

Efficient Clustering with Proposed Load Balancing Technique for MANET

Supreet Kaur
Deptt of CSE,
NSIT, Dwarka, India

Varsha Kumari
Deptt of CSE,
NIT, Delhi, India

ABSTRACT

In this paper, an efficient and modified load balancing algorithm is proposed. This paper shows the results of the proposed energy aware load balancing algorithm. Global re-clustering is initiated when the network becomes significantly unbalanced i.e. if the variance of degree of the cluster heads in the network is greater than a pre-determined threshold. Degree of all cluster heads would be required at a single node to evaluate the average of degree of all cluster heads in the network and subsequently the variance among themselves, which is not possible in adhoc network because of its distributed nature. For this, a formula is derived in which variance can be updated at each hop i.e. variance of N nodes can be expressed in terms of variance of (N-1) nodes and average of (N-1) nodes. The results of simulation indicate that compared with the load balanced routing algorithm DEECA, the proposed algorithm increased the network lifetime, packet delivery ratio at different traffic.

Keywords

MANET, Clustering, Cluster Maintenance, Load Balancing, Energy Conservation

1. INTRODUCTION

In the recent years, Mobile AD hoc NETWORKS (MANETs) have gained more interest from both the scientific community and industry sector, due to their performance, low cost and auto-configuration capabilities.

MANETs are collections of both mobile and static terminals that exchange data by supporting each other's communication. Every node may behave like a terminal and a router, and the packets may be exchanged between sources and destinations through multi-hop routes in a way using the best effort.

The idea of clustering in MANETs is not new. Clustering transforms a physical network into a virtual network of interconnected clusters or group of nodes. These clusters are dominated by clusterheads(CHs) and are connected by gateways or border mobile terminals. Clustering in MANET has been used in order to facilitate management, to improve routing efficiency, to support QoS, and to save power consumption. Once the clusters are established, the route stability between a source-destination pair is based on the CHs involved in that route.

Clustering not only makes a large MANET appear smaller, but more importantly, it makes a highly dynamic topology to appear less dynamic. It has advantages such as improving bandwidth utilization by reducing communication overheads and reducing energy consumption.

In this paper, the result of the algorithm proposed in [1] is discussed and a Load Balanced Reclustering Algorithm(LBRA) is proposed. The algorithm proposes a load balancing heuristic to extend the life of a cluster head to the

maximum budget before allowing the clusterhead to retire and give way to another node. With this, the responsibility of acting as a clusterhead is evenly distributed among all the nodes. In our clustering algorithm, the clusterhead election procedure is not periodic. This reduces system updates and hence computation and communication costs.

The proposed algorithm consists of two distinct phases. The start-up phase performs initial partitioning of a network into clusters, while the maintenance phase updates the topology of a network as time goes on[1]. Depending on the situation, LBRA performs either local updation or global re-clustering. The load balancing in LBRA is accomplished by maintaining a pre-defined threshold on the number of nodes that a clusterhead can cover ideally.

Re-clustering is initiated when a network is highly imbalanced i.e. if the variance of degree difference of all the nodes is greater than pre-determined threshold. Since, degree of all clusterheads was required at a single node to evaluate the variances, which is not possible in adhoc network(distributed network), we derived a formula in which variance of N nodes can be expressed in terms of variance of (N-1) nodes and average of (N-1) nodes.

The paper is organized as follows. In Section 2, several clustering and load balancing algorithms proposed previously are reviewed. Section 3 presents the proposed load balanced re-clustering algorithm for ad hoc networks. In Section 4, formula to calculate variance is derived. Section 5 presents the performance of the proposed algorithm. Finally, the conclusion of the paper is discussed in Section 6.

2. RELATED WORK

Many strategies have been proposed so far to address ad-hoc networks clustering problem.

The Weighted Clustering Algorithm (WCA)[5] employs combined-metrics based clustering : node degree, Cluster head serving time(to estimate residual energy capacity) and mobility/moving speed, are taken into account to calculate a weight factor. Mobile nodes with local minimum weight are elected as clusterheads. To measure how well balanced the clusterheads are, a parameter called *load balancing factor* (LBF) is introduced which is the inverse of the variance of the cardinality of the clusters. WCA does not invoke re-clustering when a member node changes its attaching cluster.

In [6], clustering approach WBACA is presented which is based on the availability of position information via a reliable position locating system, that is, the Global Positioning System (GPS). The weight is decided by a generalized formula that take into account : transmission rate, transmission power, battery power, mobility and degree. The algorithm finds the local minima of weights for clusterhead selection and also assures that no two clusterheads be one-hop neighbours.

Overlapping clusters algorithm (OCA) proposed in [7] consists of two distinct phases. The startup phase performs initial partitioning of a network into clusters based on the following input parameters: battery capacity, transmission range, bandwidth capacity, density of local area, mobility and buffer occupancy. The maintenance phase periodically monitors the status of the network and keeps the network topology updated through local and global re-clustering. OCA may be the preferred algorithm in applications that require higher reliability but do not focus much on efficient cluster formation.

A cluster head, apart from supporting its members with the radio sources, has also to route messages for other nodes belonging to different clusters. Therefore, it is not desirable to have any cluster head overly loaded while others being lightly loaded. At the same time, it is hard to maintain a perfectly load balanced system at all times due to frequent detachment and attachment of the nodes from and to the cluster heads.

Load balancing in [8] is treated as an extension to cluster head election. It allows all nodes an equal opportunity to be a cluster head and extend its duration based on some input parameters.

In [9], it tries to find the best node to share the load but does not consider communication overhead and energy consumption.

Hussein, Yousef and Arabiyat [10] proposed the enhancement on weighted clustering algorithm (EWCA) that leads to a high degree of stability in the network and improves the load balancing. The load balancing is accomplished by determining a pre-defined threshold on the number of nodes that a clusterhead can cover ideally. The stability of the network can be improved by reducing the number of nodes detachment from its current cluster and attachment to another existing cluster. When a cluster size exceeds its threshold/predefined limit, local election procedure is repeated to adjust the number of mobile nodes in that cluster.

In [11], a load balancing algorithm is presented based on clustering where a subset of nodes 'clusterheads' is elected to maintain some balance within their respective clusters while minimizing the overall communication cost. The global members load information of each cluster is collected by the clusterhead which ensures some balance between its members. A minimum threshold of energy is defined for each clusterhead. If it is reached, the clusterhead election procedure is invoked. It is also invoked when a node leaves its cluster and does not find any other cluster to attach itself.

A dynamic energy efficient clustering algorithm presented in [12] selects the clusterheads based on energy and mobility and locally alters the topology to increase the network lifetime and reducing their energy consumption while periodically monitoring their energy. It uses two dynamically-computed energy thresholds: a yellow threshold to achieve local load balancing by distributing the load equally among the adjacent cluster-heads and a red threshold to trigger local re-clustering in the network.

In [13], load balancing among clusterheads is initiated on the basis of their transmission range. If a clusterhead is loaded, it reduces its transmission range while an unloaded clusterhead raises its transmission range and hence fairly redistribute the nodes in the cluster.

For efficient data transmission and load balancing in MANETs, [14] and [15] presents various load balanced

routing protocols like Alternate Path Routing (APR), Dynamic Load Aware Routing (DLAR).

It is concluded from the existing research that several tradeoffs exist for the elected cluster head and cluster members. Firstly, the cluster head has to bear high resource utilization that may deplete its battery sooner than other cluster members. Secondly, despite fair share responsibility of cluster head role, it is possible that heavy burst of traffic takes place causing some cluster heads to use maximum resources, while others encounter low traffic bursts resulting in minimum resource use. Thirdly, the load balancing technique [8] might result in a cluster head that will not provide the optimal path for routing, or a link breakage. Moreover, cluster members are privileged as they don't pay a routing penalty and have resources dedicated for own usage only.

3. MODIFIED PROPOSED ALGORITHM

3.1 Problem Definition

The algorithm proposed in [1] is considered and a load balanced re-clustering algorithm (LBRA) is proposed. It allows for electing a node to play the role of cluster head taking into consideration its degree and energy. It establishes the basis for an energy efficient environment by allowing each node to monitor its energy expenditure and to strive its utmost during the network lifetime.

3.2 Algorithm

The network is divided into clusters. Cluster head is elected in each cluster. Nodes in the network can have any one status :

UNDECIDED(UD), GATEWAY (GW) , CLUSTER MEMBER(CM), or CLUSTER HEAD (CH).

Gateways are responsible for communication between adjacent clusters and connects two clusterheads. Clusterhead is elected on the basis of node's degree and its remaining energy where, **Degree(d)** of a node is defined as the number of neighbors of a node, and **Energy(b)** is defined as the battery/energy left with the node.

Since all nodes are mobile, clusters must be updated frequently to maintain the network topology. For this, all nodes periodically broadcast their identity.

3.3 Algorithm Explanation

The following is a detailed description of algorithm.

3.3.1 Start-up phase

Initial cluster formation takes place in this phase. Each node is considered to have different energy. On the basis of energy left in the node and its degree, clusterhead are elected.

Steps :

1. Initially all nodes are in UD state.
2. Each node i broadcasts *HELLO* packet and sets hello timer, t_h time interval in which it can receive similar *HELLO* packet from its neighbor.
3. Whenever i receives *HELLO* packet, it updates its neighbor information in neighbor table.
4. As t_h expires, i broadcasts *DEGBAT* packet containing its degree and energy left and sets a timer, t_{db} .

5. Nodes receiving this packet will update the neighbour table with degree and battery information of neighbours.
6. When t_{db} expires, each node i calculates its weight W and all of its neighbors, which is an indicative of how capable a node is of becoming a clusterhead.

LBRA converts the clustering problem into an optimization problem and an objective function is formulated, as shown in equation (1).

In [1], each node is assigned a weight that indicates its suitability for the clusterhead role. The weighing factors w_1 and w_2 are kept fixed for a given system. The weighing factors also give the flexibility of adjusting the effective contribution of each of the parameters in calculating the combined weight, W .

$$W = (w_1 * (1 - (d_d / \text{MAX_DEGREE}))) + (w_2 * b / \text{MAX_BATTERY}) \quad \text{--- (1)}$$

where $w_1 = \alpha$

$$w_2 = 1 - \alpha \quad 0 \leq \alpha \leq 1$$

Note that, with this computation, weight value fall in the range of [0..1] because the parameter values and the parameter weight are also normalized[1].

$$d_d = \text{abs}(d_a - d_i)$$

MAX_DEGREE is the maximum degree among all nodes in its neighborhood and itself.

MAX_BATTERY is the maximum battery among all nodes in its neighborhood and itself.

Degree difference (d_d) is a difference of actual degree of the node, d_a and its ideal degree, d_i .

Node declares itself as a Cluster head if

- (i) Its weight is same as maximum weight of its neighbors then the conflict is resolved by selecting node with lower ID.
- (ii) Its weight is maximum among all its neighbors.

All CH's thus selected informs its cluster members. A node having more than one CH as its neighbor becomes a GW. If a battery of a node is less than cluster head threshold then it will not participate in cluster election process.

If a node is in UD state,

- (i) If it has neighbors and at least one of them is in UD state, it initiates CH election procedure among those nodes.
- (ii) If it has no neighbor i.e. the node is an isolated node, it remains in UD state. It changes its status when it comes in contact with other nodes.
- (iii) If it has neighbors but they are all CM's or GW's, it declares itself as a CH.

Once all of the nodes have decided on their status, the network is completely divided into clusters. At this point, the proposed load-balancing algorithm begins its maintenance phase.

Clustering algorithms should have two properties : dominance property and two-hop property[2].

Dominance Property : Initially all nodes start out as UD, being neither a CH nor a CM. At the end of initial clustering procedure, every node will either be a clusterhead itself, or it will be a member of at least one cluster. Thus, the proposed algorithm satisfies the dominance property.

Two-Hop Property : Since the dominance property is satisfied, as discussed above, all nodes in the network are either clusterheads or cluster members. Therefore, it takes a node at most two hops to reach another node in the same cluster. So the LBRA satisfies the two-hop property.

3.3.2 Maintenance phase

When network topology is highly dynamic, resulting in frequent cluster topology updates, the control overhead of cluster maintenance increases drastically. Thus, clustering consumes a large portion of network bandwidth and drain mobile nodes' energy quickly[4]. Minimizing explicit control messages for clustering or limiting re-clustering situations can maintain the cluster structure well without excessive consumption of network resources [3].

In LBRA, clustering is not executed periodically but in an event driven manner i.e., re-clustering is only invoked when there is significant load imbalance among clusterheads in the network.

More precisely, the election procedure is delayed as long as possible to reduce the computation cost as frequent updates result in high information exchange among the nodes resulting in high communication overhead.

If a cluster head battery is less than the minimum threshold then that cluster is dissolved and a local re-clustering takes place. When the network becomes significantly unbalanced i.e. the variance of degree difference of all the nodes is greater than pre-determined threshold, global re-clustering is triggered[1].

3.3.2.1 Cluster Updation

Due to mobility, a node's neighborhood changes with time. Therefore, clusters and cluster heads must be updated frequently to maintain accurate network topology.

Steps involved in cluster updation :

1. Each moving node i sends *HELLO* message every t_h seconds.
2. Whenever a node receives this message, it sets its updation timer, t_u ($3 * T$).
3. As a CM receives this message from a CH, it updates its CH list.
4. If the number of CHs in its neighbor now becomes greater than one then it become GW node and send its CH list to all its CH's.
5. As the timer t_u goes off, node checks its neighbor table for that particular node and if the node has not received any *HELLO* message since the updation timer, t_u was set, it delete its entry from its neighbor information table.

The proposed algorithm resolves the contention of two cluster heads that move into the transmission range of each other.

1. The cluster head with the largest weight or lowest ID, if the weights are equal, will keep its role as a cluster head, while the other one will change its

status to CM and broadcasts a *NOT_CLUSHEAD* packet to all its members.

2. All the nodes will update their CH list. The node whose CH list size now become zero, they will change their status to UD and will elect their new CH among themselves.

3.3.2.2 Global Re-Clustering

Any node in the network whose degree is greater than MAX_DEGREE or less than MIN_DEGREE can initiate *RECLUSCHK* packet.

As a clusterhead receives this packet,

- (a) It updates the variance in the packet if it receives the packet for the first time and retransmits it.
- (b) Otherwise, it discards the packet.

As the node that initiated the *RECLUSCHK* packet receives it,

- (a) it stores the variance and number of clusterheads traversed if it received the packet for the first time.
- (b) Otherwise, it updates the variance if the number of nodes traversed is greater than number of nodes previously stored.

As the timer, *network_traverse_time* at the node that initiated the *recluster_check* packet expires,

- (a) it checks the variance.
- (b) If variance is greater than the pre-determined threshold value, clustering is initiated.
- (c) Node initiates *RECLUSDO* packet in the whole network.
- (d) Nodes after receiving this packet stops sending data packet and any other control packet and re-transmits this re-clustering packet.
- (e) Node then resets all tables and becomes UD.
- (f) Same process of cluster formation (Phase 1) is repeated.

As variance of degree among CH's is checked in the network through *RECLUSCHK* packet, re-clustering in the network may or may not be initiated. But other nodes in the network are restricted from sending the *RECLUSCHK* packet in the network for *Recluster_check* time period, t_{chk} .

4. SUCCESSIVE VARIANCE CALCULATION

To quantitatively measure how well balanced the clusterheads are, the aim was to first calculate the variance of all CHs. If the difference between the heaviest loaded and the lightest loaded nodes is minimized, the average work execution time can be reduced, the energy of the nodes will be better exploited and the nodes lifetime can be extended.

Since, degree of all CHs were required at a single node to evaluate the variances, which is not possible in ad hoc network (distributed network), a formula is derived to calculate average and variance at each CH using previous average and variance.

Variance of a node can be evaluated in terms of variance till previous hop, average of nodes till previous hop and degree of that node.

Average at n^{th} node

$$A_n = \frac{\sum_{i=1}^n a_n}{n} \quad - (2)$$

$$\Rightarrow A_n = (a_1 + a_2 + \dots + a_{n-1} + a_n)/n$$

$$A_n = ((A_{n-1} * (n-1)) + a_n) / n$$

$$[As (a_1 + a_2 + \dots + a_{n-1}) = (A_{n-1} * (n-1))]$$

Variance at n^{th} node

$$V_n = \frac{\sum_{i=1}^n (D_i - A_n)^2}{(n-1)} \quad - (3)$$

$$\Rightarrow V_n = ((D_1 - A_n)^2 + (D_2 - A_n)^2 + \dots + (D_{n-1} - A_n)^2 + (D_n - A_n)^2) / (n-1) \quad - (4)$$

$$V_{n-1} = ((D_1 - A_{n-1})^2 + (D_2 - A_{n-1})^2 + \dots + (D_{n-1} - A_{n-1})^2) / (n-2) \quad - (5)$$

Subtracting (5) from (4),

$$\Rightarrow (n-1) V_n - (n-2) V_{n-1} = (D_1 - A_n)^2 - (D_1 - A_{n-1})^2 + (D_2 - A_n)^2 - (D_2 - A_{n-1})^2 + \dots + (D_{n-1} - A_n)^2 - (D_{n-1} - A_{n-1})^2 + (D_n - A_n)^2$$

\Rightarrow Simplifying the above expression, we get

$$V_n = ((D_n - A_n)^2 + (n-2) V_{n-1} + (n-1) (A_n - A_{n-1})^2) / (n-1) \quad \text{if } A_n > A_{n-1} \quad - (6)$$

And

$$V_n = ((D_n - A_n)^2 + (n-2) V_{n-1} + (n-1) (A_n - A_{n-1})^2) / (n-1) \quad \text{if } A_n < A_{n-1} \quad - (7)$$

Where

V_n is the variance of n nodes

V_{n-1} is the variance of (n-1) nodes

D_n is the degree of n nodes

A_n is the average of n nodes

A_{n-1} the average of (n-1) nodes

5. SIMULATION STUDY

In this section, the performance of the proposed LBRA with DEECA[12] is compared through simulations. The algorithm is implemented using GloMoSim 2.03. The wireless ad hoc network simulation area was 800*800m². The 802.11 protocol with a bandwidth of 11 Mbps was used at the MAC layer. The packet size was set to 100 KB. The simulation time lasted for 12000sec with a network of 50 nodes. Two Ray Ground was used as a radio propagation model and set the maximum energy of a node to 100 Joules.

5.1 Summary of Experimental Results

To measure the performance of the system, the focus was on the *network lifetime*. The control packets and route selection method was modified resulting in enhanced LBRA performance with respect to traffic load balancing and energy consumption balancing. The *network lifetime* was defined to expire when 20% of the nodes run out of energy.

To vary the load on the cluster heads, initially 30% of the nodes sent packets to their corresponding cluster heads, and then this percentage was gradually increased to 70% while each transmitting node was generating a packet every 3 seconds.

Since in DEECA, a cluster head compares its load with the average of its adjacent cluster heads, and when its load is more than the average, it asks its members to affiliate with other neighbouring CH's. Average of nodes can be misleading. Few of the neighbouring nodes may be highly loaded and few of them may be lightly loaded. Standard deviation of load is considered in LBRA. Also, in DEECA, nodes selected as CH will continue as CH's till they reach red threshold. While in LBRA, during re-clustering phase, each node is given an equal opportunity to be elected as CH.

Figure 1 shows that the network lifetime decreases when the load on the cluster heads increases. Figure 2 shows the impact of increasing the traffic on the network lifetime and energy. In this scenario, 50% of the 50 nodes were transmitting packets then, to vary network traffic slowly, from 1 packet per node every 5 seconds to 1 packet per node every 1 second.

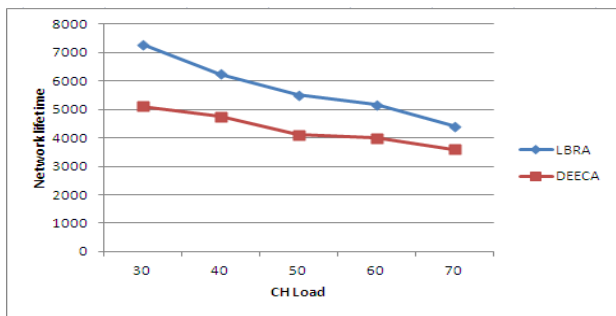


Figure 1. Network Lifetime vs CH Load

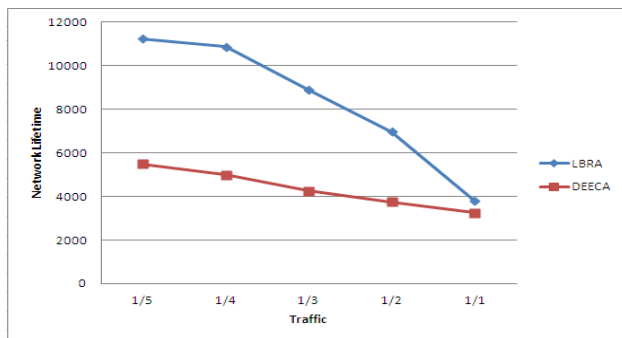


Figure 2. Network Lifetime vs Traffic

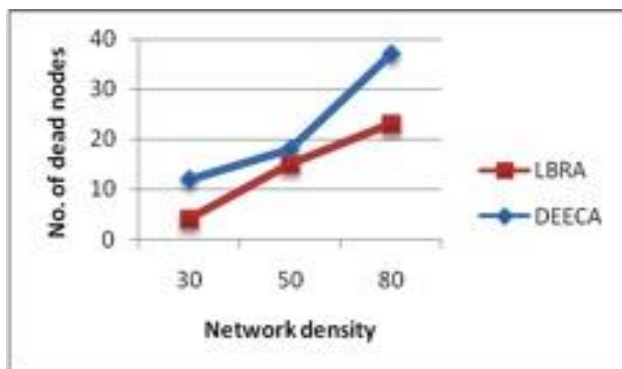


Figure 3. No. of dead nodes vs. Network Density

Figure 3 shows number of dead nodes when network density is varied. The number of dead nodes during the simulation tends to increase in DEECA as the number of nodes in the network is increased. This is due to the periodic approach followed by time driven re-clustering, which results in communication overhead.

Figure 4 shows packet loss percentage and Fig.5 shows packet delivery ratio when network density is varied. It is clear from the figure that the packet loss percentage and packet delivery ratio in LBRA is consistently better than the DEECA as the number of nodes is increased.

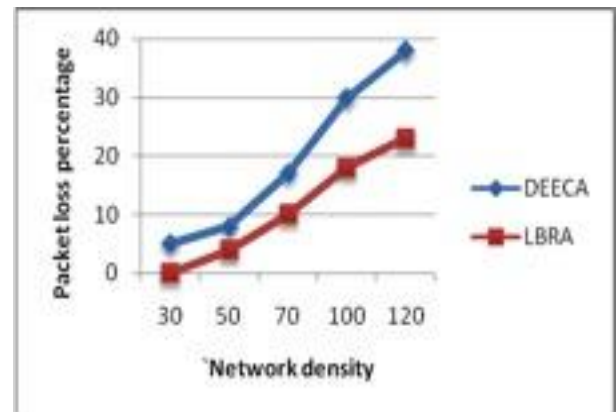


Figure 4. Packet Loss Percentage vs. Network Density

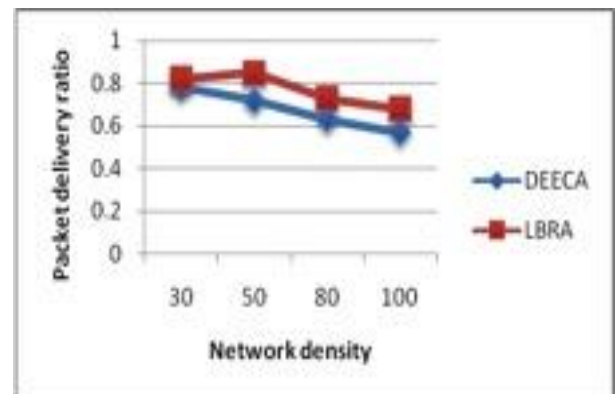


Figure 5. Packet Delivery Ratio vs. Network Density

Nevertheless, compared to DEECA, LBRA maintains a much higher NLT for the same reasons discussed earlier.

6. CONCLUSION AND FUTURE WORK

This paper proposes a Load Balanced Re-clustering Algorithm (LBRA) in which the primary objective was to increase the network lifetime by invoking re-clustering whenever the network load imbalances. The network performance can be enhanced by keeping the variance of degree of clusterhead within limits. The performance of the algorithm is evaluated by simulation. The result of simulation indicates that, compared with the load balanced routing algorithm DEECA, the proposed algorithm improves the network lifetime while varying several parameters such as traffic and clusterhead load. As a future extension, a mobility prediction method can be introduced to identify group mobility patterns and to provide steadier cluster formations. A distributed clustering approach can be used that enables each mobile node to anticipate the availability of its neighbors.

7. REFERENCES

- [1] Supreet Kaur, Varsha Kumari ,”A Proposed Model For Load Balancing in MANET”, IJCA, July 2014.
- [2] C.R.Dow, J.H.Lin, S.F.Hwang, Y.W.Wang, “An Efficient Distributed Clustering Scheme for Ad Hoc Wireless Networks”, IEICE Transactions on Communications, 2002.
- [3] J. Yu, P. Chong, “A survey of Clustering Schemes for mobile ad-hoc networks”, IEEE Communication Surveys, 7(1), pp. 32-48, March 2005.
- [4] B.McDonald, F.Znati, “Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks, “Proceesings of the 34th Annual Simulation Symposium, pp. 27-35, April 2001.
- [5] Mainak Chatterjee, Sajal K. Das, Damla Turgut. “WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks”, 2002.
- [6] S.K.Dhurandher and G.V.Singh, “Weight Based Adaptive Clustering in Wireless Ad Hoc Networks”, IEEE, 2005.
- [7] Nevin Aydin, Farid Na`it-Abdesselam, Volodymyr Pryyma and Damla Turgut, “Overlapping Clusters Algorithm in Ad Hoc Networks”, 2010.
- [8] Alan, D.A., Ravi Prakash, “Load Balancing Clusters in Wireless Ad Hoc Networks”, IEEE 2000.
- [9] Turgut, D.K. Das, S.K.Elmasri, R., “Balancing loads in Ad Hoc Networks”, 2003.
- [10] Abdel Rahman H. Hussein, Sufian Yousef, and Omar Arabiyat, “A Load-Balancing and Weighted Clustering Algorithm in Mobile Ad-Hoc Network”.
- [11] Rachida Aoudjit, Mustapha Lalam, Abdelaziz M’zoughi, Malika Belkadi and Mehammed Daoui, “Load Balancing: An Approach Based on Clustering in Ad Hoc Networks”.
- [12] Haidar Safa, Omar Mirza & Hassan Artail, “A Dynamic Energy Efficient Clustering Algorithm for MANETs ”.
- [13] Labdah AL Ghafran, Zulkefli Bin Muhammed Yusof, “Load-Balancing Technique in Clustered Mobile Ad-Hoc Networks”, 2013 International Conference on Advanced Computer Science Applications and Technologies.
- [14] Shashank Bharadwaj, Vipin Kumar, Ankit Verma, “A Review of Load Balanced Routing Protocols in Mobile Adhoc Networks”, International Journal of Engineering Trends and Technology – July to Aug Issue 2011.
- [15] D.Maheshwari, R.Nedunchezian, “Load Balancing in Mobile Ad Hoc Networks : A Survey”, International Journal of Computer Applications, December 2012.