Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-5, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Distributed Clustering Protocol for Wireless Sensor Network

Shiv Kumar, Pankaj Kumar, Anjali Datyal and Ashish

Abstract—*In wireless sensor networks, energy constrained sensor nodes communicate with each other wirelessly. To develop an energy efficient routing protocol for wireless sensor networks is the most challenging problem now days. To prolong the network life time by utilizing the energy in a smart way is crucial issue in sensor networks. In this paper, we propose an energy efficient Distributed Clustering routing Protocol (DCP) for prolonging the lifetime of sensor networks. In DCP, clusters are formed in first round are fixed for further rounds and cluster heads (CH_S) are appointed in every round within the clusters.*

Keywords—Clustering, Data Aggregation, Energy Efficiency, Network Lifetime, Routing, Wireless Sensor Networks.

I. Introduction

In wireless sensor networks, sensor nodes should work for long time to maximize the lifetime of sensor network [8]. Energy is an important constraint to prolong the sensor networks lifetime, therefore energy aware protocols are needed at every layer of protocol stack [7]. At network layer it is important to develop energy efficient routing algorithms for routing to prolong the network lifetime [10]. Wireless sensor networks (WSNs) have many challenges which depends on applications where sensor nodes are deployed. Challenges may be related to sensor network protocols, designing of algorithms in sensor network, and software's related to sensor nodes. Sensor networks can contain hundreds or thousands of sensing nodes. It is desirable to make these nodes as cheap and energy-efficient as possible and rely on their large numbers to obtain high quality results [3]. Device failure is a regular or common event due to energy depletion or destruction. Since, the sensor nodes are often inaccessible; the lifetime of a sensor network depends on the lifetime of the power resources of the nodes. To prolong network lifetime, a sensor node must enter into periods of reduced activity when running low on battery power [5]. This work is focused to prolong the lifetime of WSNs by designing an energy efficient protocol. In DCP to prolong the sensor network lifetime, sensor nodes are clustered once for whole lifetime of network, data aggregation is done by cluster heads (CHs) after that aggregated data is sent to the sink. DCP efficiently handles failure of nodes during election of cluster heads. DCP utilizes energy efficiently and smartly as compared to other previous clustering algorithms. In DCP, first clusters are made by the algorithm runs at the sink and are fixed for further rounds. After the first round, cluster heads are rotated within the clusters once formed. Selected cluster heads will directly communicate to the sink using single hop communication. In WSN, energy consumed by sensor nodes during communication can be estimated by free space model and multipath fading model [4].

II. Related Work

In wireless sensor networks, various clustering algorithms have been proposed. In all previous clustering algorithms, clusters are made through the grouping of nodes and one node plays the role of cluster head. Cluster heads receive the data from the member sensor nodes, aggregate them and send to the sink. Most crucial clustering protocol is LEACH [3]. In LEACH, network arranges itself into clusters. In LEACH, round wise new clusters are made and new cluster heads are selected by threshold function to balance the energy throughout the network because cluster heads consume more energy compare to non-cluster heads nodes. LEACH-C [4] also works similarly as LEACH except set-up phase. In EEMC [6] multilevel clustering concept is given to increase the scalability of the networks. In EEMC, super cluster contains the sub-clusters. Cluster heads of the sub-clusters transmit aggregated data to upper level cluster heads until they reach to super cluster head. At the last super cluster head aggregates received data and send it to the sink. In HRP [9], first network is organized into clusters, after that a routing tree is constructed on cluster heads. Finally root cluster head send the aggregated data to the base station. SOLAC [11], divides the sensor field into concentric rings, and the SOLAC protocol is designed for intra-ring clustering and inter-ring routing. Cluster heads in a ring closer to the sink have smaller sizes than those in the rings further away from the sink, and hence CHs can spend less energy for intra-cluster data processing and more energy for inter-cluster data relay. In [12], a load balanced clustering scheme for wireless sensor networks has been proposed. The algorithm is optimal for the case in which the sensor nodes have equal load. In MCDA [14], the algorithm not only has effect on lessening blind broadcasting but also decrease the message exchange. It performs efficient centralized decision making for cluster designing and energy aware distributed cluster head selection and cluster member allocation process. In ECOMP [13], each sensor node consumes a small amount of transmission energy in order to reach the neighboring sensor node in the bidirectional ring, and the cluster heads do not need to receive any sensed data from member nodes.

In direct communication protocols, each sensor node sends its data to the sink directly. In minimum-transmission-energy (MTE) routing protocol, nodes send data to the base station by intermediate nodes [1]. In static clustering, nodes are once organized into clusters and there is one fixed cluster head, which receives the data from their cluster members, aggregate it before transmitting it to sink.

III. Assumptions

DCP has been designed with following assumptions:

1) Every node knows its position from each other and from the sink through GPS system and by range-based or range free localization schemes.
2) Every node is aware about its energy level.
3) Nodes can directly communicate with the sink.
4) Every node has a table to maintain the information about its neighbors.
5) All nodes sense same type of event.
6) Sensor nodes and the sink are static and the sink has constant power source.
7) All sensor nodes are homogeneous.

IV. Proposed Scheme

During the setup phase every node in an area sends its energy (*E*), distance from the sink (d_n), density (number of neighboring nodes = *D*) to the sink as shown in Fig. 1. Let co-ordinates of a node are (X_n, Y_n) and co-ordinates of the sink is (X_{si}, Y_{si}) then distance between node and the sink will be calculated at the node using equation (1).

$$d_n = \sqrt{(x_{si} - x_n)^2 + (y_{si} - y_n)^2} \quad (1)$$

The density can range from few sensor node to few hundred sensor nodes in an area, which can be less than 10m in diameter. The density (*D*) can be calculated according to the following equation (2).

$$\mu(R) = (N\pi R^2)/A \quad (2)$$

Where, *N* is the number of scattered sensor node in an area *A*; and *R*, the transmission range. Basically, *μ(R)* gives the number of nodes with in the transmission radius of each node in an area *A* [5]. In proposed scheme we are considering *R*, the sensing range and *μ(R)* gives the number of nodes with in the sensing range of each node in an area *A*.

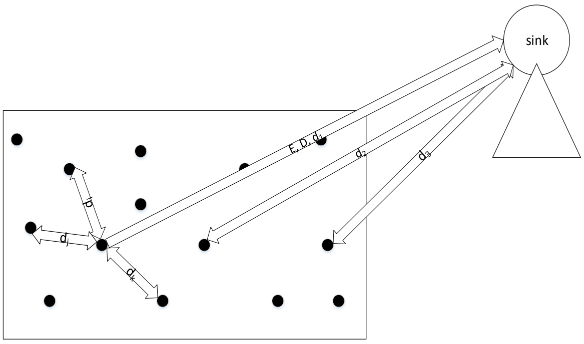

Fig. 1. Nodes transmit its parameters to the Sink.

After getting the energy(*E*), distance(*d*), density(*D*) information from nodes will create dense clusters so that communication between CH and Cluster Nodes (CNs or member nodes) within a cluster can be minimized [4]. The clusters formed are in range 2-4 for 100 nodes are considered as dense clusters.

If CHID matches with node ID then node becomes a CH. If CHID doesn't match then node will send an *Attach Message* to the nearest CH whose ID is received by it from the sink.

After getting the *Attach Message* (according to Algorithm 1) cluster head will prepare the TDMA schedule and multicast it to all the nodes who joined it according to Algorithm 2. After getting the TDMA schedule every node in cluster will update its table for number of nodes in its cluster. Now CNs will send data to its respective CH according to its TDMA schedule to avoid collision. After getting the data from all CNs, CH will aggregate it and send it to the sink. This whole process is the first round. In further rounds clusters will be fixed until all the nodes in an area exhaust their energy. Only cluster heads will be changed in clusters to equally distribute the energy among the nodes in clusters. So the Timeline for data transmission for DCP is shown in Fig. 2.

In Fig. 2, in SET-UP phase, cluster heads will be selected and clusters will be formed. In DATA-TRANSMIT phase, CHs will receive data from member nodes, aggregate data and send to the sink.

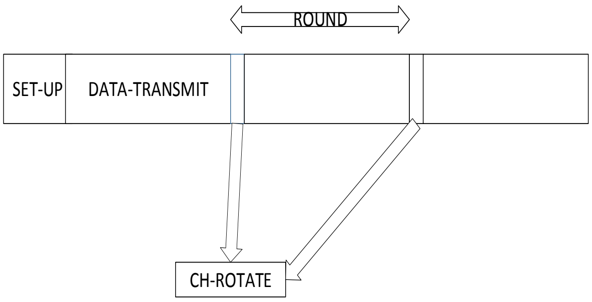

Fig. 2. Timeline.

V. Cluster Head Selection In Cluster For Further Rounds

Every node maintains a table which contains the following parameter values:
THR: Required threshold value of V_{CCN} to communicate with sink which is dependent on application, area and energy.

CLUS. nodes: All nodes in a cluster.

V_{CCN}: Current eligible cluster head nodes' values.
Each node calculates V_{CCN} according to the following equation.

$$V_{CCN} = RE_i \times (1-(DS_i/100)) \quad (3)$$

Where
RE_i = residual energy of node i.
DS_i = distance between node i and the sink.

PCH: whether a node was CH in previous rounds. It is indicated as yes (Y or 1) or no (N or 0);
Node table for node i is shown in Fig. 3.

i

Node i table		
CLUS. nodes	V_{CCN}	*PCH*
A	20	N(0)
B	80	N(0)
C	10	N(0)
D	80	Y(1)

Fig. 3. Node table.

Algorithm 1. Algorithm to send Attach Message.

AMS(V_{CCN}, *THR*)
{
for (i = 1 to n){ //number of nodes who updated its tables by V_{CCN}.
if (V_{CCNi} is largest and ($V_{CCNi} \geq THR$) and PCH_i = = N) // V_{CCNi} is i-th nodes' V_{CCN} {
Node will send *Attach Msg* to i-th node.}

else if (more than one node have same largest V_{CCN} and ($V_{CCN} \geq THR$) and *PCH* = = N) {
Node will send *Attach Msg* to randomly one selected node.}

else if (more than one node have same largest V_{CCN} and ($V_{CCN} \geq THR$) and some have *PCH* = = Y){
Node will send *Attach Msg* to randomly one selected node among those who have *PCH* = = N
}

else if (more than one node have same largest V_{CCN} and ($V_{CCN} \geq THR$) and all have *PCH* = = Y){
Make entry for all nodes' PCH = = N and send *Attach Msg* to randomly one selected node. }

}
}

Algorithm 2. Algorithm to select cluster head.

CHS(Attach Msg A, n_1) // n_1 is number of nodes which have same highest V_{CCN}
{
for (i = 1 to n_1){
if (A_i is largest)//number of *Attach Massages* at node i{
i-th node will send TDMA to member nodes.}
else {
Nodes n_1 will send *Attach Msg* to each-other only randomly.
CHS(A,n_1); }
}
}

Node which has highest V_{CCN} value and ($V_{CCN} \geq THR$) will be considered cluster head for current round. After sending the data to the sink in first round cluster heads node will send election schedule to cluster members' node to start election for new cluster heads.

In Fig. 4, member nodes A, B, C, D get the election schedule t1, t3, t2, t4 respectively. First node A will start election and send its V_{CCN} to other member nodes. After getting the A' V_{CCN}, only those nodes will response to election whose $V_{CCN} \geq$ As' V_{CCN}. So node C will not respond to election. Nodes B, D will respond to election.

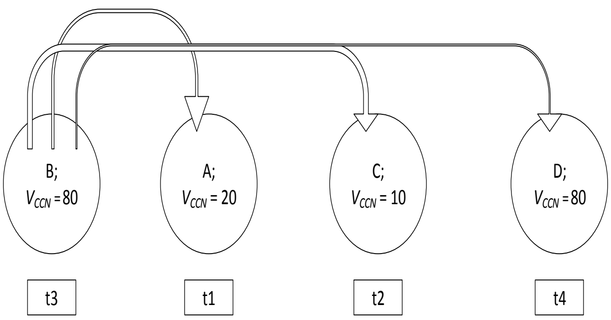

Fig. 4. Propagation of V_{CCN} by nodes.

From Fig. 5 when node B will multicast its V_{CCN} value then it will take time according to equation (4).

$$T_1 = T_{trans} + T_{prop} \quad (4)$$

And to get V_{CCN} value of another node it will wait for time T according to equation (5).

$$T = T_1 + T_{prop} = T_{trans} + 2T_{prop} \quad (5)$$

We are ignoring T_{trans} from another node for V_{CCN} messages response. If node will get V_{CCN} from another node within time T then it will wait again for next T' time for another nodes' V_{CCN} value to update its table. Here T'= T and T_{trans} is transmission time of one node and T_{prop} is propagation time from one node to another nodes during multicast (according to Algorithm 3).

Algorithm 3. Algorithm for timeout.

Atimeout(n,T) //here n is number of nodes and T is time.
{
for (i=1 to n){
if (node get V_{CCN} of other node in time T){
Lower V_{CCN} valued node will not response and wait for next T' time for another nodes' V_{CCN} value msg.
}
else if (node didn't get V_{CCN} of other node in time T'){
Node will check its updated table and send TDMA schedule if it has highest V_{CCN} value. Otherwise wait for TDMA schedule to send data.
}
}
}

Case 1

If (more than one node has same highest V_{CCN} value and
($V_{CCN} \geq THR$) and *PCH* = = N means in case of ties)

Then member nodes will send randomly *Attach Messages* to the nodes which have highest V_{CCN}.

Highest V_{CCN} values' nodes will compare the number of *Attach Messages* received by them respectively. Node with, maximum number of received *Attach Messages* will be elected as CH for the current round. CH will then send TDMA schedule for member nodes as shown in Fig. 6.

If A, B, C get the same number of *Attach Messages* as in Fig. 7. Then A, B, C will send *Attach Messages* randomly to each other only, one who will get maximum *Attach Messages* will win and send TDMA schedule to member nodes as shown in Fig. 8.

If A, B, C send *Attach Messages* to each other randomly yet number of *Attach Messages* are same as in Fig. 9. Then A, B, C will send again *Attach Messages* to each other until one of them has more *Attach Messages* than others as shown in Fig. 8.

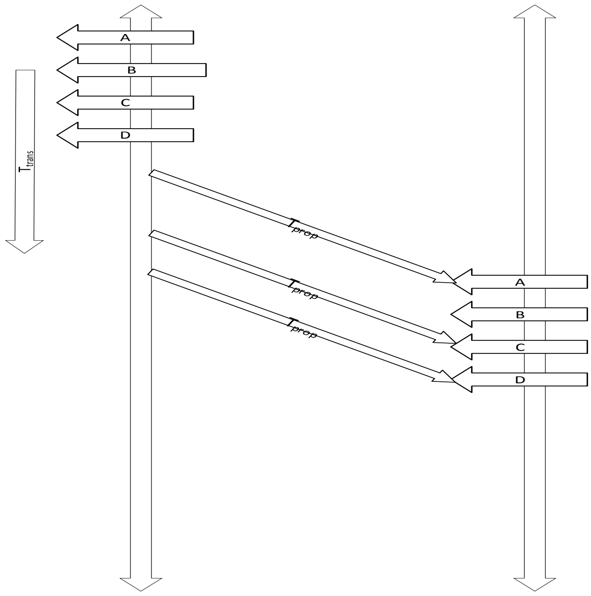

Fig. 5. Transmission and Propagation time of V_{CCN}.

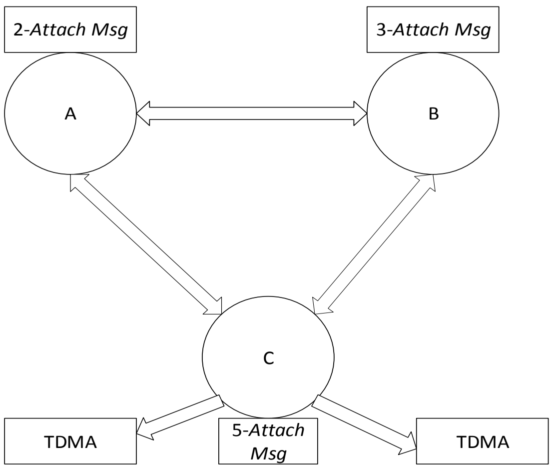

Fig. 6. Node sends TDMA after win.

Case 2

If (more than one node have same highest V_{CCN} value and
($V_{CCN} \geq THR$) and some nodes have *PCH* = = Y)

Then follow the *Case 1*. Only for those nodes who have PCH = = N.

Case 3

If (more than one node have same highest V_{CCN} value and ($V_{CCN} \geq THR$) and all nodes have *PCH* = = Y)

Then make entry for all nodes' PCH = = N. and follow *Case 1*.

In first round after getting the CHID from the sink every node will update its *PCH* value for respective cluster head. So for next round, nodes will not consider it as cluster head.

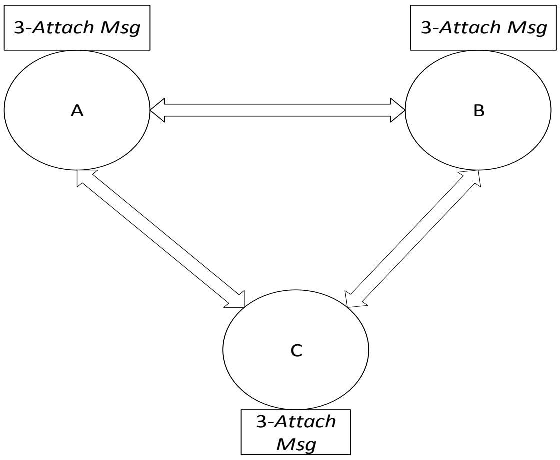

Fig. 7. Tie during cluster head selection.

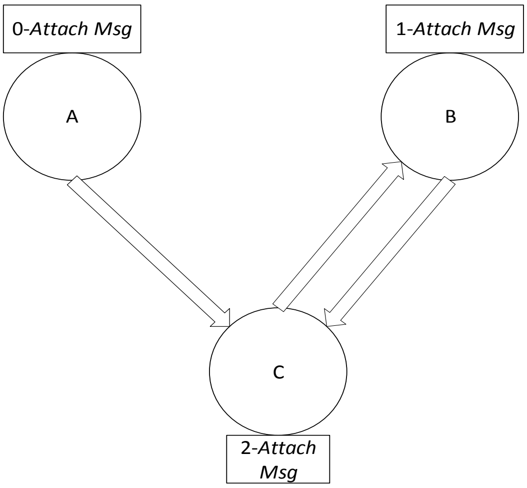

Fig. 8. Voting process after tie.

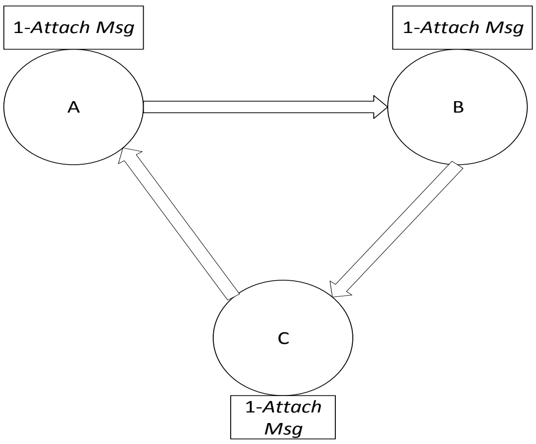

Fig. 9. Tie after voting process.

In second round after updating its tables, all nodes will know their respective cluster head for current round. There is no need to send messages to join the cluster head by member nodes because clusters are fixed. Nodes will send only *Attach Messages* to select cluster head as discussed in earlier.

CHs after checking their table, will send TDMA schedule for remaining nodes along with nodes of last rounds' cluster head for coverage. After getting the TDMA schedule nodes will send data to their respective CHs.

In each round after sending the data to the sink, current cluster heads node will send election schedule to all its member nodes in clusters to initiate next round. In proposed scheme, clusters are independent to each other therefore, there is no need of time synchronization among clusters to initiate next round for new cluster heads selection, resulting in energy conservation.

Member nodes inside the clusters will communicate to each other to update its tables using DSSS (Direct Sequence Spread Spectrum) [4] to avoid interference from neighboring clusters. Inside the clusters, cluster heads node will send short feedback messages to each of the member nodes telling them to increase or decrease their transmission power, as is done in cellular systems [4] for communication within the clusters. Cluster heads will send the aggregated data to the sink using CSMA.

VI. Results

DCP scheme was simulated using Castalia-3.2 [16] simulator. It is based on the OMNeT++ [15] platform. The parameters used in simulations are same as in [9]. Simulation results presented here are the average of 150 independent experiments where in each experiment sensor nodes are deployed randomly.

Fig. 10 shows the graph of the number of data signal received at the sink with respect to time. Here performance of DCP is better than LEACH-C and LEACH because in DCP, clusters are fixed and independent to each other so delay in time synchronization among clusters to form new clusters is reduced. Fig. 10 shows that graph of the data signal received at the sink is constantly increasing with respect to time and becomes constant. This is because of as time increases more data packets are transmitted from the nodes which leads to decay in energy of the nodes. So with respect to time, nodes will start dying so after some time graph of data signal received at the sink with respect to time becomes constant.

Fig. 11 shows the graph of the number of data signal received at the sink with respect to total energy consumed by the nodes. Here performance of DCP is better than LEACH-C and LEACH because in DCP, clusters are made in first round by communication to the sink and in further rounds cluster heads are selected by communication of member nodes within the clusters so much amount of energy is saved in DCP. Fig. 11 shows that as the number of data signal received at the sink increases, total energy consumed by the nodes also increases till all the nodes exhaust their energy.

Fig. 12 shows the number of nodes alive with respect to time. Here performance of DCP is better than LEACH-C and LEACH because in DCP, time synchronization cost among clusters to form new clusters is saved in each round and communication cost for clusters formation and cluster heads selection, to the sink in each round is reduced. Fig. 12 shows that at start graph of the number of nodes alive with respect to time is constant for some time and after that graph starts to decrease. This is because till all nodes have energy greater than threshold energy, graph is constant and as time increases nodes start to die. It is seen from the figure that nodes are alive for a longer duration in DCP when compared with other schemes. This shows that our proposed scheme is more energy conserving as compared to LEACH-C and LEACH protocols.

Fig. 13 shows that as the number of data messages received at sink increases, more nodes starts to die. It is seen from the fig. 13 that DCP is energy efficient in handling more data messages as compared to LEACH-C and LEACH schemes. More data messages are transmitted to sink by DCP as compared with the other schemes.

Fig. 14 shows the number of nodes alive with respect to number of rounds. Here performance of DCP is better than LEACH-C and LEACH because it is more energy conserving as compared to LEACH-C and LEACH. Fig. 14 shows the number of nodes alive with respect to the number of rounds is constant for some time. As number of rounds increase, nodes start dying which results in falling of the graph till all the nodes exhaust their energy. Fig. 14 shows that more number of data communication rounds are performed by DCP, hence its performance is better as compared to LEACH-C and LEACH.

VII. Conclusion

In this paper, to prolong the sensor network lifetime clustered routing is used by our proposed scheme namely DCP to select cluster heads without communication to the sink. Simulation results show that DCP prolong the network lifetime by reducing energy consumption for clusters formation and cluster heads selection. Simulation results also show that our proposed scheme outperforms LEACH-C and LEACH protocols.

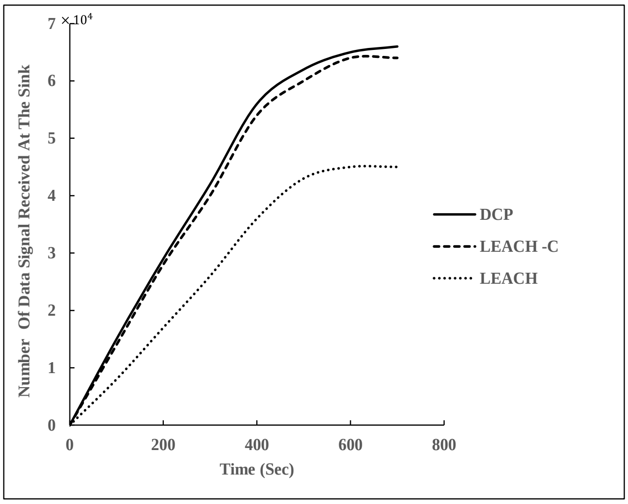

Fig. 10. Data signal received at the sink with respect to time.

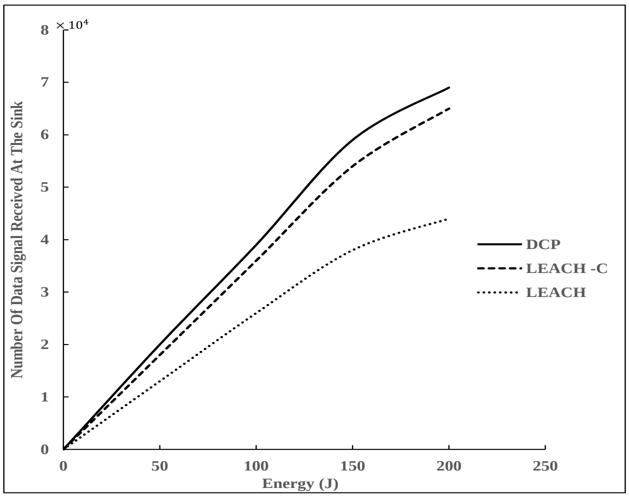

Fig. 11. Received data signal with respect to total energy consumed.

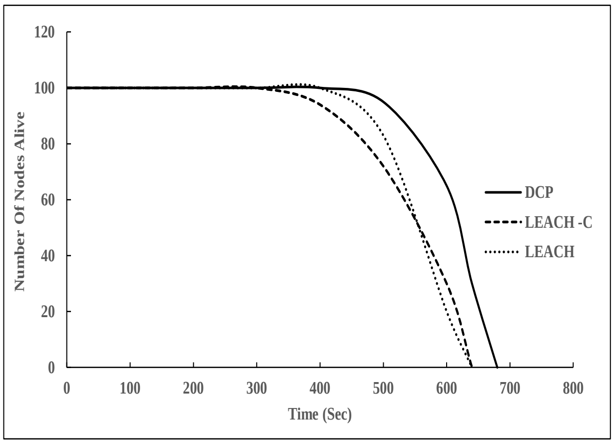

Fig. 12. Number of nodes alive with respect to time.

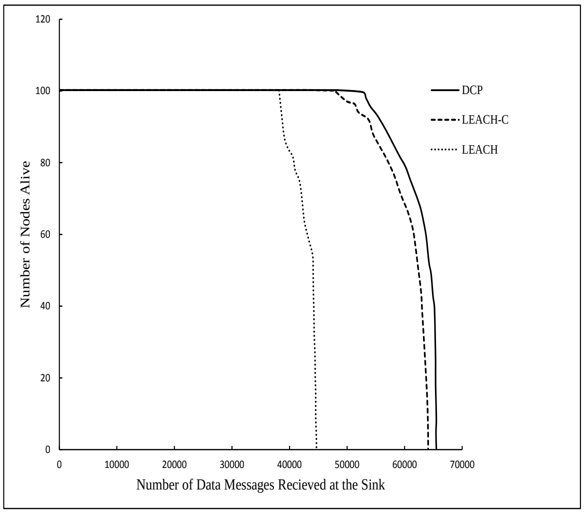

Fig. 13. Number of nodes alive with respect to data messages received at the sink.

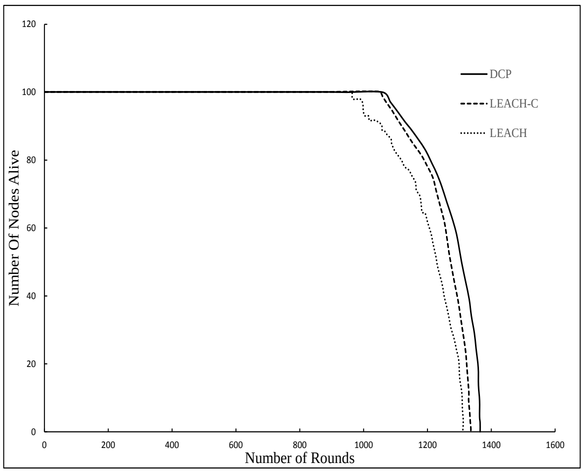

Fig. 14. Number of nodes alive with respect to number of rounds.

REFERENCES

[1] X. Lin, I. Stojmenovic, "Power-Aware Routing in Ad Hoc Wireless Networks," *In SITE, University of Ottawa*, TR-98-11, Dec. 1998.

[2] J. Pottie, W.J. Kaiser, Embedding the internet, "wireless integrated network sensors," *ACM Journal of Communications* 43 (5),51–58, 2000.

[3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks,"*Massachusetts Institute of Technology, Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000.

[4] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE transaction on wireless communication*, vol.1, no. 4, October 2002.

[5] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks* 38, 393–422, 2002.

[6] Y. Jin, L. Wang, Y. Kim, X. Yang, "EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks," *Computer Networks* 52, 542–562, 2008.

[7] Y. C. Chang, J. P. Sheu, "An energy conservation MAC protocol in wireless sensor networks," *Wireless Personal Communications Journal*, 48, 261–276, 2009.

[8] I. F. Akyildiz, M. C. Vuran, "Vuran wireless sensor networks," *London:Wiley*, 2010.

[9] M. Zeynali, A. Mollanejad, L. Khanli , "Novel hierarchical routing protocol in wireless sensor network," *Procedia Computer Science* 3,292–300, 2011.

[10] M. Eslaminejad, S. AbdRazak, " Fundamental lifetime mechanisms in routing protocols for wireless sensor networks: A survey and open issues," *Journal of Sensors*, 12, 13508–13544, 2012.

[11] Y. Mo, B. Wang, W. Liu, Laurence T, Yang, "A Sink-Oriented Layered Clustering Protocol for Wireless Sensor Networks," *Mobile Netw Appl, Springer Science+Business Media New York*, 18:639–650 DOI 10.1007/s11036-013-0443-1, May 2013.

[12] P. Kuila, Prasanta K, Jana, "Approximation schemes for load balanced clustering in wireless sensor networks," *Springer Science+Business Media New York*, Sept. 2013.

[13] F. Bajaber, I. Awan, "An efficient cluster-based communication protocol for wireless sensor networks," *Springer Science+Business Media New York*, August 2013.

[14] S. Jabbar, A. Minhas, A. Paul, S. Rho, "Multilayer cluster designing algorithm for lifetime improvement of wireless sensor networks," *Springer Science+Business Media New York*, Feb. 2014.

[15] OMNeT++ network simulation framework. http://www.omnetpp.org/.

[16] Castalia Wireless Sensor Network Simulator. castalia.forge.nicta.com.au.

Shiv Kumar obtained his M.C.A degree from BundelKhand University-Jhansi(U.P.), India in 2011. He has completed his M.Tech. Degree in Mobile Computing at Department of Computer Science and Engineering, NIT-Hamirpur (H.P.), India in 2015. Currently he is a software engineer in a MNC. His area of interest includes Wireless Sensor Networks and Adhoc Networks.

Pankaj Kumar obtained his B.Tech. degree from IEET, baddi (H.P.), India. He has completed his M.Tech. degree in Mobile Computing at Department of Computer Science and Engineering, NIT-Hamirpur (H.P.), India in 2015. Currently he is an assistant professor at MIT engineering college, bani (H.P.), India. His area of interest includes Wireless Sensor Networks and Adhoc Networks.

Anjali Datyal obtained her B.Tech. and M.Tech. degree from Sri sai college of engineering, Badhani, Pathankot (Punjab), India in electronic engineering. Currently she is an assistant professor at MIT engineering college, bani (H.P.), India. Her area of interest includes digital image processing, Wireless Sensor Networks and Adhoc Networks.

Ashish obtained his B.Tech. degree from UIET, Kurukshetra University, India. He has completed his M.Tech. degree in Mobile Computing at Department of Computer Science and Engineering, NIT-Hamirpur (H.P.), India in 2015. Currently he is a software engineer in a MNC at chandigarh, India. His area of interest includes Wireless Sensor Networks and Adhoc Networks.