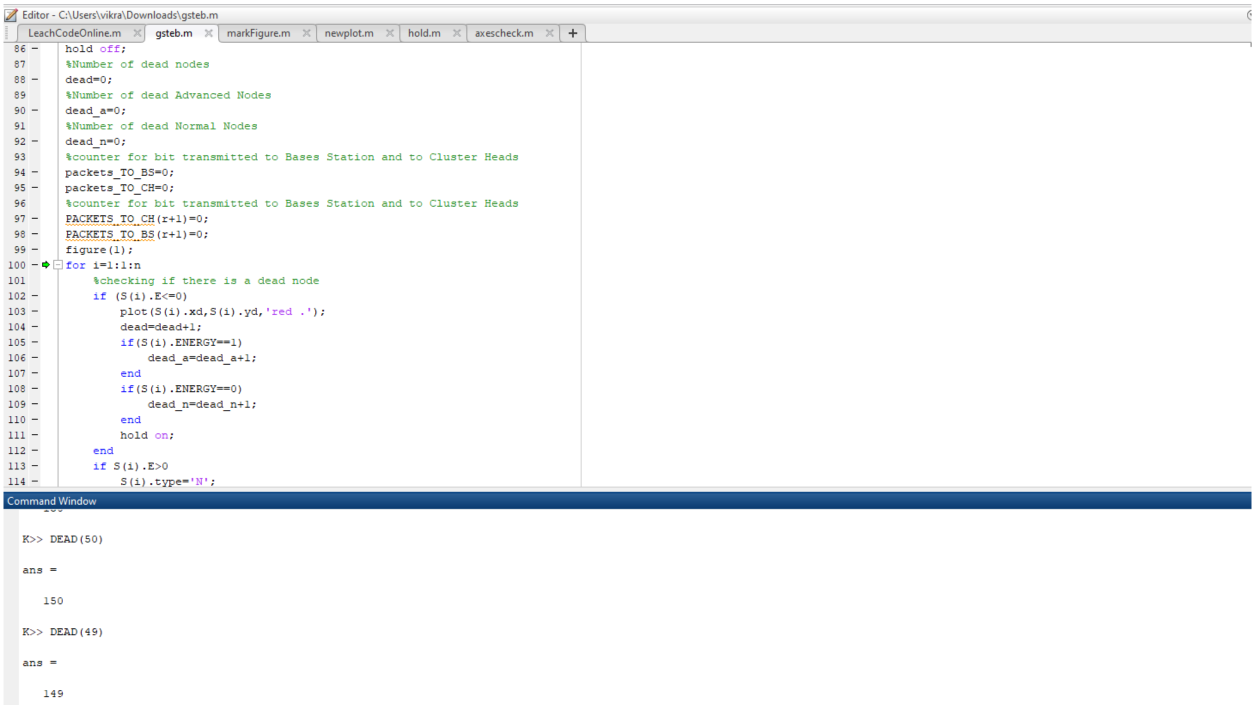
N=200, E_i = 0.01, GSTEB



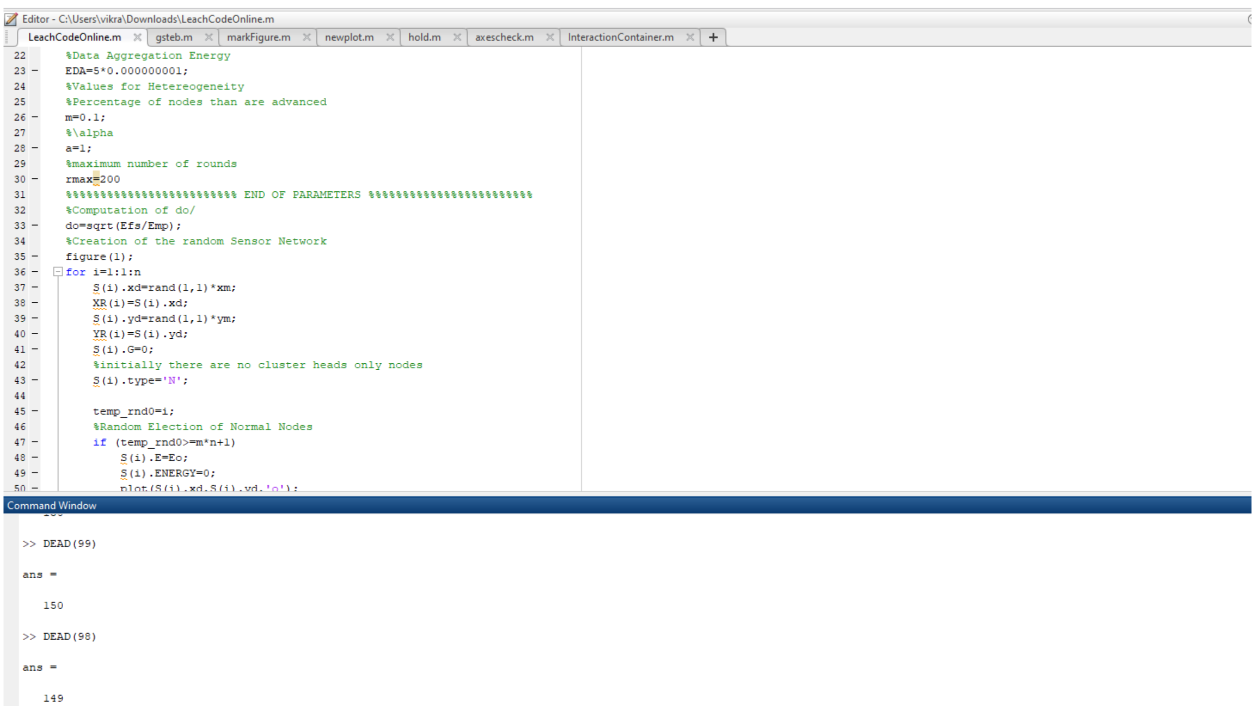
N=200, E_i = 0.01, CGSTEB



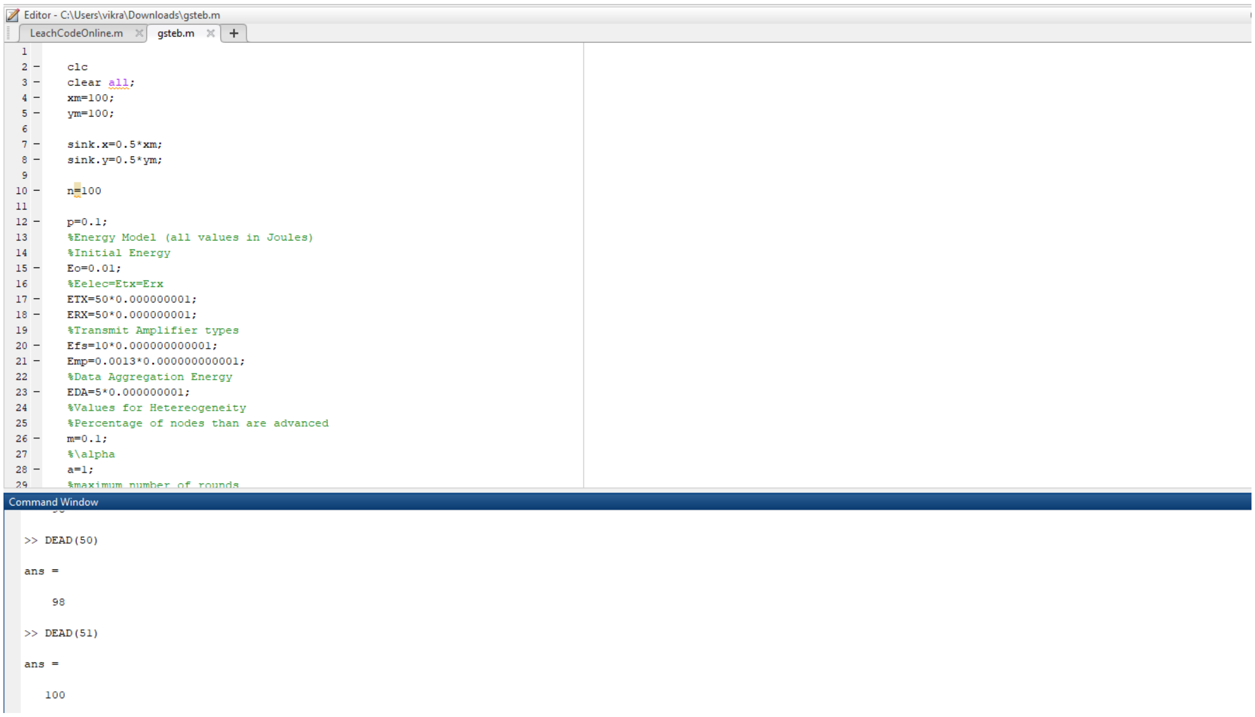
N=150, E_i = 0.01, GSTEB



N=150, E_i = 0.01, CGSTEB



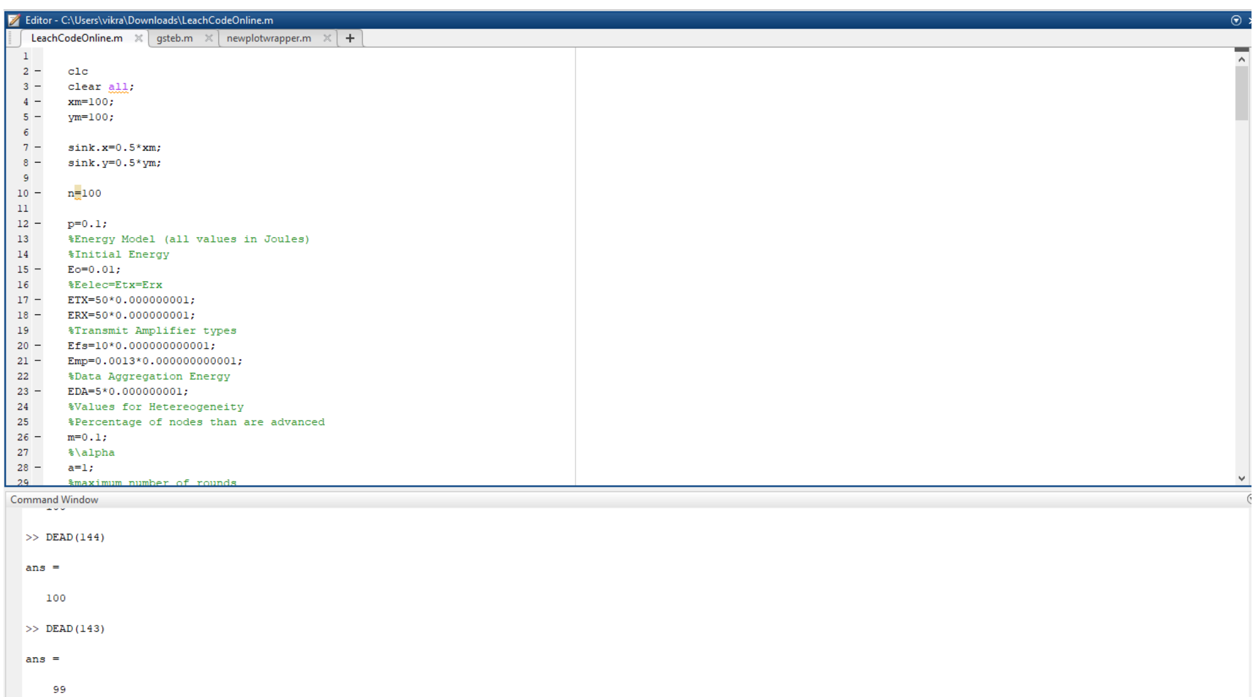
N=100, E_i = 0.01, GSTEB




```
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=Etx=Erx
ETX=50*0.000000001;
ERX=50*0.000000001;
%Transmit Amplifier types
Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;
%Data Aggregation Energy
EDA=5*0.000000001;
%Values for Hetereogeneity
%Percentage of nodes than are advanced
m=0.1;
%\alpha
a=1;
%maximum number of rounds
```

```
>> DEAD(50)

ans =

    98

>> DEAD(51)

ans =

   100
```

N=100, E_i = 0.01, CGSTEB

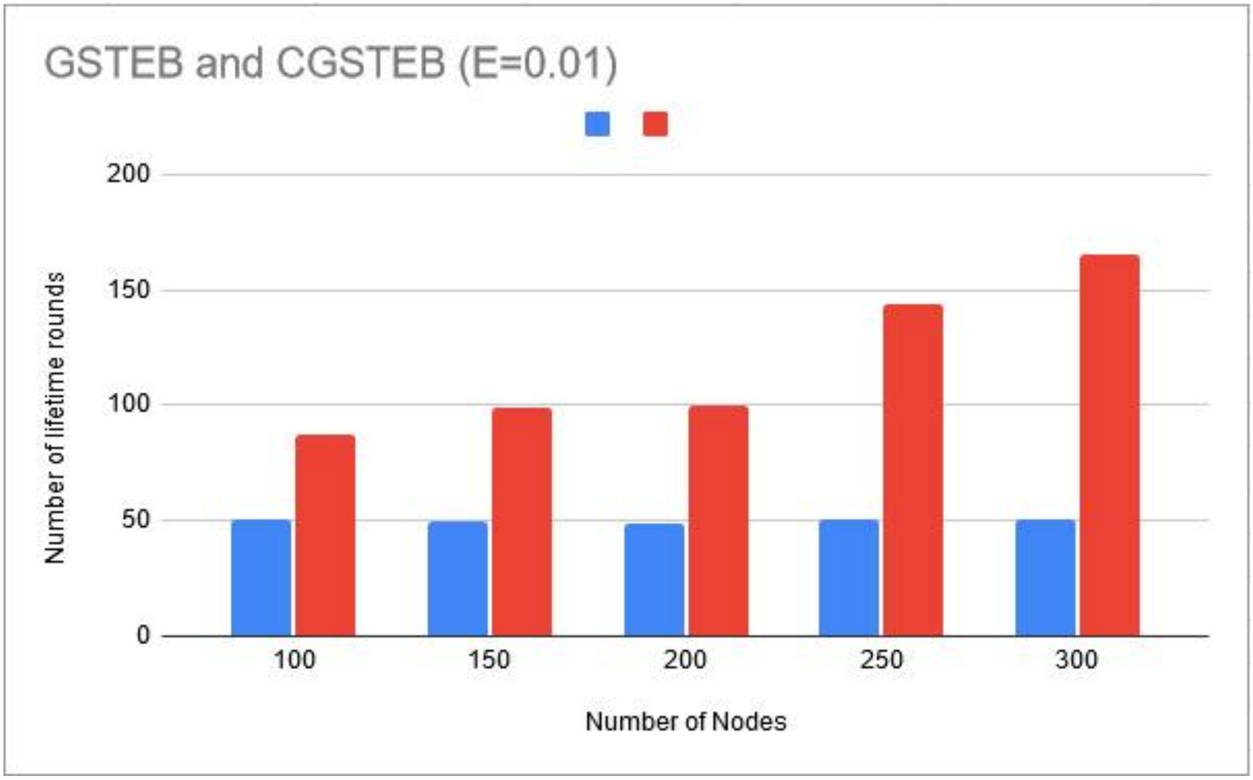


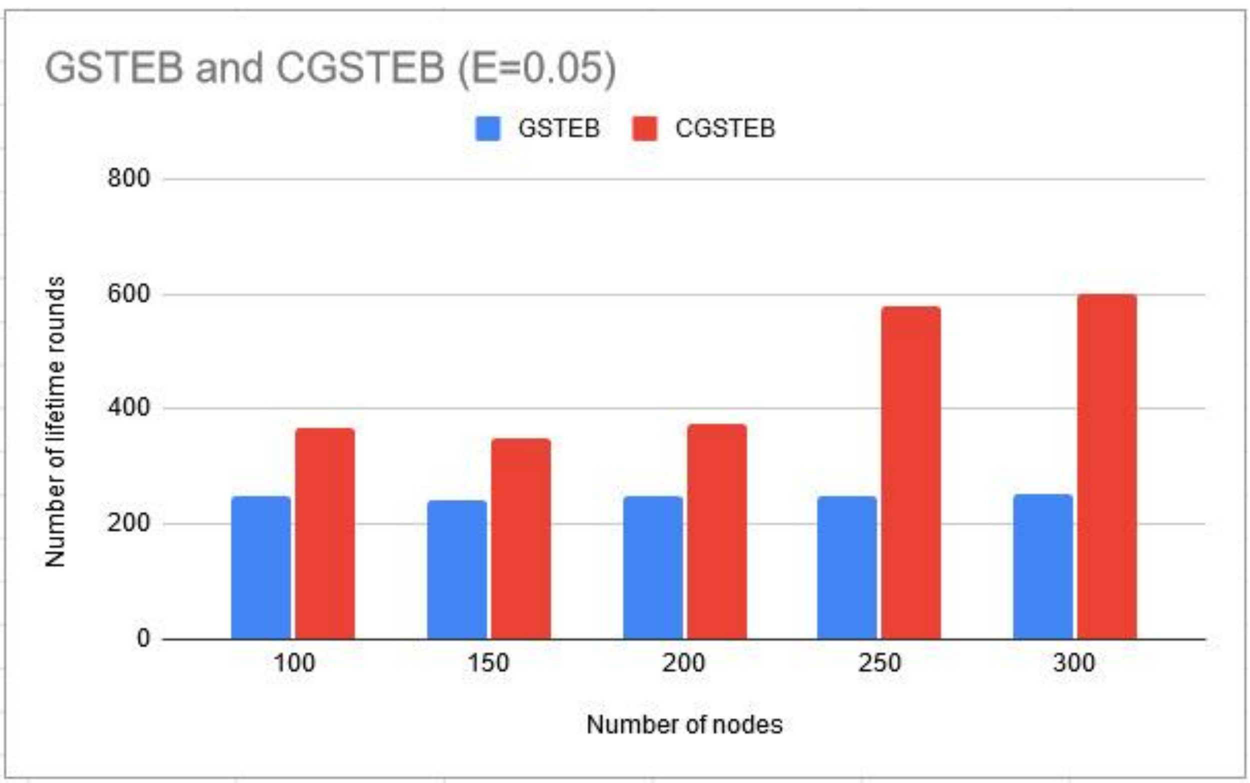

```
clc
clear all;
xm=100;
ym=100;

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=Etx=Erx
ETX=50*0.000000001;
ERX=50*0.000000001;
%Transmit Amplifier types
Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;
%Data Aggregation Energy
EDA=5*0.000000001;
%Values for Hetereogeneity
%Percentage of nodes than are advanced
m=0.1;
%\alpha
a=1;
%maximum number of rounds
```

```
>> DEAD(144)

ans =

   100

>> DEAD(143)

ans =

    99
```

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

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


GSTEB and CGSTEB (E=0.01)
200
150
100
50
0
Number of lifetime rounds
100
150
200
250
300
Number of Nodes


GSTEB and CGSTEB (E=0.05)
GSTEB
CGSTEB
800
600
400
200
0
Number of lifetime rounds
100
150
200
250
300
Number of nodes

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.