ISSN 2047-3338

An Innovative Clustering Method for MANET Based on Cluster Convergence

Nima Karimi and Mohammad Shayesteh

***Abstract*—Mobile Ad hoc Networks have special properties and one of the most significant of those properties is nodes mobility which in part plays a considerable role in network's different parameters. Many efforts were made to make an infrastructure for these networks. In these infrastructures or clusters, a main node called clusterhead has a vital role in keeping the structure, routing and enhancing the efficiency of the network. In this article, an innovative clustering algorithm in mobile ad hoc networks is presented which has emphasis on a parameter that represents the convergence of the cluster with predict mobility. In this method, three parameters ,Relative speed, Ndm and battery power are used to calculate the primary weight of nodes, then a convergence coefficient can be achieved using the strenght of received signals of neighbors and predicting the nodes mobility . This value will be multiplied to the primary weight of the nodes afterwards, so that the final weight can be computed. In simulation, the proposed algorithm has been compared with WCA, MOBIC and the Lowest_ID algorithm. The results of simulation reveal that the proposed algorithm achieves the goals.**

***Index Terms*— Clustering Algorithm, Convergence, Cluster-head and Mobile Ad Hoc Network**

I. INTRODUCTION

A MANET is a multi-hop wireless network in which mobile nodes can freely move around in the network, leave the network and join the network. These mobile hosts communicate with each other without the support of any preexisting communication infrastructure. Typically, if two nodes are not within mutual transmission range, they communicate through intermediate nodes relaying their messages. In other words, the communication infrastructure is provided by the nodes themselves.

Through the nature of MANET, we have many challenges. The most important challenges are stability, routing and scalability. Clustering is the most way to improve the stability, routing and scalability. Have knowledge about the changes of node's status, can present useful information about the stability of it between its neighbors. This information is effective in clustering approach and in cluster head selection.

Nima Karimi is with Islamic Azad University of Bandar-abbas Branch, Iran (e-mail: nimakarimi@gmail.com)

Mohammad Shayesteh is with Islamic azad university of Banddar-abbas Branch, Iran (e-mail: shayesteh.au@gmail.com)

In wireless ad hoc network applications, such as outdoor teaching and the communications in the disaster area (the scenes of a fire, the flood, the earthquake and so on), a number of mobile hosts (MHs) are organized into several disjointed communication groups, which may move together and overlap with each other. Members within the same group have similar mobility patterns and can directly communicate with each other. Members of a group communicate with other nodes outside its group through the group clusterhead, which serves as a gateway to other groups. In the group mobility, the clusterhead equips with two network interfaces, one is used for local networks and the other is used for external networks. The local networks mean wireless ad hoc networks that are used in a group or between overlapped groups. The external networks denote the Internet, 2G, GPRS, and 3G…etc.

Clustering algorithms can be performed dynamically to adapt to node mobility [2]. MANET is dynamically organized into groups called clusters to maintain a relatively stable effective topology [1]. By organizing nodes into clusters, topology information can be aggregated. This is because the number of nodes of a cluster is smaller than the number of nodes of the entire network. Each node only stores fraction of the total network routing information. Therefore, the number of routing entries and the exchanges of routing information between nodes are reduced [3]. Apart from making large networks seem smaller, clustering in MANETs also makes dynamic topology appear less dynamic by considering cluster stability when they form [2]. Based on this criterion, all cluster members that move in a similar pattern remain in the same cluster throughout the entire communication session. By doing this, the topology within a cluster is less dynamic. Hence, the corresponding network state information is less variable [3]. This minimizes link breakage and packet loss.

Clustering is usually performed in two phases: clustering set-up and clustering maintenance. In the clustering set-up phase, clusterheads are chosen among the nodes in the network. The roles of clusterheads are coordinators of the clustering process and relaying routers in data packet delivery. After electing clusterheads, other nodes affiliate with its neighbor clusterhead to form clusters. Nodes which are not clusterheads are called ordinary nodes. After the initial cluster set up, reaffiliations among clusterheads and ordinary nodes are triggered by node movements, resulting reconfiguration of clusters. This leads to the second phase, the clustering maintenance.

As election of optimal clusterheads is an NP-hard problem [4], many heuristic clustering algorithms have been proposed [1]-[10]. To avoid excessive computation in the cluster maintenance, current cluster structure should be preserved as much as possible. however, any clusterhead should be able to change its role to an ordinary node to avoid excessive power drainage. In this way, the overall lifespan of the system can be extended.

The goal of this algorithm is to decrease the number of cluster forming, maintain stable clustering structure and maximize lifespan of mobile nodes in the system. To achieve these goals, we propose a new algorithm. In this algorithm, selection of a clusterhead is done during two stages. In the first, each node calculates its primary weight by using a new presented weighted function. second, each node predict mobility of its neighborhoods to calculates its convergence coefficient. convergence coefficient will be multiplied to the primary weight of the nodes afterwards, so that the final weight can be computed.. The result of simulation shows that the proposed algorithm provides better performance than WCA, MOBIC and Lowest_ID in terms of Clusterhead changes, Clusterhead lifetime and the average number of orphan clusters.

The rest of this paper is organized as follows. In Section 2, we review several clustering algorithms proposed previously. Section 3 presents the proposed algorithm for mobile ad hoc networks. The simulation of the proposed algorithm is given in Section 4. Finally, Section 5 concludes this paper.

II. RELATED WORK

A large number of clustering algorithm have been proposed according to certain environment and characteristic of mobile node in mobile ad hoc network to choose clusterhead. We will give each of them a brief description as follows:

1) Highest degree clustering algorithm [5] uses the degree of a node as a metric for the selection of clusterheads. The node with highest degree among its neighbors will be elected as clusterhead, and its neighbors will be cluster members. In this scheme, as the number of ordinary nodes in a cluster is increased, the throughput drops and system performance degrades.

2) The Lowest-Identifier algorithm (LID) [6] chooses the node with the minimum identifier (ID) as a clusterhead. The system performance is better than Highest-Degree heuristic in terms of throughput [4]. However, since this heuristic is biased to choose nodes with smaller IDs as clusterheads, those nodes with smaller IDs suffer from the battery drainage, resulting short lifetime span of the system.

3) Least Movement Clustering Algorithm [7]. In this algorithm, each node is assigned a weight according to its mobility. The fastest the node moves, the lowest the weight is. And the node with highest weight will be elected as clusterhead.

4) The Distributed Clustering Algorithm (DCA) [8] and Distributed Mobility Adaptive clustering algorithm (DMAC) [9] are enhanced versions of LID; each node has a unique weight instead of just the node's ID, these weights are used for the selection of clusterheads. A node is chosen to be a clusterhead if its weight is higher than any of its neighbor's weight; otherwise, it joins a neighboring clusterhead. The DCA makes an assumption that the network topology does not change during the execution of the algorithm. Thus, it is proven to be useful for static networks when the nodes either do not move or move very slowly. The DMAC algorithm, on the other hand, adapts itself to the network topology changes and therefore can be used for any mobile networks. However, the assignment of weights has not been discussed in the both algorithms and there are no optimizations on the system parameters such as throughput and power control.

5) MOBIC [7] uses a new mobility metric Instead of static weights; Aggregate Local Mobility (ALM) to elect clusterhead. ALM is computed as the ratio of received power levels of successive transmissions (periodic Hello messages) between a pair of nodes, which means the relative mobility between neighboring nodes.

6) The Weighted Clustering Algorithm (WCA) [4] is based on the use of a combined weight metric that takes into account several parameters like the node-degree, distances with all its neighbors, node speed and the time spent as a clusterhead. Although WCA has proved better performance than all the previous algorithms, it lacks a drawback in knowing the weights of all the nodes before starting the clustering process and in draining the clusterheads rapidly. As a result, the overhead induced by WCA is very high.

Most of previous algorithms were using only one metric for clustering purposes. Therefore, the resulted clustering topology fits just in terms of that specific metric [10]. As Mobile Ad Hoc networks are generally complex and dynamic networks, existing of only one specific metric can not reference the whole situation of the network. Those types of clustering topologies which are optimal in terms of just one metric are suitable for particular scenarios and have poor performance in other scenario. For these reasons, we use different metrics in our algorithm to select the clusterhead. On the other hand, in clustering approaches based on weighted functions such as WCA, efforts are concentrated to select the best node among neighborhood nods by using of available metrics. In these methods, only those nodes are selected as clusterhead which have better properties than other neighborhood nodes such as more rest battery power, having more neighborhoods and less average distance from neighborhoods. but in most of them do not consider to the movement model of the clusterhead toward the other nodes of the cluster. This factor causes the formed clusters be unstable and increase the overload of cluster reelection process. While in approaches within move sensitive clustering category such as MOBIC, clustering is done based on the nodes movements or approaching/escaping to each other. in this approaches, the main parameter for clustering is the mobility of nodes and therefore other parameters such as the energy of the battery, the number of neighborhoods and so on, are not considered.

In the proposed approach we present an optimal method without appearing previous methods problems by combining the useful characteristics of those methods in order to reach to the goal of our algorithm.

III. THE PROPOSED ALGORITHM

In the algorithm presented in this paper, selection of a clusterhead is done during the calculation of primary weight and final weight. In the first, each node calculates its primary weight by using a new presented weighted function. second, each node calculates its convergence coefficient with predict mobility of its neighborhood. Then based on the primary weight final weight is calculated. Finlay the node with higher weight selected as clusterhead.

A. Setup Procedure

First, we allocate IDs for the nodes. In the proposed algorithm, each node Ni (member or clusterhead) is identified by a state such as: Ni (idnode , idCH , flag , Weightp), it also has to maintain a 'node_table' wherein the information of the local members is stored. However, the clusterheads maintain another clusterhead information table 'CH_table' wherein the information about the other clusterheads and member node is stored.

In complex networks, the nodes must coordinate between each other to update their tables. The Hello messages are used to complete this role. A Hello contains the state of the node; it is periodically exchanged either between clusterheads or between each clusterhead and its members in order to update the 'CH_tables' and the 'node_tables' respectively.

We define a flag for every node which determine their role. The value of flag is 3 if the node is the clusterhead, is 1 if the node is an ordinary node, is 2 if the node is a gateway and is zero if the node has an undetermined status.

A. Weighted Function

To enhance stability of clusters we must find out problems that cause stability to be decreased and as a result cause a cluster to disappear. If we know and solve these problems, we can enhance stability of the clusters as much as possible.

The first parameter which causes clusters to disappear is Excessive battery consumption at a clusterhead. In MANETs, the nodes not only bear the responsibility of sending and receiving information, but also carry out routing for packages. As a result they consume a high rate of power.

As a result a clusterhead must have the following conditions:

• It must have a high existence of battery power.

• It must require a lower battery power for interaction with neighbors.

To meet the first condition, the amount of battery power is taken into account as one of the factors for calculation of weight. To meet the second condition, we can choose a node as a clusterhead, which has less distance with its neighbors during neighborhood duration.with using of convergence coefficient we can achieve this goal. Higher convergence coefficient means that in the future the average distance between clusterhead and its neighbors is smaller than other node. The higher convergence coefficient means that the less transmission power the node requires for interaction and communication with its neighbors and as a result it consumes less battery power.

The second parameter which causes the clusters be unstable is the mobility of nodes. In the proposed algorithm for creating stable clusters, in the first, the previous mobility of nodes intended, which is accessible by calculating parameter Relative speed. In the second, stable clusters are created through calculating convergence coefficient of nodes.

The used parameters in weighted function for giving a primary weight to nodes (weightp) include:

- Battery remaining (Br): every node which wants to be the clusterhead should have threshold power Bd. A clusterhead consumes more energy in a cluster comparing with an ordinary node. In addition, we prefer to choose a more powerful node to play its role as a clusterhead because such a node looses its energy later results in the late starting of new clusterhead selection process and therefore increases the stability of clusters. Equation (1) shows that the each node how to calculate battery remaining of itself.

$$B_r = \mathrm{B_d} + \frac{N_{dm}}{N} B_d \qquad (1)$$

- *Number of nodes moving towards a node (N_{dm})*

- *Relative speed (S):* The relative mobility of the node with its neighbors, which means how long node has spent their time beside the node. A lower relative speed simply means that the neighbors of a certain node has spent a longer time in its transmission range, we conclude that the mentioned node has a more stable situation. The relative speed is calculated by Eq. (2).

$$S = \sum_{i=1}^{N} \frac{S_L - S_F}{T_L - T_F} \qquad (2)$$

{n is the number of node's neighbors}

Where SL is the signal strength of last packet reception, SF is the signal strength of first packet reception, TL is the time of the last packet reception and TF is the time of the first packet reception.

Each node uses the above mentioned three parameters to calculate its primary weight (weightp). The Eq. (3) shows how the nodes calculate weightp.

$$Weight_P = c_1 B_r + c_2 N_{dm} + c_3 S \quad (3)$$

In the Eq. (3), ci(s) are the weight factors of normalization.

B. The calculation of final Weight

After Each node calculate its primary weight, to get final weight should calculate the convergence coefficient. For calculate the convergence coefficient we should first predict the neighbors mobility. We predict neighbors mobility based on last two received signal power. For predicting the next signal power received by nodes, we use linear extrapolation method. In this method in order to obtain the signal power at the time t+1, we use the powers of signal at the times t and t-1. Eq. (4) shows how to calculate the power of a signal received at the time t+1:

$$P_{t+1} = P_{t-1} + \left(\frac{T_{t+1} - T_{t-1}}{T_t - T_{t-1}} * (P_t - P_{t-1}) \right) \quad (4)$$

After predicting the power of the received signal of all neighbors at the time t+1, we can calculate the convergence coefficient of this node by the Eq. (5):

$$C = \frac{N_{dm}}{N} \sum_{i=1}^{N} P_{t+1} - P_t \quad (5)$$

Each node use Eq. (6), for calculating the final weight according to convergence coefficient:

$$Weight_F = C * weight_p \quad (6)$$

The node with highest final weight in its Neighbourhood selected as clusterhead.

In this way, we can select suitable clusterhead for all nodes not only by considering the suitable condition of clusterhead in the network but also with regarding to the node status comparing with the clusterhead. This leads to better local selection of the clusterhead resulting in more stable clusters creation preventing future cluster reform rippling.

Table 1 show the messages with its description used in proposed algorithm.

Table 1: Messages used in the algorithm

Message	Description
Hello (id_{node} , id_{CH} , flag, $weight_p$)	To update the tables of the nodes
Best(id_{node} , id_{CH})	Offer the node to be a clusterhead
Join_request(id_{node} , id_{CH})	To affiliate a cluster
Join_accept(id_{node} , id_{CH})	The node accepts the welcome_ACK
CH_Wel(id_{node} , id_{CH})	The CH accepts a Join_Request
CH_NWel(id_{node} , id_{CH})	The CH rejects a Join_Request
CH_ACK(id_{node} , id_{CH})	The CH adds the node as a member
CH_info(id_{node} , id_{CH})	The CH accepts the presence of a new CH in the network
CH_change(id_{CH})	The CH notifies a CH change
Leave(id_{node} , id_{CH})	The node leaves the cluster

C. New Arrival Nodes Mechanism

Once a wireless node is activated, its idCH field is equal to NULL since it does not belong to any cluster. The node continuously monitors the channel until it figures out that there is some activity in its neighbourhood. This is due to the ability to receive the signals from other present nodes in the network. The node still has no stable state, thus its state is not full identified. In this case, it broadcasts a Join_Request in order to join the most powerful clusterhead. Thus, it waits either for a CH_Wel or for a CH_NWel.

When the entry node does receive neither CH_Wel nor CH_NWel . If this persists for certain number of attempts, the node declares itself as an isolated node, and restarts by broadcasting a new Join_Request after a period of time.

We note that just the clusterheads may response by a CH_Wel or CH_NWel; the ordinary members have to ignore any Join_Request received even if they are in the transmission range of the new entry node. This allows simplifying the management of the clusters.

In the case where the node receives a response (CH_Wel or CH_NWel), it does not take immediately any decision, this allows the node to be certain that it has received all the responses from all the neighboring clusterheads. The CH_Wel and CH_NWel messages do not indicate that the clusterhead has added the node to its table; they just signify that the clusterhead is waiting for a Join_Accept in order to add the node to its table. When the node receives multiple CH_Wels, Based on the primary weight of clusterheads calculate the final weight of them and select the node with highest final weight as the clusterhead. After that, it sends a Join_Accept to the chosen clusterhead and waits for CH_ACK from this CH. The CH_ACK has to contain a confirmation that the id_{node} has been added to the CH_table. Thus the node can fully-define its state. The reason that we use four ways to confirm the joining procedure is to prevent other clusterheads that they can serve the entry node to add this node to their tables and cause conflicts.

In the case where the node was just receiving CH_NWels, it considers these responses as rejection messages from the clusterheads. This may occur when the clusterheads are saturated and decide to reject the adhesion of new nodes. When the number of attempts reaches a certain value, the node prefers not to stay isolated, thus it declares itself as clusterhead.

D. Clusterhead Nodes Mechanism

A clusterhead has an id_{node} field is equal to id_{CH} field. As a clusterhead, the node calculates periodically its weight, thus it sends periodically Hello messages to its members and to the neighboring clusterheads in order to update the node_tables and CH_tables respectively. The clusterhead must monitor the channel for Leave, Hello and Join_Request messages. in the proposed algorithm this operation is limited to clusterhead to allow easier management on clusters.

When the clusterhead receives a Join_Request (id_{CH}=NULL) from a new arrival node or a Join_Request (full state) from a node which belongs to another cluster, the clusterhead can accept or reject the request basing on its capacities. This procedure gives more flexibility to the members by allowing them to leave a weak clusterhead and join another one which seems stronger than the current clusterhead. It may not be possible for all the clusters to reach the cluster size λ. We have tried to reduce the clusters formed by merging the clusters that have not attained their cluster size limit. However, in order not to rapidly drain the clusterhead's power by accepting a lot of new nodes, we define thresholds which allow the clusterhead to control the number of nodes inside its cluster.

The re-election is not periodically invoked; it is performed just in case of a higher received weight, it allows minimizing the generated overhead encountered in previous works As we explained above. the re-election may not result a new clusterhead, it depends on the stability of the new node for playing the clusterhead's role.

In the proposed algorithm clusterhead will check regularly incoming messages from neighboring nodes. if clusterhead received a message that contains higher primary weight from his weight, then it check the relative mobility with the desired node, if its relative mobility to this node were in the first group and all of the cluster members exist in neighboring of this node, assign clusterhead role to the desired node. The node does it with save the ID of this node in its CH_ID field.

Then send a CH_info message to new clusterhead to declare that this node as a new clusterhead selected. Then copy their tables in to new clusterhead and send a CH_change message to neighboring nodes to defines a new clusterhead. in This new approach selecting the new clusterhead is based on stability of it in the cluster. In this case where a new clusterhead is elected, the procedure is soft and flexible in order not perturb the clusters while to copying the databases from the old clusterhead to the new clusterhead.

E. Member Nodes Mechanism

After joining a cluster, the node declares itself as a member of this cluster. Hence, it calculates periodically its weight and sends periodically Hello messages to its clusterhead. As a member, this node should just handle the Hello, the CH_change and the CH_info messages. This allows optimizing the resources (bandwidth, battery, etc) and minimizing the job of the nodes.

When the node receives a Hello from its clusterhead, the node has to update its node_table. When the node receives Hellos from the neighboring clusterheads, the node has the possibility to migrate to another clusterhead if there is a Hello which has a higher weight than the current clusterheads weight, Member node get this decision by calculating the final weight of the new clusterhead. it sends a Join_Request to the clusterhead which is Hello's source and continues as a member of the current clusterhead until the reception of CH_ACK. In this case, the node can send a Leave_Request to the last clusterhead. This method allows us to minimize the number of the formed clusters in the network.

When the node member receives a CH_info message as a result of the re-election procedure, thus it realizes that it is going to become the new clusterhead in the cluster. When a node member does not receive any message from its clusterhead, it considers that the clusterhead has gone brusquely down; in this case, the nodes have no choice and must restart the clustering setup procedure.

IV. SIMULATION AND RESULTS

In this paper we use GloMoSim tool for simulation. The simulation environment is a Mobile Ad Hoc Network consists of 20 to 100 nodes in an 800*800 area. We assume that each node will be activated by a 2.4GHz radio frequency. The simulated area is considered as a two dimensional square and nodes move freely throughout the area. The movement of nodes has been simulated according to Random Waypoint model.

In order to evaluate the performance and efficiency of the proposed algorithm, a set of simulations were operated and duration of them was 1200 seconds. We select a set of parameters to show the efficiency of our algorithm. Our proposed algorithm was compared with WCA, MOBIC and Lowest_ID method which is the most famous clustering algorithms. These parameters include:

A. Clusterhead changes

Fig. 1, Fig. 2 and Fig. 3 show that the clusterhead changes in the proposed algorithm less than the WCA, MOBIC and Lowest_ID algorithms that leads to long life of clusters so will have more stable clusters. The reason is that the proposed algorithm selected such as the node as the clusterhead that have more Presence and battery power therefore more time is left as clusterhead. Fig. 1 shows the Average number of clusterhead changes against the speed of node, Fig. 2 shows the average number of clusterhead changes against the transmission range and Fig. 3 shows the count of clusterhead changes against the number of nodes.

B. Clusterhead lifetime

Fig. 4 and Fig. 5 show that the clusterhead lifetime in the proposed algorithm higher than the WCA, MOBIC and Lowest_ID algorithms. Fig. 4 shows the average lifetime of clusterhead against the speed of node and Fig. 5 shows the average lifetime of clusterhead against the transmission range.

C. The average number of clusters

As you can see in Fig. 6, the number of formed clusters is increased by increasing the number of nodes. Fig. 6

shows the average number of clusters against the number of node.

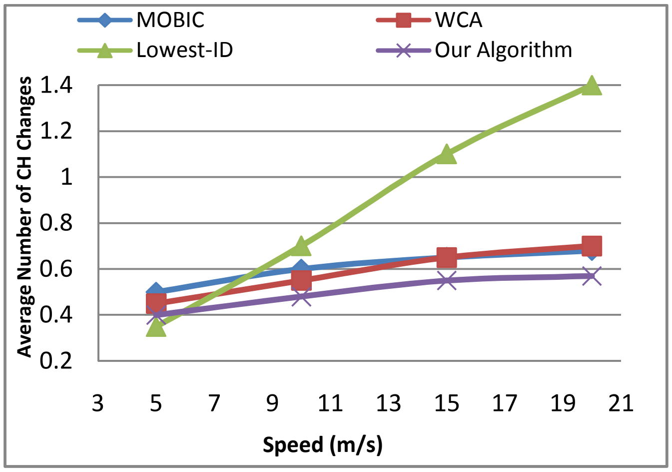

Fig. 1. the Average number of clusterhead changes vs the speed of node

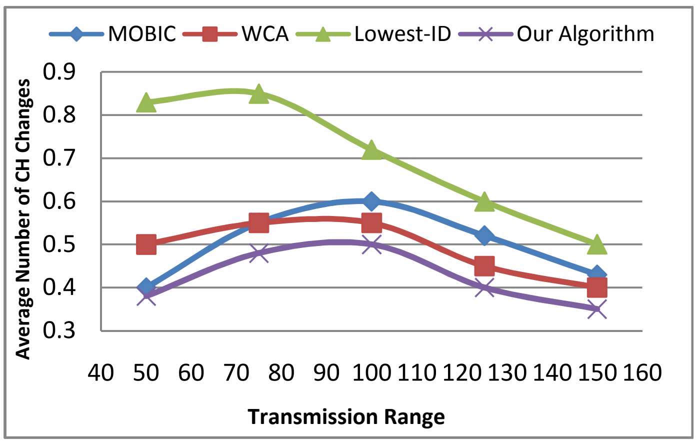

Fig. 2. the Average number of clusterhead changes vs. the transmission range

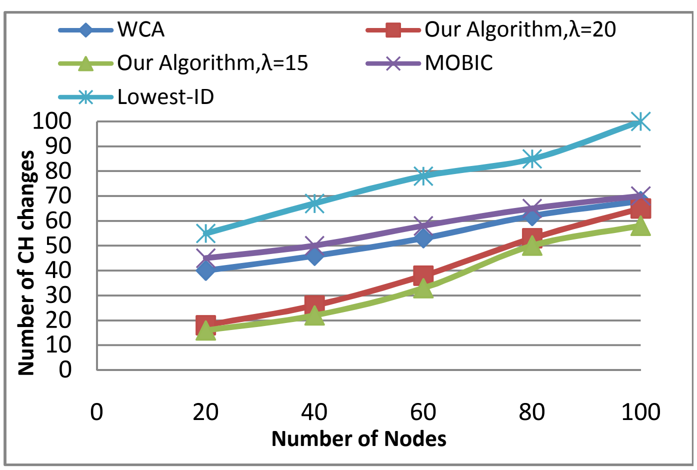

Fig. 3. Count of clusterhead changes vs. the number of nodes

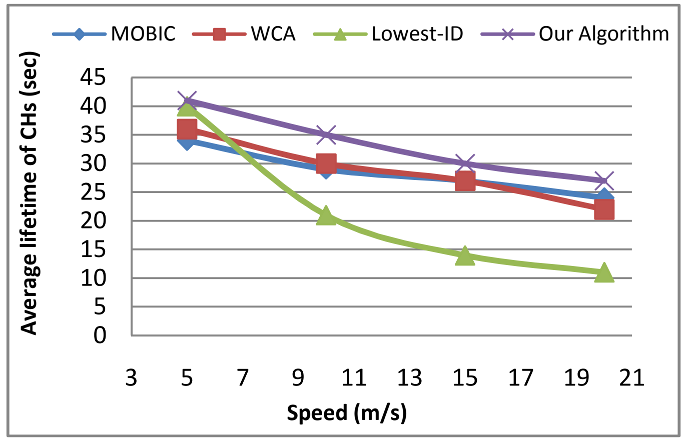

Fig. 4. Average lifetime of clusterhead vs. the speed of node

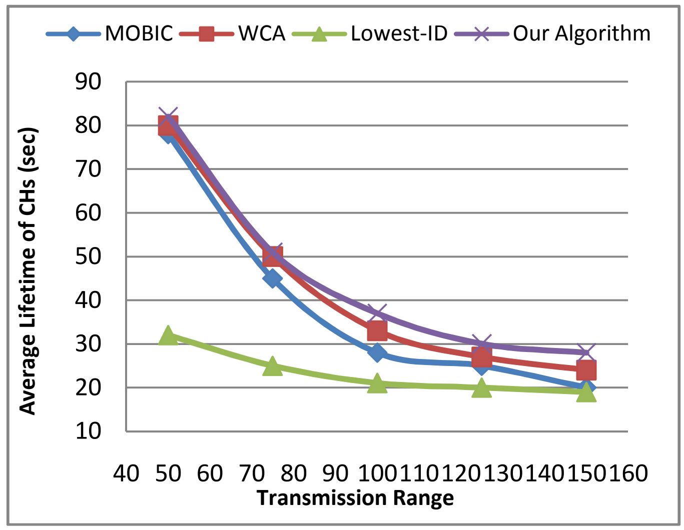

Fig. 5. Average lifetime of clusterhead vs. the transmission range

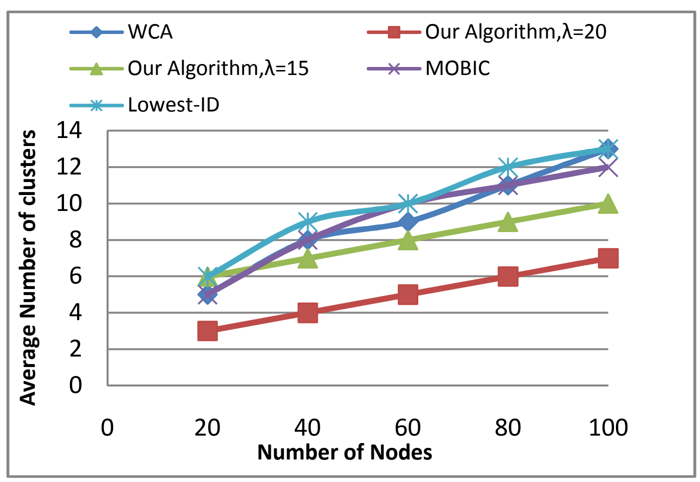

Fig. 6. Average number of clusters vs. the number of node

D. The average number of orphan clusters

As you can see in figure 7, the number of single node clusters or orphan clusters in our algorithm is less than

WCA, MOBIC and the Lowest_ID algorithms. The reason is that in our algorithm, the clusterheads are selected from central safe area.

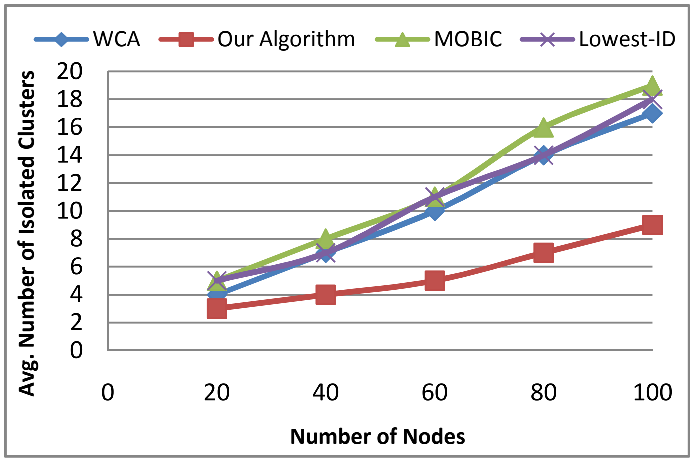

Fig. 7. Number of single node clusters or orphan clusters

V. CONCLUSION

In this paper, we have presented a new clustering algorithm in Mobile Ad Hoc Network. In this algorithm selection of a clusterhead is done during the weighted function. In the first, each node calculates its primary weight by using a new presented weighted function. In the second, each node calculates its convergence coefficient with predict neighborhoods mobility. A number of parameters of nodes were taken into consideration for assigning weight to a node. The proposed algorithm chooses the cluster-heads based on the information of neighbor nodes, and maintains clusters locally. Also it has a feature to control battery power consumption by switching the role of a node from a cluster-head to an ordinary node. We assumed a predefined threshold for the number of nodes to be created by a clusterhead, so that it does not degrade the MAC function and to improve the load balancing. We conducted simulation that shows the performance of the proposed enhancement clustering in terms of the average number of clusters formation, stability of clusters, and lifetime of a clusterhead. We also compared our results with the WCA, MOBIC and Lowest_ID. The simulation results show that our enhancement clustering algorithms have a better performance.

REFERENCES

[1] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. IEEE Journal on Selected Areas in Communications, 15(7):1265-1275, Sept. 1997.

[2] A. B. McDonald and T. F. Znati. A mobility-based framework for adaptive clustering in wireless ad hoc networks. IEEE Journal on Selected Areas in Communications, 17(8):1466-1486, Aug. 1999.

[3] C. E. Perkins, editor. Ad Hoc Networking. Addison-Wesley, 2001.

[4] Chatterjee, M., Das, S., and Turgut, D., "WCA: a weighted clustering algorithm for mobile ad hoc networks," Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), 5, 2002, pp. 193-204.

[5] Gerla M., Tsai J. T. C., "Multicluster, Mobile, Multimedia Radio Network," ACM/Baltzer Wireless Networks Journal 95, vol. 1, Oct. 1995, pp. 255-265.

[6] Baker D.J., Ephremides A., "A distributed algorithm for organizing mobile radio telecommunication networks," Proceedings of the 2nd International Conference on Distributed Computer Systems, Apr. 1981, pp. 476-483.

[7] Basu P., Khan N., Little T. D. C.. A mobility based metric for clustering in mobile Ad hoc networks[A]. proceedings of IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile computing[C], phoenix, A Z, April 2001:413-418.

[8] Basagni S., "Distributed clustering for ad hoc networks," Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, Jun. 1999, pp. 310-315.

[9] Basagni S., "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," Proceedings of Vehicular Technology Conference, VTC, vol. 2, fall 1999, pp. 889-893.

[10] Hui Cheng, Jiannong Cao, Xingwei Wang, Sajal K. Das, Stability-based Multi-objective Clustering in Mobile Ad Hoc Networks, The Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, August 7–9, 2006, Waterloo,Ontario, Canada © 2006 ACM.