clear all

xm=100;

ym=100;

sink.x=0.5\*xm; %location of sink on x-axis

sink.y=0.25\*700; %location of sink on y-axis

n=100 %nodes

P=0.1; %probability of cluster heads

Eo=1; %initial energy

ETX=50\*0.000000001; %tx energy

ERX=50\*0.000000001; %rx energy

Efs=10\*0.000000000001; %free space loss

Emp=0.0013\*0.000000000001; %multi path loss

%Data Aggregation Energy

EDA=5\*0.000000001; %compression energy

a=0; %fraction of energy enhancment of advance nodes

rmax=7000 %maximum number of rounds

do=sqrt(Efs/Emp); %distance do is measured

Et=0; %variable just use below

A=0;

for i=1:1:n

S(i).xd=rand(1,1)\*xm; %generates a random no. use to randomly distibutes nodes on x axis

XR(i)=S(i).xd;

S(i).yd=rand(1,1)\*ym; %generates a random no. use to randomly distibutes nodes on y axis

YR(i)=S(i).yd;

S(i).G=0; %node is elegible to become cluster head

talha=rand\*a;

S(i).E=Eo\*(1+talha);

E(i)= S(i).E;

A=A+talha;

Et=Et+E(i); %estimating total energy of the network

%initially there are no cluster heads only nodes

S(i).type='N';

end

d1=0.765\*xm/2; %distance between cluster head and base station

K=sqrt(0.5\*n\*do/pi)\*xm/d1^2; %optimal no. of cluster heads

d2=xm/sqrt(2\*pi\*K); %distance between cluster members and cluster head

Er=1700\*(2\*n\*ETX+n\*EDA+K\*Emp\*d1^4+n\*Efs\*d2^2); %energy desipated in a round

S(n+1).xd=sink.x; %sink is a n+1 node, x-axis postion of a node

S(n+1).yd=sink.y; %sink is a n+1 node, y-axis postion of a node

countCHs=0; %variable, counts the cluster head

cluster=1; %cluster is initialized as 1

flag\_first\_dead=0; %flag tells the first node dead

flag\_teenth\_dead=0; %flag tells the 10th node dead

flag\_all\_dead=0; %flag tells all nodes dead

dead=0; %dead nodes count initialized to 0

first\_dead=0;

teenth\_dead=0;

all\_dead=0;

allive=n;

%counter for bit transmitted to Bases Station and to Cluster Heads

packets\_TO\_BS=0;

packets\_TO\_CH=0;

sum1=[];

for i=1:n

d12(i)=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );

sum1=sum1+d12(i);

end

d122=(sum1/n);

Nprox=min(d122, d12(i));

CH\_count=1;

P\_CH=0.1;

for r=0:1:rmax

r

if(mod(r, round(1/P) )==0)

for i=1:1:n

S(i).G=0;

S(i).cl=0;

end

end

Ea=Et\*(1-r/rmax)/n;

dead=0;

for i=1:1:n

if (S(i).E<=0)

dead=dead+1;

if (dead==1)

if(flag\_first\_dead==0)

first\_dead=r;

flag\_first\_dead=1;

end

end

if(dead==0.1\*n)

if(flag\_teenth\_dead==0)

teenth\_dead=r;

flag\_teenth\_dead=1;

end

end

if(dead==n)

if(flag\_all\_dead==0)

all\_dead=r;

flag\_all\_dead=1;

end

end

end

if S(i).E>0

S(i).type='N';

end

sume=0;

for i=1:1:n

if S(i).E>0

sume=sume+S(i).E;

end

end

Er=ceil(sume/n);

Relay\_mx=max(sume, S(i).E);

end

STATISTICS.DEAD(r+1)=dead;

STATISTICS.ALLIVE(r+1)=allive-dead;

countCHs=0;

cluster=1;

for i=1:1:n

if Ea>0

p(i)=P\*n\*(1+a)\*E(i)/(n+A)\*(Ea);

%p(i)=P\*n\*S(i).E\*E(i)/(Et\*Ea);

if(S(i).E>0)

temp\_rand=rand;

if ( (S(i).G)<=0)

if(temp\_rand<= (p(i)/(1-p(i)\*mod(r,round(1/p(i))))))

if CH\_count<P\_CH

%%CH selection

if(temp\_rand<= (p(i)\*ceil(S(i).E)\*Nprox\*Relay\_mx/(1-p(i)\*mod(r,round(1/p(i))))))

end

end

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(i))-1;

C(cluster).xd=S(i).xd;

C(cluster).yd=S(i).yd;

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;

distance;

if (distance>do)

S(i).E=S(i).E- ( (ETX+EDA)\*(1700) + Emp\*1700\*( distance\*distance\*distance\*distance ));

end

if (distance<=do)

S(i).E=S(i).E- ( (ETX+EDA)\*(1700) + Efs\*1700\*( distance \* distance ));

end

end

end

end

end

end

STATISTICS.COUNTCHS(r+1)=countCHs;

%(5)??????????(???????)

%???????????????????

for i=1:1:n

if ( S(i).type=='N' && S(i).E>0 )

if(cluster-1>=1)

min\_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );

min\_dis\_cluster=0;

for c=1:1:cluster-1

temp=min(min\_dis,sqrt( (S(i).xd-C(c).xd)^2 + (S(i).yd-C(c).yd)^2 ) );

if ( temp<min\_dis )

min\_dis=temp;

min\_dis\_cluster=c;

end

end

%???????1700bit???????

if(min\_dis\_cluster~=0)

min\_dis;

if (min\_dis>do)

S(i).E=S(i).E- ( ETX\*(1700) + Emp\*1700\*( min\_dis \* min\_dis \* min\_dis \* min\_dis));

end

if (min\_dis<=do)

S(i).E=S(i).E- ( ETX\*(1700) + Efs\*1700\*( min\_dis \* min\_dis));

end

S(C(min\_dis\_cluster).id).E = S(C(min\_dis\_cluster).id).E- ( (ERX + EDA)\*1700 );

packets\_TO\_CH=packets\_TO\_CH+1;

else

min\_dis;

if (min\_dis>do)

S(i).E=S(i).E- ( ETX\*(1700) + Emp\*1700\*( min\_dis \* min\_dis \* min\_dis \* min\_dis));

end

if (min\_dis<=do)

S(i).E=S(i).E- ( ETX\*(1700) + Efs\*1700\*( min\_dis \* min\_dis));

end

packets\_TO\_BS=packets\_TO\_BS+1;

end

S(i).min\_dis=min\_dis;

S(i).min\_dis\_cluster=min\_dis\_cluster;

else

min\_dis=sqrt( (S(i).xd-S(n+1).xd)^2 + (S(i).yd-S(n+1).yd)^2 );

if (min\_dis>do)

S(i).E=S(i).E- ( ETX\*(1700) + Emp\*1700\*( min\_dis \* min\_dis \* min\_dis \* min\_dis));

end

if (min\_dis<=do)

S(i).E=S(i).E- ( ETX\*(1700) + Efs\*1700\*( min\_dis \* min\_dis));

end

packets\_TO\_BS=packets\_TO\_BS+1;

end

end

end

STATISTICS.PACKETS\_TO\_CH(r+1)=packets\_TO\_CH;

STATISTICS.PACKETS\_TO\_BS(r+1)=packets\_TO\_BS;

if (dead==n)

break

end

end

first\_dead

teenth\_dead

all\_dead

STATISTICS.DEAD(r+1)

STATISTICS.ALLIVE(r+1)

STATISTICS.PACKETS\_TO\_CH(r+1)

STATISTICS.PACKETS\_TO\_BS(r+1)

STATISTICS.COUNTCHS(r+1)

figure(1)

plot(STATISTICS.ALLIVE, 'r')

hold on

axis([0 5500 0 100]);

xlabel('Number of rounds')

ylabel('Number of Alive nodes')

% legend('CAFL', 'FBECS','Proposed')

grid on;

load Jai

title('Network Lifetime for N1 case ')

figure(2)

plot(pack\_p(:,1), pack\_p(:,2),'r')

hold on

axis([0 3000 0 20000]);

xlabel('Number of rounds')

ylabel('Packets to BS')

% legend('CAFL', 'FBECS','Proposed')

grid on;