International Journal of Applied Research 2017; 3(7): 379-384

ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor: 5.2
IJAR 2017; 3(7): 379-384
www.allresearchjournal.com
Received: 06-05-2017
Accepted: 07-06-2017

Ashish
M. Tech Student in E.C.E., IIET (Jind) under Kurukshetra University, India

Sunaina
Asstt. Prof. in E.C.E., IIET (Jind) under Kurukshetra University, India

Correspondence
Ashish
M. Tech Student in E.C.E., IIET (Jind) under Kurukshetra University, India

Implementation of DEEC Protocol for Clustering in Heterogeneous WSN

Ashish and Sunaina

Abstract
The clustering is a technique that is used to reduce the consumption of energy. It helps in increasing the scalability and lifetime of the network. Many energy-efficient clustering protocols have been designed for the characteristics of homogeneous and heterogeneous wireless sensor networks. We propose and evaluate a new distributed energy-efficient clustering scheme for heterogeneous wireless sensor networks, which is called DEEC. In DEEC, the cluster-heads are elected by a probability based on the ratio between residual energy of each node and the average energy of the network. The epochs of being cluster-heads for nodes are different according to their initial and residual energy. The nodes with high initial and residual energy will have more chances to be the cluster-heads than the nodes with low energy.

Keywords: Wireless Sensor Network, Heterogeneous environment, Energy-efficient, Cluster-head.

1. Introduction
WSN consists of a large number of sensor nodes, moreover these sensor nodes run on non-rechargeable batteries. So to serve the objective of fault-tolerance, load balancing and network connectivity, grouping of nodes is required. Clustering is a process of dividing sensor nodes into a number of groups and selecting a group head from each group. The groups are called clusters and group heads are called Cluster Heads (CHs) of the clusters. Guidelines for creating the clusters include distance between cluster head and its member, intra-cluster communication cost, residual energy of sensor nodes, location of node with respect to BS etc. There are various kinds of clustering protocols used in wireless sensor networks. Some of them are Distributed Energy-Efficient clustering (DEEC) Protocol, Power Efficient GAthering in Sensor Information Systems (PEGASIS) Protocol, Threshold Sensitive Energy Efficient Sensor Network (TEEN) Protocol, Adaptive Threshold Sensitive Energy Efficient Sensor Network (APTEEN) Protocol, Hybrid Energy Efficient Distributed (HEED) Clustering Protocol etc. Here we will discuss about Distributed Energy Efficient Clustering (DEEC) Protocol in brief.

2. The DEEC Protocol
DEEC uses the initial and residual energy level of the nodes for selecting the cluster-heads. DEEC evaluates the ideal value of network life-time, which is used to calculate the reference energy that each node should dissipate throughout a round.

Cluster-head selection algorithm based on residual energy
Let denote the number of rounds to be a cluster-head for the node, and we refer to it as the rotating epoch. In homogeneous networks, to guarantee that there are average cluster-heads every round, Leach let each node ($i = 1,2,\ldots,N$) becomes a cluster-head once every rounds. Remember that all the nodes cannot have the same residual energy when the network evolves. In our DEEC protocol, we choose different based on the residual energy of node at round r.
Let, which can be also regarded as average probability to be a cluster-head during rounds. When nodes have the same amount of energy at each epoch, choosing the average probability to be can assure that there are cluster-heads every round and all nodes die generally at the same time. If nodes have different amounts of energy, of the nodes with

more energy should be larger than. Let represents the average energy at round *r* of the network, which can be obtained by (1)
To compute by Eq. (1), each node should have the knowledge of the total energy of all nodes in the network. Using to be the reference energy, we have (2)
This guarantees that the average total number of cluster-heads per round per epoch is equal to: (3)
It is the optimal cluster-head number we want to achieve. We get the probability threshold, that each node use to determine whether itself to become a cluster-head in each round, as follow (4)
Where G is the set of nodes that are eligible to be cluster-heads at round *r*. if node has not been a cluster-head during the most recent rounds, we have. In each round *r*, when node finds it is eligible to be a cluster-head, it will choose a random number between 0 and 1. If the number is less than threshold, the node becomes a cluster-head during the current round.
Note the epoch is the inverse of. From Eq. (2), is chosen based on the residual energy at round *r* of node as follow
(5)
Where denote the reference epoch to be a cluster-head. Eq. (5) shows that the value of rotating epoch of each node varies around the reference epoch. The nodes with high residual energy become the cluster heads more times than lower-ones.

2. Coping with Heterogeneous Nodes

is the reference value of the average probability, which determine the rotating epoch and threshold of node. In homogeneous networks all the nodes are equipped with the same initial energy, thus nodes use the same value to be the reference point of. When the networks are heterogeneous, the reference value of each node should be different according to the initial energy. In the two-level heterogeneous networks, we alter the reference value with the weighted probabilities given in Eq. (4) for normal and advanced nodes. (6)
Therefore, is changed into (7)
Substituting Eq. (7) for on (4), we can get the probability threshold used to elect the cluster-heads. Thus the threshold is correlated with the initial energy and residual energy of each node directly.
This model can be easily extended to multi-level heterogeneous networks. We use the weighted probability shown in Eq. (8) (8)
To replace of Eq. (2) and obtain the for heterogeneous nodes as (9)
From Eqs. (8) and (9), expresses the basic rotating epoch of node, and we call it reference epoch. It is different for each node with different initial energy. Note, thus the rotating epoch of each node fluctuates around its reference epoch based on the residual energy. If, we have, and vice versa. This means that the nodes with more energy will have more chances to be the cluster-heads than the nodes with less energy. Thus the energy of network is well distributed in the evolving process.

3. Estimating Average Energy of Networks

From Eqs. (7) and (9), the average energy is required to calculate the average probability. It is difficult to realize such scheme, which presumes that each node knows the average energy of the network. We will estimate in this paragraph.
The average energy is used to be the reference energy for each node. It is the ideal energy that node should own in current round to keep the network alive to the greatest extent. In such ideal situation, the energy of the network and nodes are constantly shared, and all the nodes die at the same time. Thus we can estimate the average energy of *r*th round as follow, (10)
Where *R* denotes the total rounds of the network lifetime. It means that every node consumes the same amount of energy in each round, which is also the target that energy-efficient algorithms should try to achieve. From Eq.(5), considering as the standard energy, DEEC controls the rotating epoch of each node according to its current energy, thus controls the energy expenditure of each round. As a result, the actual energy of each node will fluctuate around the reference energy. Therefore, DEEC guarantees that all the nodes die at almost the same time. In fact, it is the main idea of DEEC to control the energy expenditure of nodes by means of adaptive approach.
To compute by Eq. (10), the network lifetime *R* is needed, which is also the value of an ideal state. Let Us consider that all the nodes die at the same time, *R* is the total of rounds from the network begins to all the nodes die. Let denote the energy consumed by the network in each round *R* can be approximated as follow. (11)
In the analysis, we use the same energy model as proposed earlier. In the process of transmitting an *l*-bit message over a distance *d*, the energy expended by the ratio is given by:
(12)
Where is the energy dissipated per bit to run the transmitter or the receiver circuit, and or is the amplifier energy that depend on the transmitter amplifier model.
We assume that the *N* nodes are spread constantly in an $M \times M$ region, and the base station is located in the center of the field for simplicity. Each non-cluster-head send *L* bits data to the cluster-head a round. Thus the total energy dissipated in the network during a round is equal to:
(13)
Where *k* is the number of clusters, is the data aggregation cost expended in the cluster-heads, is the average distance between the cluster-head and the base station, and is the average distance between the cluster members and the cluster-head. Assuming that the nodes are uniformly distributed, we can get: (14)
By setting the derivative of with respect to *k* to zero, we have the optimal number of clusters as (15)
Substituting Eqs. (14) and (15) into Eq. (13), we obtain the energy dissipated during a round. Thus we can compute the lifetime *R* by (11).
In Fig. 1, we show the value of analytical lifetime when *a* and *m* are changed. Because of the affection of energy heterogeneity, the nodes can't die exactly at the same time. If let R of Eq. (10) be the estimating value by Eq. (11), the reference energy will be too large in the end, as we can see from Eq. (10), that is to say that the network will not have a single cluster-head and a few nodes will not die finally. The presumption of the energy of the network and nodes being constantly spread is not prerequisite in practical operation of DEEC. The approximation of *R* is enough to get the reference energy, thus DEEC can adapt well to heterogeneous environments.

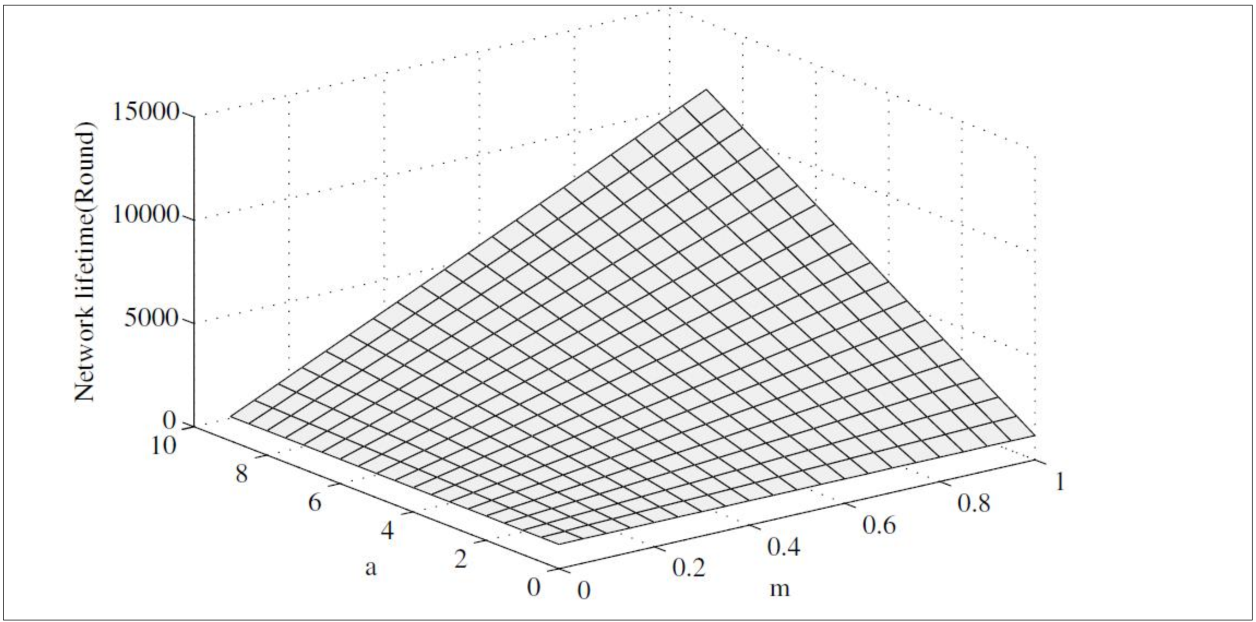


Fig 1: Estimate of Network Lifetime.

Initially, all the nodes need to know the total energy and lifetime of the network, which can be determined a priori. In our DEEC protocol, the base station could broadcast the total energy and estimate value *R* of lifetime to all nodes. When a new epoch starts, each node will use this instruction to calculate its average probability by Eqs. (10) and (9). Note will substitute into Eq. (4), and get the election threshold, which is used to decide if node should be a cluster-head in the current round.

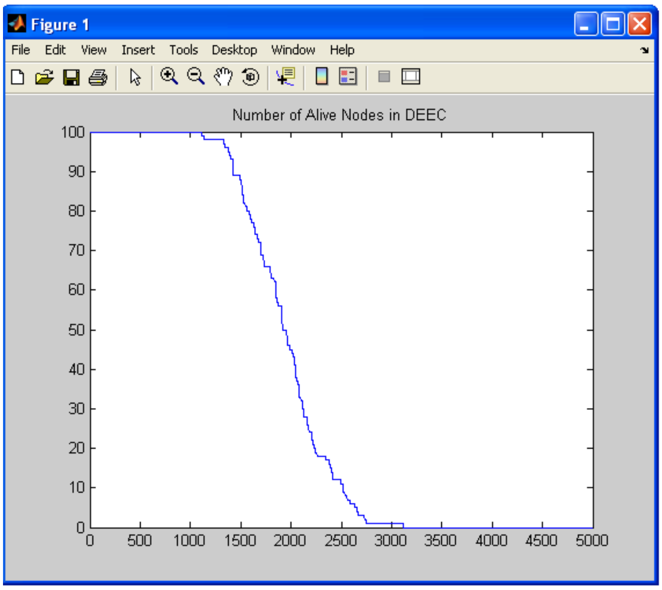
3. Implementation

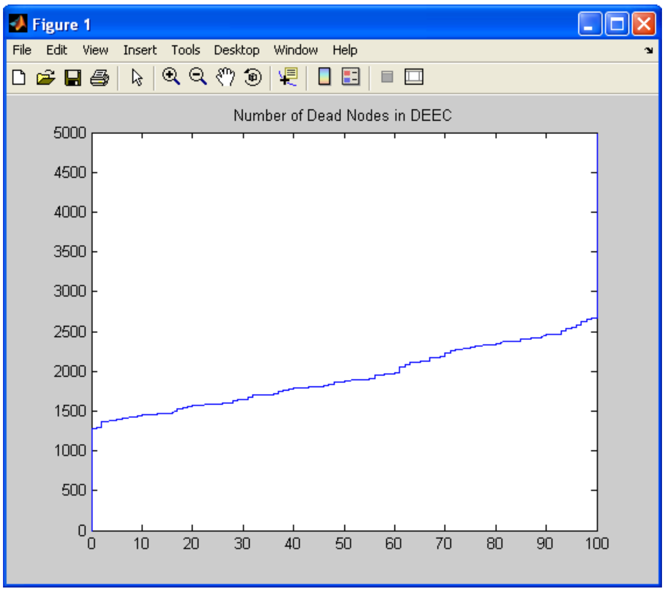
```
clear all
xm=100;
ym=100;
sink.x=0.5*xm; %location of sink on x-axis
sink.y=0.5*ym; %location of sink on y-axis
n=100 %nodes
P=0.1; %probability of cluster heads
Eo=0.5; %initial energy
ETX=50*0.000000001; %tx energy
ERX=50*0.000000001; %rx energy
Efs=10*0.000000000001; %free space loss
Emp=0.0013*0.000000000001; %multi path loss
%Data Aggregation Energy
EDA=5*0.000000001; %compression energy
a=1; %fraction of energy enhancment of advance nodes
rmax=5000 %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=4000*(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;
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)
```

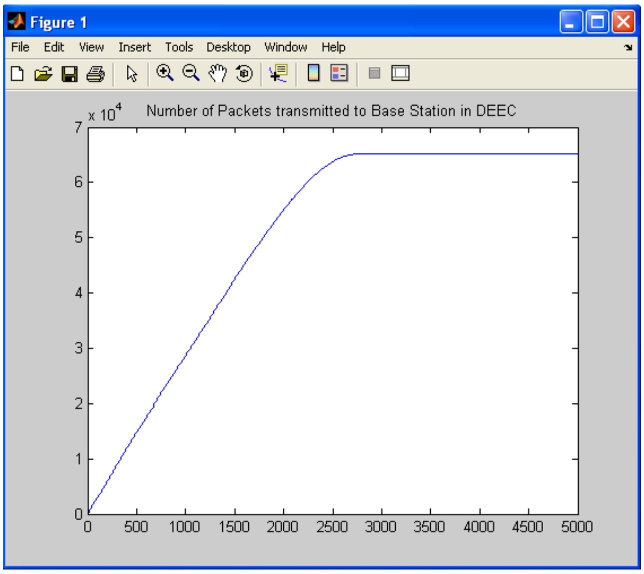
```
dead=dead+1;
if (dead==1)
if(flag_first_dead==0)
first_dead=r;
flag_first_dead=1;
end
end
if(dead==0.1*n)
if(flag_teenth_dead==0)
teenth_dead=r;
flag_teenth_dead=1;
end
end
if(dead==n)
if(flag_all_dead==0)
all_dead=r;
flag_all_dead=1;
end
end
end
if S(i).E>0
S(i).type='N';
end
end
STATISTICS.DEAD(r+1)=dead;
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))))))
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)*(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
end
STATISTICS.COUNTCHS(r+1)=countCHs;
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
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;
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*(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;
end
end
end
STATISTICS.PACKETS_TO_CH(r+1)=packets_TO_CH;
STATISTICS.PACKETS_TO_BS(r+1)=packets_TO_BS;
end
first_dead
teenth_dead
all_dead
```

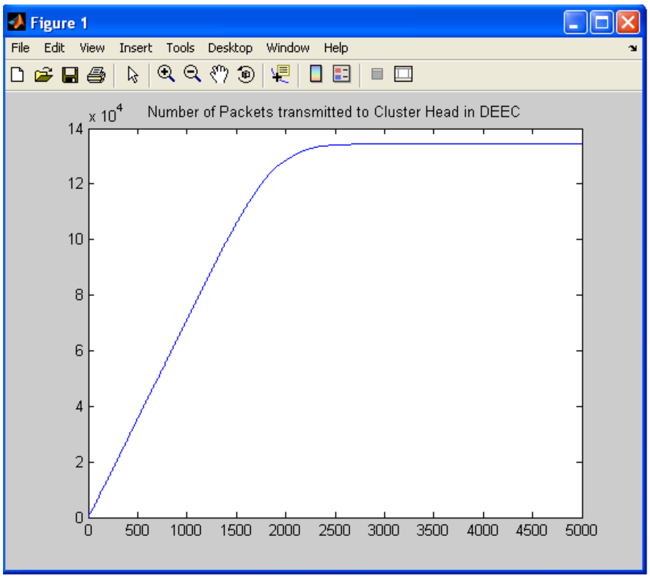
```
STATISTICS.DEAD(r+1)
STATISTICS.ALLIVE(r+1)
STATISTICS.PACKETS_TO_CH(r+1)
STATISTICS.PACKETS_TO_BS(r+1)
STATISTICS.COUNTCHS(r+1)
r=0:5000;
plot(STATISTICS.DEAD,r);
title('Number of Dead Nodes in DEEC');
plot(r,STATISTICS.ALLIVE);
title('Number of Alive Nodes in DEEC');
plot(r,STATISTICS.PACKETS_TO_BS);
title('Number of Packets transmitted to Base Station in DEEC');
plot(r,STATISTICS.PACKETS_TO_CH);
title('Number of Packets transmitted to Cluster Head in DEEC');
plot(r,STATISTICS.COUNTCHS);
title('Number of Cluster Heads in DEEC');
```

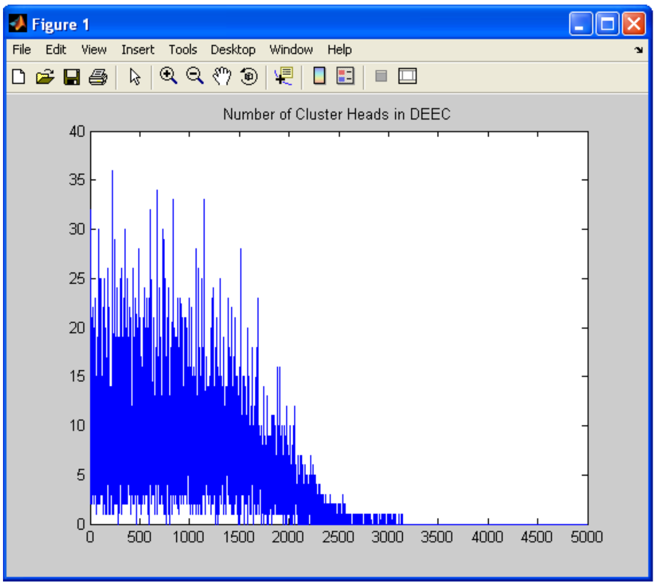
4. Result











5. Conclusion

In this research, we have discussed about Distributed Energy-Efficient (DEEC) Protocol. DEEC is an energy-aware adaptive clustering protocol which is designed to deal with nodes of heterogeneous WSNs. For CH selection, DEEC uses initial and residual energy level of nodes. DEEC evaluates the ideal value of network life-time, which is used to calculate the reference energy that each node should dissipate throughout a round. In DEEC, every sensor node independently elects itself as a cluster-head based on its initial energy and residual energy. To restraint the energy disbursement of nodes by means of adaptive approach, DEEC use the average energy of the network as the reference energy. Thus DEEC does not require any global knowledge of energy at every election round. Unlike SEP and LEACH, DEEC can perform well in multi-level heterogeneous wireless sensor networks.

References

1. Akkaya K, Younis M. A survey on routing protocols for wireless sensor networks. Ad hoc networks. 2005; 3(3):325-349.
2. Deng S, Li J, Shen L. Mobility-based clustering protocol for wireless sensor networks with mobile nodes. IET wireless sensor systems. 2011; 1(1):39-47.
3. Jindal P, Gupta V, Study of energy efficient routing protocols of wireless sensor networks and their further researches: a survey. Energy. 2013; 2(2).
4. Muruganathan SD, Ma DC, Bhasin RI, Fapojuwa A. A centralized energy-efficient routing protocol for wireless sensor networks. Communication Magazine, IEEE, 2005; 43(3):S8-13.
5. Pantazis NA, Nikolidakis SA, Vergados DD. Energy-efficient routing protocols in wireless sensor networks: A survey. Communications Surveys & Tutorials, IEEE, 2013; 15(2):551-591.
6. Akyildiz F, Su W, Sankarasubramaniam Y, Cayirici E. A survey on sensor networks, IEEE communications magazine. 2002; 40(8):102-114.
7. Heinzelman AP, Chandrakasan H, Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS-33), 2000.
8. Mhatre V, Rosenberg C. Design guidelines for wireless sensor networks: communication, clustering and aggregation, Ad Hoc Network Journal. 2004; 2(1):45-63.
9. Ye M, Li C, Chen G, Wu J. EECS: an energy efficient cluster scheme in wireless sensor networks, in: IEEE International Workshop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks (IEEE IWSEEASN2005), Phoenix, Arizona, 2005.
10. Depedri A, Zanella A, Verdone R. An energy efficient protocol for wireless sensor networks, in: Autonomous Intelligent Networks and Systems (AINS 2003), Menlo Park, CA, 2003.