MOBILE NETWORK PERVASIVE COMPUTING (MULTIHOP ROUTING – LEACH)

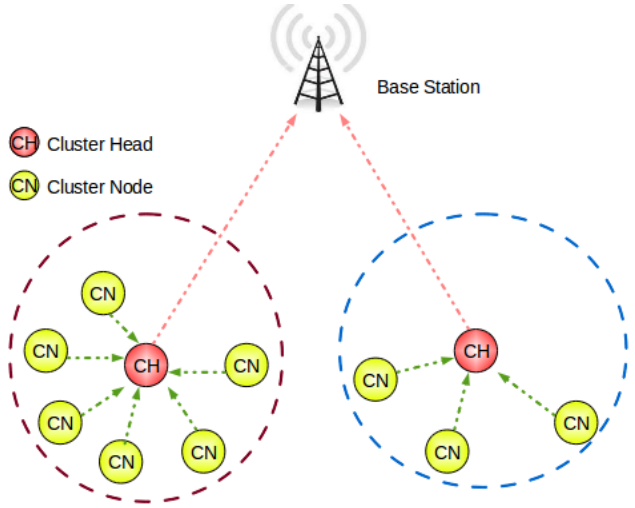
Mochammad Zen Samsono Hadi, ST. MSc. Ph.D

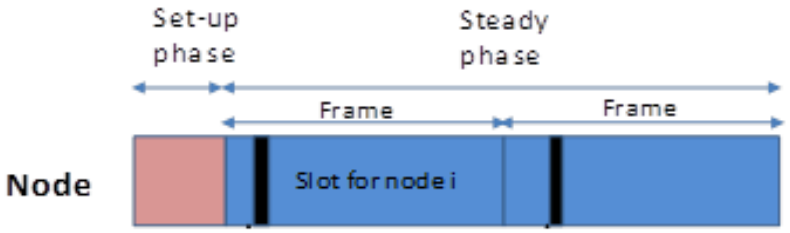
Discussion

- LEACH Protocol
- Code explanation
- Performance Evaluation

LEACH: Low-Energy Adaptive Clustering Hierarchy

- LEACH protocol use de-centralized method to determine cluster.
- LEACH-C protocol use centralized method when determining cluster. The cluster formation is done by Base Station.

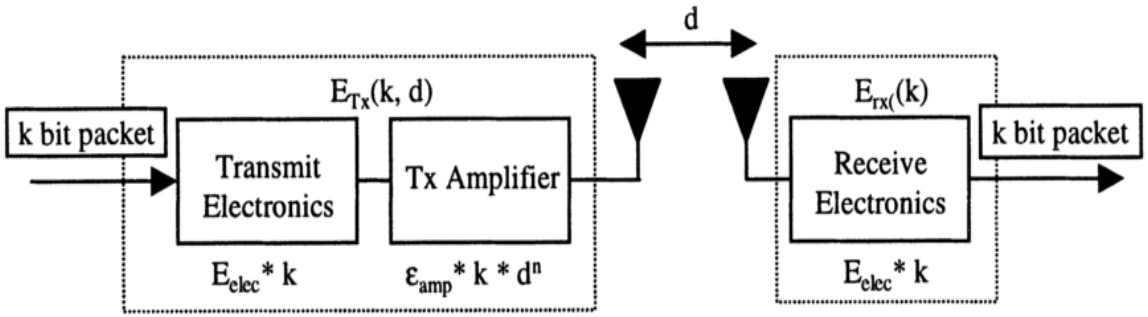




Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An application specific protocol architecture for wireless microsensor networks, IEEE Trans. Wirel. Commun., 2002, 1, (4), pp. 660–670

Energy Model (1/3)

Radio energy dissipation model



Transmitter energy

$$E_{TX}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d)$$

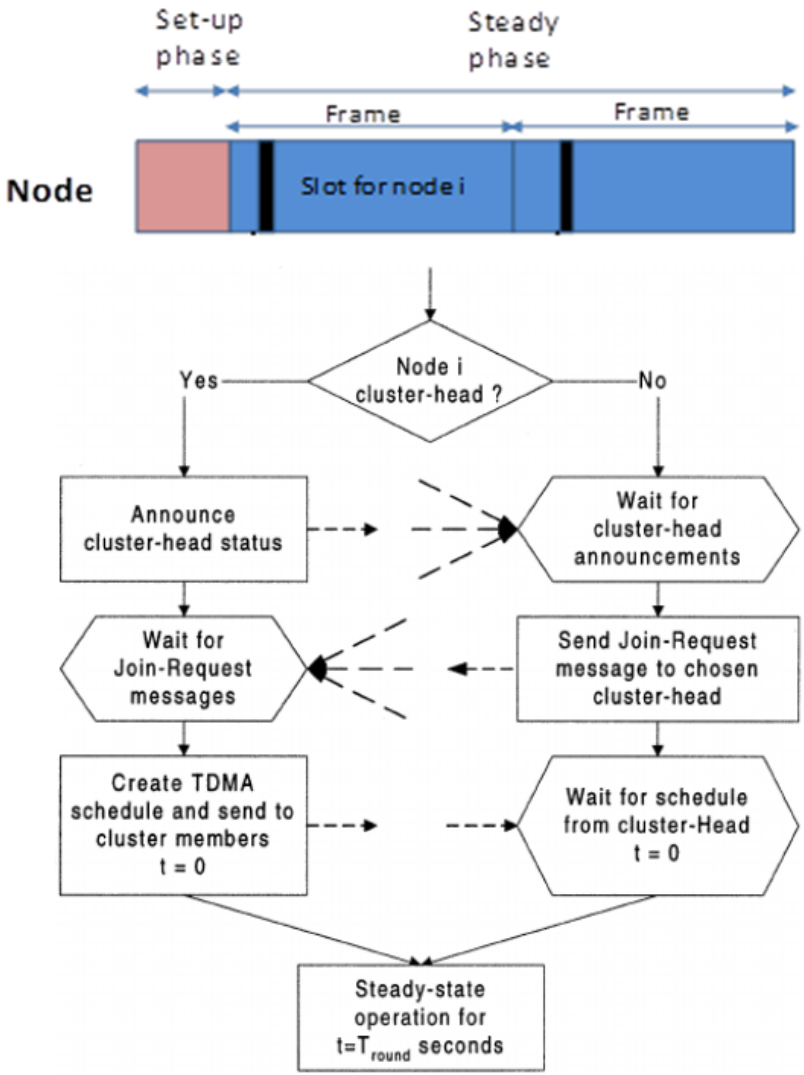
$$E_{TX}(k,d) = \begin{cases} k.E_{elec} + k.\varepsilon_{fs}.d^2, & d < d_{crossover}\ (Free\ Space) \\ k.E_{elec} + k.\varepsilon_{mp}.d^4, & d \geq d_{crossover}\ (Multipath) \end{cases}$$

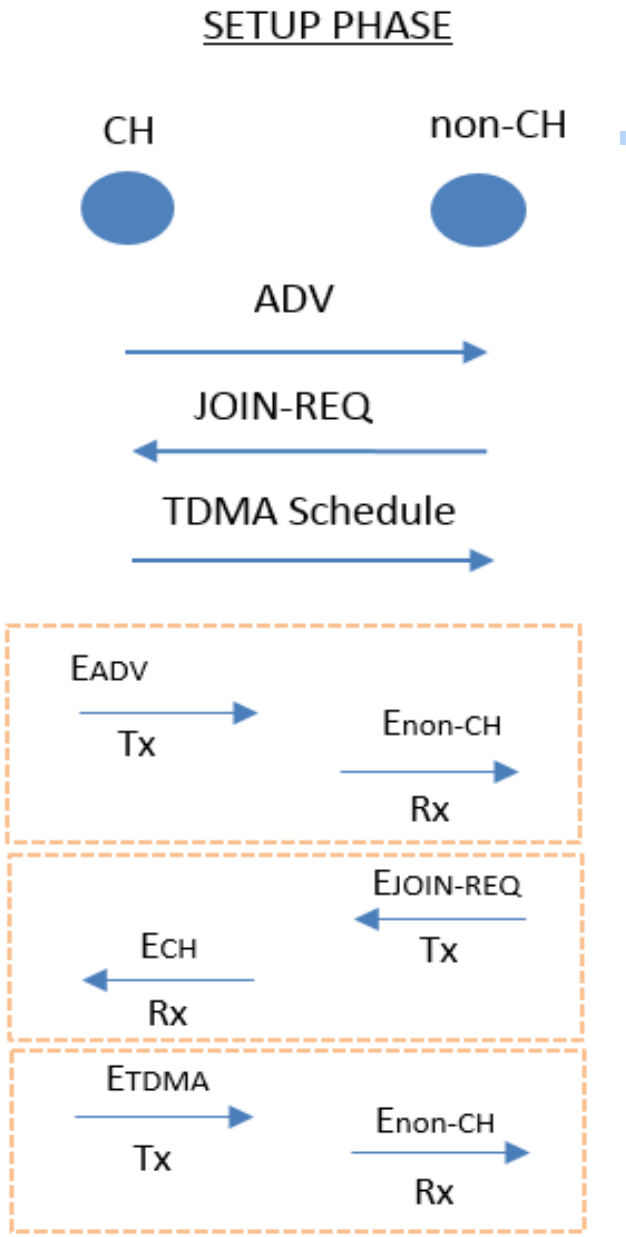
Receiver energy

$$E_{RX}(k) = E_{RX-elec}(k) = k\,.E_{elec}$$

Energy Model (2/3)

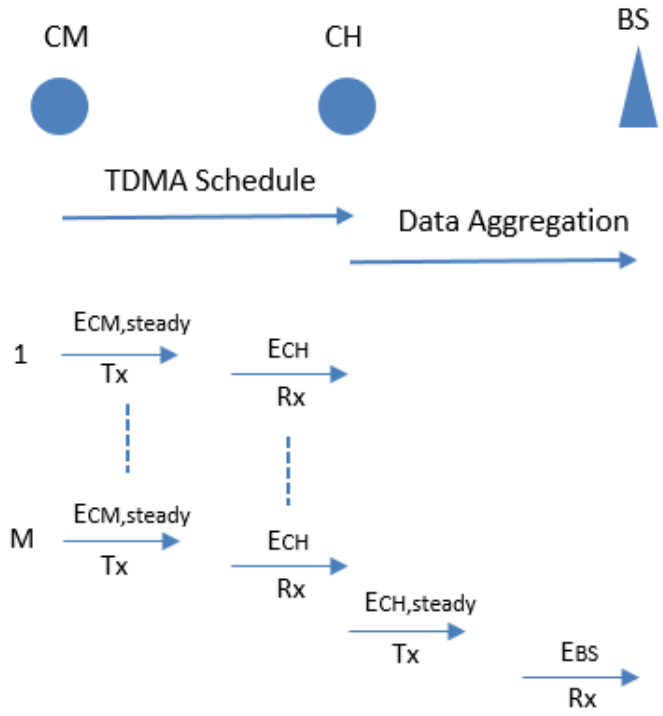
Energy dissipation for setup





Energy Model (3/3)

STEADY PHASE



Energy dissipation for steady

$$E_{CM,steady} = p \,.\, E_{elec} + p \,.\, D_{toCH}$$

$$E_{CH,steady} = E_{RX} + E_{DA} + E_{TX}$$

$$E_{CH,steady} = \left(\frac{N}{k} - 1\right).p \,.\, E_{elec} + \frac{N}{k} \,.p \,.\, E_{DA} + p \,.\, E_{elec} + p \,.\, D_{toBS}$$

where $$D_{toBS} = \begin{cases} \varepsilon_{fs} d^2, & d < d_0 \\ \varepsilon_{mp} d^4, & d \geq d_0 \end{cases}$$

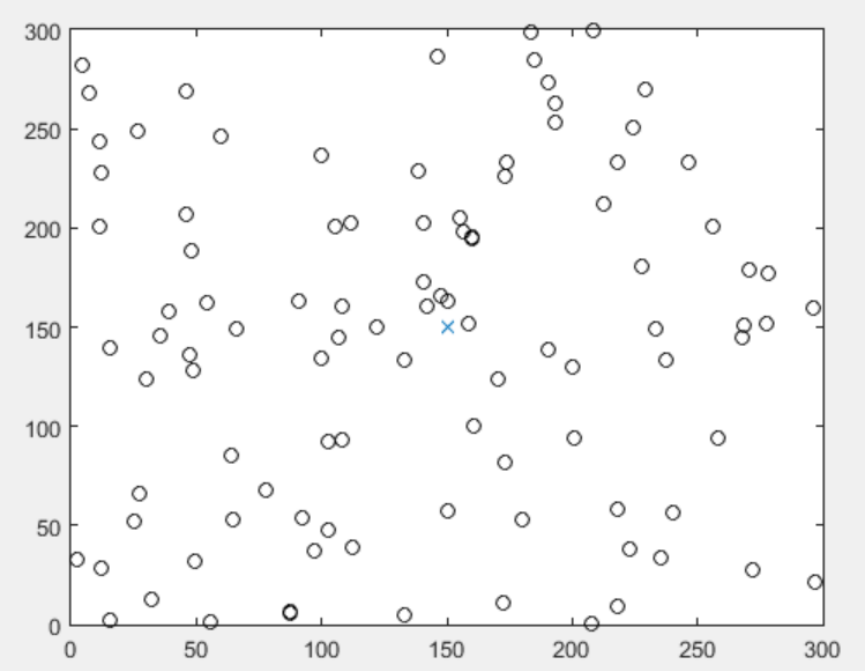
$$E_{steady} = \left(\frac{N}{k} - 1\right) . E_{CM,steady} + E_{CH,steady}$$

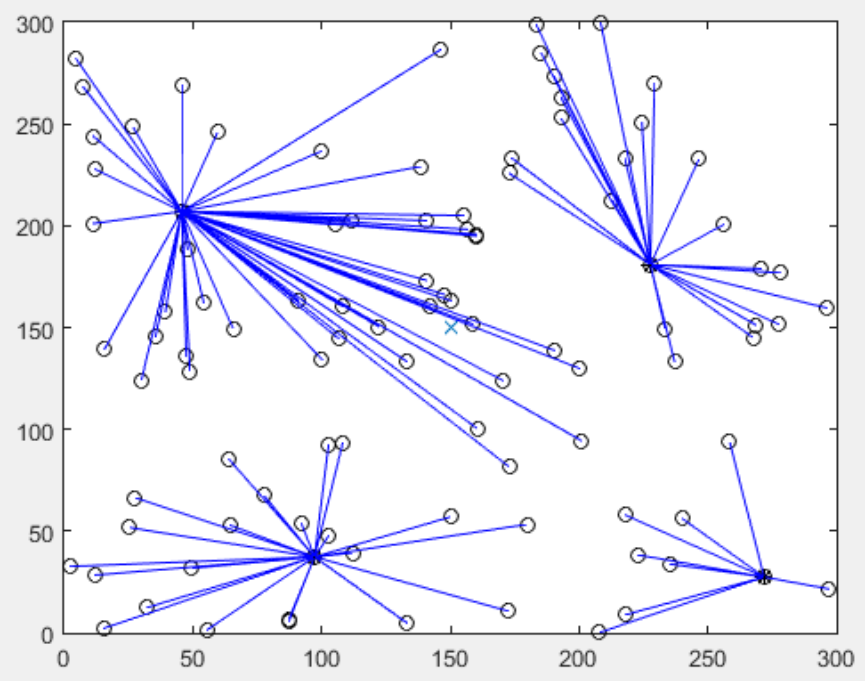
Simulation Parameters

Parameter	Value
Size of network	300x300m^2
Size of packet	500 bytes
E_{elec} (Radio Electronics Energy)	50 nJ/bit
E_{amp} (Radio Amplifier Radio)	100 pJ/bit/m^2
E_{init} (Initial Energy)	0.5 J
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^2
E_{DA}	5 nJ/bit/signal
Number of nodes	100
Location BS	Center network area

1. Program: Nodes & Cluster

- Run: LEACHcluster.m with probability of cluster heads (CH) = 0.05





2. Program: Parameters

- Run: LEACHprocess.m

```
xm=300;
ym=300;
sink.x=0.5*xm;
sink.y=0.5*ym;
n=100;
p=0.05;
Eo=0.5;
ETX=50*0.000000001;
ERX=50*0.000000001;
Efs=10e-12;
Emp=0.0013e-12;
EDA=5*0.000000001;
rmax=2500;
do=sqrt(Efs/Emp);
Et=0;
```

Network area

Sink node at center network

Probability of CH = 0.05

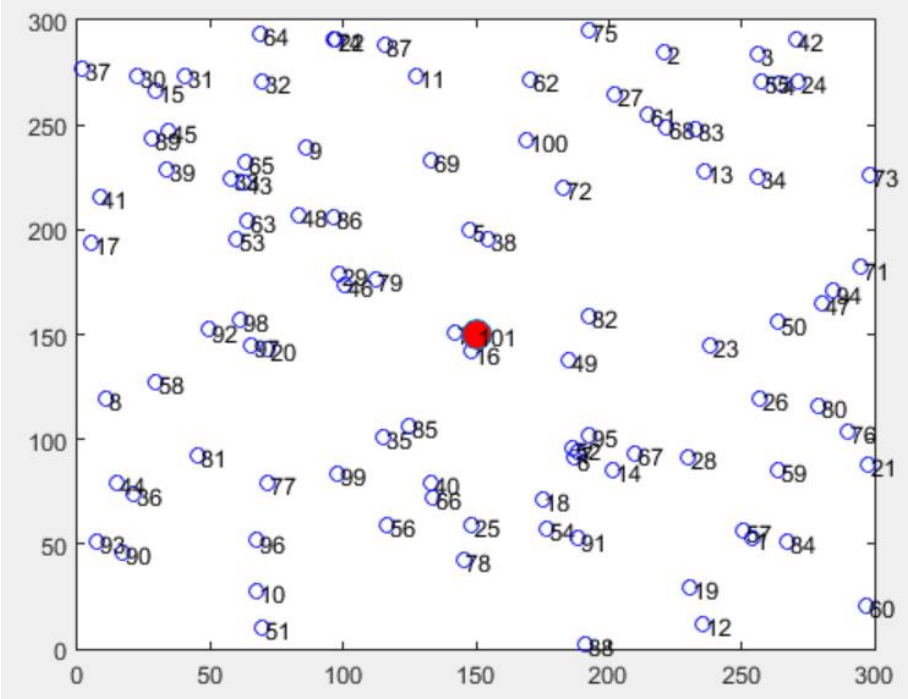
Initial Energy

Energy for Tx and Rx

Number of round

3. Program: Create Nodes

```
%Creation of the random Sensor Network
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;
    distance=sqrt( (S(i).xd-(S(n+1).xd) )^2 + (S(i).yd-(S(n+1).yd) )^2 );
    S(i).distance=distance;
    S(i).G=0;
    %initially there are no cluster heads only nodes
    S(i).type='N';
    S(i).E=Eo;
    Et=Et+S(i).E;
    figure(h*10)
      plot(S(i).xd,S(i).yd,'bo');
      text(S(i).xd+1,S(i).yd-0.5,num2str(i));
      hold on;
end
```

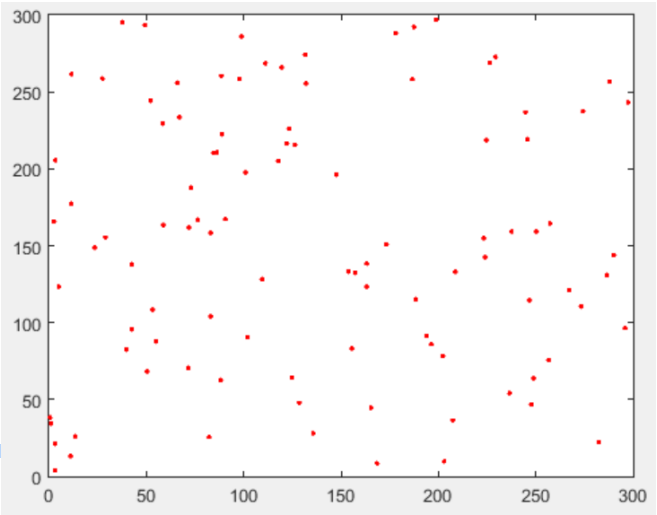


4. Program: Checking Dead Nodes

```
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 (dead==1)
          if(flag_first_dead==0)
              first_dead=r;
              flag_first_dead=1;
          end
        end
        if(dead==0.5*n)
          if(flag_half_dead==0)
              half_dead=r;
              flag_half_dead=1;
          end
        end
        if(dead==n)
          if(flag_all_dead==0)
              all_dead=r;
              flag_all_dead=1;
          end
        end

        %hold on;
    end
    if S(i).E>0
        S(i).type='N';
    end
end
```



5. Program: Electing of CHs (1/2)

```
for i=1:1:n
    if(S(i).E>0)
        temp_rand=rand;
        if ( (S(i).G)<=0)

            %Election of Cluster Heads for normal nodes
            if ( temp_rand <= ( p/ ( 1 - p * mod(r,round(1/p)) )) )

                countCHs=countCHs+1;
                packets_TO_BS=packets_TO_BS+1;
                packets_TO_BS_per_round=packets_TO_BS_per_round+1;
                PACKETS_TO_BS(r+1)=packets_TO_BS;


                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*');
```

Threshold value:
p = k/N
k = no of possible CHs
N = no of nodes

5. Program: Electing of CHs (2/2)

Calculate distance to BS

```
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
```

6. Program: Election of Cluster Members (CM) (1/2)

Calculate min distance to CH

```
%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=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
```

6. Program: Election of Cluster Members (CM) (2/2)

```
%Energy dissipated by associated Cluster Head by CM
if(min_dis_cluster~=0)
     min_dis;
     if (min_dis>do)
         S(i).E=S(i).E- ( ETX*(4000) + Emp*4000*( min_dis * min_dis * min_dis * min_dis));
     end
     if (min_dis<=do)
         S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*( min_dis * min_dis));
     end

     S(C(min_dis_cluster).id).E = S(C(min_dis_cluster).id).E- ( (ERX + EDA)*4000 );
     packets_TO_CH=packets_TO_CH+1;
else
     min_dis;
     if (min_dis>do)
         S(i).E=S(i).E- ( ETX*(4000) + Emp*4000*( min_dis * min_dis * min_dis * min_dis));
     end
     if (min_dis<=do)
         S(i).E=S(i).E- ( ETX*(4000) + Efs*4000*( min_dis * min_dis));
     end
     packets_TO_BS=packets_TO_BS+1;
     packets_TO_BS_per_round=packets_TO_BS_per_round+1;
     PACKETS_TO_BS(r+1)=packets_TO_BS;
end
 S(i).min_dis=min_dis;
S(i).min_dis_cluster=min_dis_cluster;
```

CM connects to CH

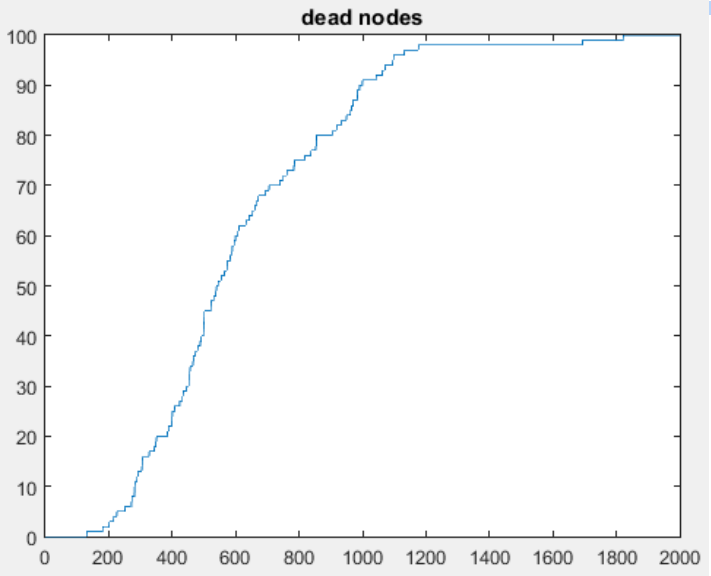
CM connects to BS

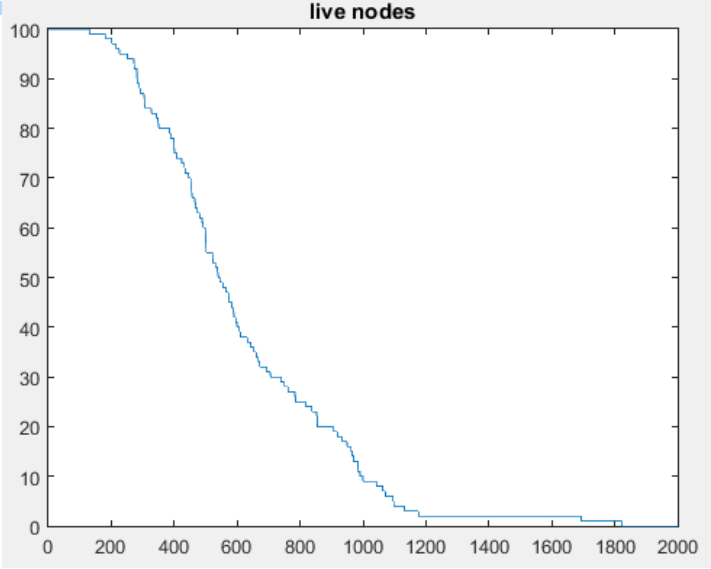
7. Program: Performance Evaluation

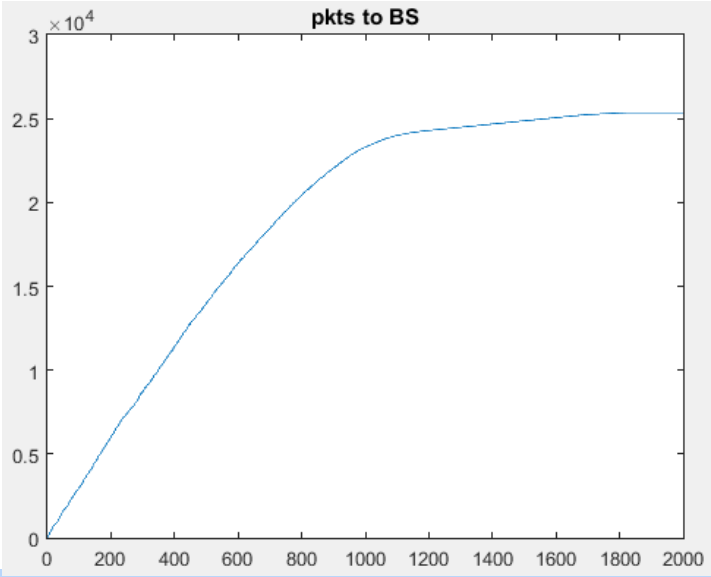
- Analyze for:
  - ✓Dead nodes
  - ✓Node alives
  - ✓Number of packets received by CH
  - ✓Number of packets received by BS
  - ✓Number of packets received by BS per round
  - ✓Throughput
  - ✓Number of CHs per round
  - ✓Remaining energy

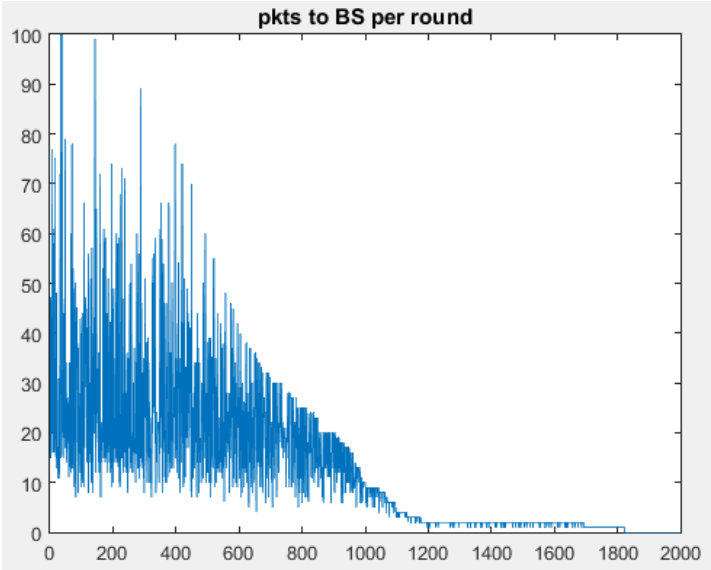
```
for r=0:rmax
    STATISTICS.DEAD(h+1,r+1)=sum(STATISTICS.DEAD(:,r+1))/h;
    STATISTICS.ALLIVE(h+1,r+1)=sum(STATISTICS.ALLIVE(:,r+1))/h;
    STATISTICS.PACKETS_TO_CH(h+1,r+1)=sum(STATISTICS.PACKETS_TO_CH(:,r+1))/h;
    STATISTICS.PACKETS_TO_BS(h+1,r+1)=sum(STATISTICS.PACKETS_TO_BS(:,r+1))/h;
   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

Performance Evaluation

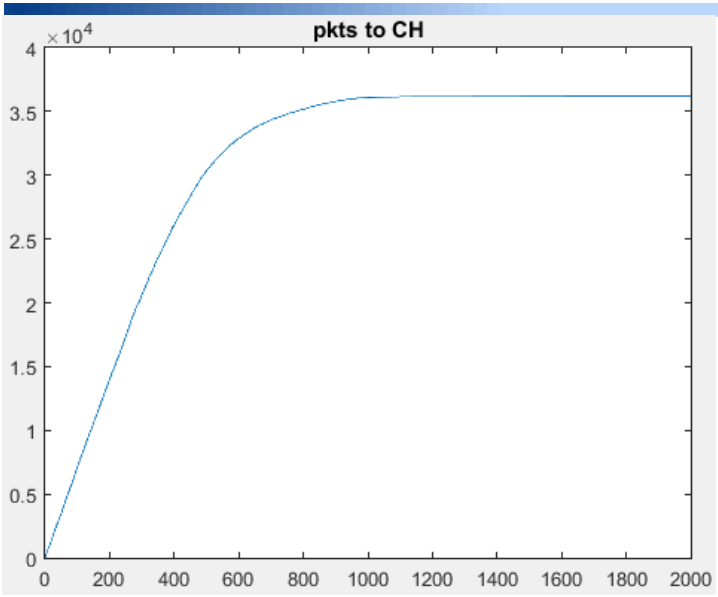


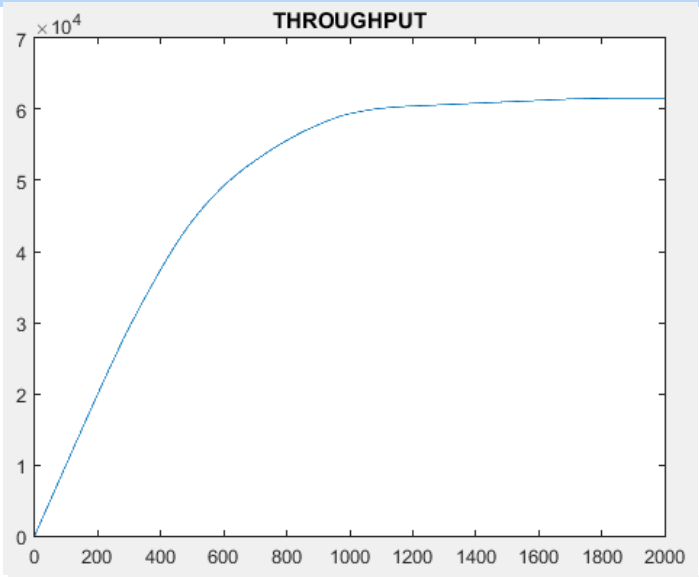


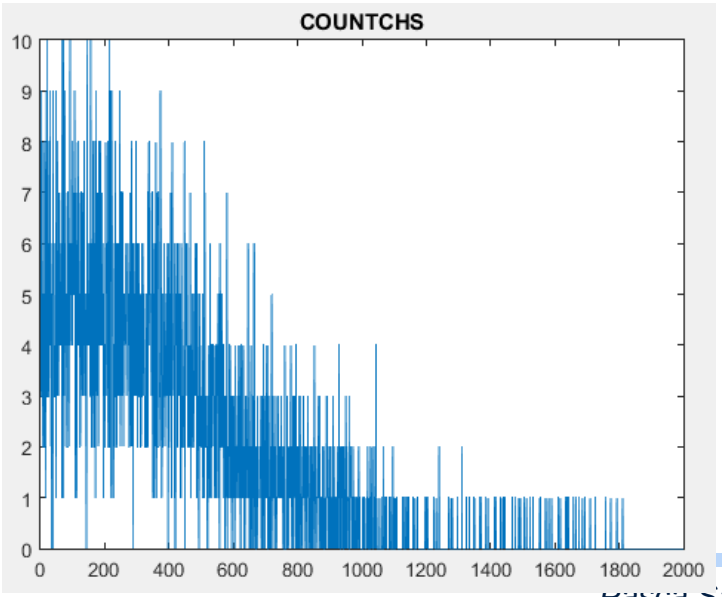


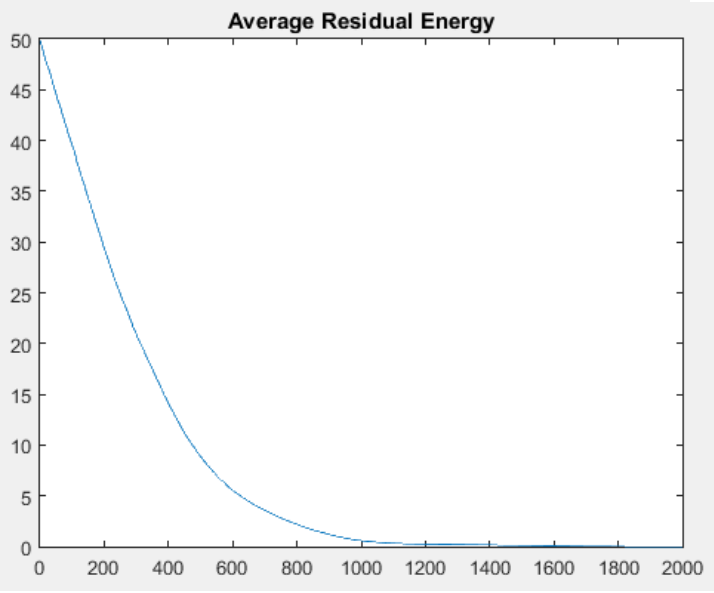


Performance Evaluation









TASK

- Change the probability of CHs become 0.01, 0.03, 0.04, 0.07 and 0.07
- Compare the:
  - ✓ Dead nodes
  - ✓ Live nodes
  - ✓ Throughput
  - ✓ Energy
- Give mobility function to the nodes and analyze the performance