

Nodes clustering using Fuzzy logic to optimize energy consumption in Mobile Ad hoc Networks (MANET)

Ehsan Amiri^{a*}, Ali Harounabadi^b and Seyed Javad Mirabedini^b

^aDepartment of Computer Engineering, Islamic Azad University, Bushehr Science and Research Branch, Iran

^bDepartment of Computer Engineering, Islamic Azad University, Central Tehran Branch, Iran

ARTICLE INFO

Article history:
Received June 25, 2012
Received in Revised form
August, 25, 2012
Accepted 28 August 2012
Available online
5 September 2012

Keywords:
MANETs
Fuzzy
Clustering
Energy consumption

ABSTRACT

Mobile Ad hoc Network (MANET) consists of some nodes, which are randomly placed in operational environment. This type of network does not have any infrastructure and nodes are completely dynamic and moveable. Each node can contact with the other in-range nodes. One of the disadvantages of MANET is the low lifetime of nodes power in which the energy consumption of the network goes up due to connectivity overheads. In this paper, by presenting a Fuzzy clustering algorithm, we endeavor for a more optimized energy consumption of this network by reducing the number of re-election of clusterheads (CHs) and reducing the re-affiliation of nodes and also by selecting the best node as CH amongst current members of each cluster. The result of simulation shows that our algorithm performs better than other existing algorithm.

© 2012 Growing Science Ltd. All rights reserved.

1. Introduction

MANET is a set of mobile nodes in wireless networks, which dynamically organize themselves without using any pre-available infrastructure. MANET applications as usual are peer to peer rather than client-server. MANET is constructed to support a certain program, so it is a program-based network (Hogje, 2007). We can call ad hoc network, as a need of now and future, because of its massive usage. MANET is easily allowing other devices to connect or disconnect from network. Some applications of this network are Tactics networks, Military services (Hogje, 2007), Emergency services (exploring operation, rebuilding events, replacing constant infrastructure, supporting doctors and nurses in hospital), commercial environments (Hogje, 2007), distributed and associated computing (Hogje, L. 2007), measuring and game (Hogje, 2007; Hoebeke et al., 2004), home networks and investing.

* Corresponding author. Tel: +98-917-9891610
E-mail addresses: amirehsan@gmail.com(E. Amiri)

Topology management is a key factor of network management in mobile ad hoc networks. The main goal of topology management is to save energy while connecting to network. Topology management includes knowing physical and logical relationships among nodes. Another sense of topology management is to have only one sub-branch of nodes to actively participate in network, so it makes fewer relationships and saves energy in nodes (Boleng, et al., 2002). Clustering or grouping nodes to reduce energy consumption is one of the ways for network topology management. There are many different ways and methods for network clustering. In fact, the purpose of clustering is to group exclusive nodes in some methods. The well-known special nodes as clusterheads are responsible for cluster forming, keeping network topology and resource allocation to all group members. Finding cluster head is an NP-hard problem (Chatterjee et al., 2002). Some nodes act as gateway for setting relationships with neighboring clusters, and the other nodes are known as cluster members.

Clustering has three main advantages; first, re-using of resources to increase the system capacity. For example, if clusters have no overlapping, they can make use of a common transmission range. Second, a set of clusterheads and gateways is able to use a virtual backbone for intra-clustering routing, therefore making separation of routing information can be restricted to a set of nodes and third, the clustering structure makes the MANET to be smaller and more stable from mobile terminals point of view (Hogie, 2007). Zadeh (1975) originally proposed the fuzzy set theory and it has been developed for expanded linguistic values (Zadeh, 1975). The linguistic values are terms, which are used instead of numbers and fuzzy set theory (Ghazanfari & Rezaie 2006).

In this paper, we propose a new optimized node-clustering algorithm based on fuzzy theory to minimize the energy consumption and to minimize the re-election of clusterheads along with the number of clusterheads and the re-affiliation of all nodes. The proposed model improves the communicative overhead problem of existing algorithms by minimizing the communicative overhead of message propagation through the network via goal-oriented broadcasting of messages. Keeping connectivity among nodes is another means of saving the network energy. In this method, we decrease the number of re-elections by selecting the best node for clusterhead using fuzzy methods and do not re-affiliate among connected nodes trying to keep the connection of connected node as much as possible. We also reduce the number of re-affiliated nodes by binding the node to become current cluster. The organization of this paper is as follow: in section 2 we review related works and limitations, in section 3 we describe our algorithm, in section 4 we show the results of simulation of our algorithm and finally, in section 5 we compare these results with other famous algorithms.

2. Related works

The concept of nodes clustering in the logical clusters was introduced first by Baker and Ephremides (1981). They developed a distributed bootstrap algorithm for propagating and maintaining communication when the mobile node as well as the node is destroyed. Several clustering algorithms have been developed for wireless networks and MANET. In this section, we describe some of the most popular algorithms.

2.1 Lowest ID Algorithm(LID)

Low identification algorithm was proposed by Baker and Ephremides (1981) and it was named as the Lowest ID (LID) algorithm. In this algorithm, every node is assigned with a unique non-negative identification number, which is a decision factor for the status of a node shown as ID. At the same time, it also receives messages from its neighbors. If a node listens to all IDs, which are higher than its own ID, then it declares itself as the cluster head among its immediate neighbors. In addition, the neighbor nodes whose status is not yet decided become the members of the newly selected head. This process is repeated until all nodes are assigned with the role of a head or a member of a cluster. This algorithm provides a solution to avoid the hidden terminal problem of wireless communication as well as how to make a quick recovery from the node or link failure in the dynamic network. This

algorithm does not allow two cluster heads to become neighbors at any point of time. So the gateway nodes provide the path for inter cluster communication.

2.2 Highest Connectivity Algorithm(HC)

Highest connectivity algorithm was proposed by Parekh (1994). This algorithm aims to reduce the number of clusters in the network. In every cluster, there exists a cluster head, which belongs to the dominating set. If N_i represents the set of adjacent nodes of a particular node i , then the degree of connectivity of i is represented as $D_i = |N_i|$, where $|N_i|$ is the cardinality of i . In the HC algorithm, a node having the highest degree of connectivity is selected as the cluster head and the adjacent node whose status is not yet decided becomes the member of the selected cluster head.

A high degree of connectivity can reduce some cluster head causes having more nodes, absolutely. Here, performance means reducing latching in setting connection from cluster head nodes. Although this algorithm has low operational potential, each cluster is associated with a source, which is common among all groups so increasing the number of nodes in each cluster causes some more potential operational and finally weakens the performance of the whole system. Therefore, the mobility of node can cause change in connectivity degree of node, which often leads to re-election of clusterhead. Therefore, the cost of keeping cluster is higher than configuration counterpart. The weak stability of cluster in this algorithm reduces its usage even though this algorithm reduces the delay in packet communication in real word.

2.3 Distributed Mobility Adaptive Algorithm(DCA, DMAC)

This algorithm proposed by Basagni et al. (1997), where distributed clustering algorithm is mobility adaptive and really distributed in nature. This algorithm is based on total weight and cluster formation algorithm. Each node with one parameter as weight (i.e. ≥ 0) makes decision about the role of the node. The weight of node can be as a function of transmission range or node mobility.

This algorithm does not allow the change in topology of network and one node has the most weight in one-hop neighbor select as a cluster head. An ordinary node opts to join a cluster head with higher weight by showing several cluster head in its proximity. This algorithm performs well for cluster formation where the maintenance of the clusters in the presence of node mobility is not specified. DCA is mostly applicable for a static or a quasi static network.

In the distributed and mobility adaptive clustering (DMAC) algorithm, the cluster formation process is almost the same as DCA (Boleng, 2002). However, the non-mobility of nodes during the execution of the algorithm is eliminated here, making it truly compatible with mobility adaptive definition. DMAC claims to be the best algorithm for cluster formation and maintenance by being mobile nodes. This algorithm starts by this assumption that each node has one special known ID and with the same weight and status in the network as well as the same for its one-hop neighbors. This proves that the cluster head is selected only with the knowledge of its local topology.

2.4 Weighted Clustering Algorithm (WCA)

Chatterjee et al. (2002) proposed a weighted clustering algorithm (WCA), where re-election with several events is taking place. For example, when there is a demand for it, the node parameters like the degree of connectivity, mobility, transmission power and available battery power are considered for selection of a cluster head and are given different weights depending on the network scenario. For example, in sensor network, energy is the biggest limitation and battery energy can make higher weight. The linear combination of weight of each node can be calculated in this way:

$$wc = w_1 D_v + w_2 M_v + w_3 T_v + w_4 P_v,$$

where D_v is the difference of the cardinality of the neighbors of a node v represented as $|N_v|$ and a scenario based threshold δ which limits the upper bound for the number of members in a cluster, i.e. $D_v = ||N_v| - \delta|$. M_v is the average speed of a node since the last reelection taken place and $T_v = \text{dist}(v, v)$, where v is the set of neighbor nodes of v . P_v is the cumulative time for which the node remains as the cluster head. All these parameters are normalized to a predetermined value and the weighing factors are chosen so that $\sum_{k=1}^4 w_k = 1$. Initially, the node with the lower weight is elected as the initial cluster head, and its neighbors by one-hop distance are elected as the members of that cluster. This process rejects the other members of being detected until all nodes maintain a determined status like cluster head or one of other cluster heads for limiting top end for some of nodes, which are members inside a cluster. This kind of limitation of member nodes improves efficiency of MAC layer. Cluster head is in two-node powers situation. It works in low power for intra-cluster relationships and high power relationship for inter-cluster relationships. Altogether, the limitation of this algorithm in the common global weight amount is hiding inside the network. To find distributed solution of this algorithm, a significant portion of information of nodes for finding the smallest saved weight is changed.

2.5 Distributed Fuzzy Score-based Clustering Algorithm for Mobile Ad-hoc Networks(DFSCA)

The purpose of this algorithm proposed by Adabi et al. (2008) is to increase the lifetime of mobile nodes in system while the number of clusters remains at minimum level. For this aim, they designed and distributed fuzzy scoring clustering algorithm. Each node calculates its own score based on four parameters: Battery Remaining, Number of Neighbors, Number of Members, and Stability depends on fuzzy method. Each node independently selects the clusterhead according to neighboring nodes. So the election of clusterhead in distributed way is calculated by low necessary over loading and it is doing for current node information. The election of cluster head in this algorithm includes the following five steps:

- Step 1: Updating the neighborhood table based on receiving the score-value messages, which would be done particularly for each node
- Step 2: Calculating score-value based on fuzzy method and broadcasting the score-value by a message
- Step 3: Checking the neighborhoods of each node to choose the node with the highest score to be chosen as clusterhead
- Step 4: Broadcasting nodes' types to all neighbors through a type-message
- Step 5: Calculating the new amount of timer where originally maintained an acceptable optional value

3 Fuzzy clustering algorithm of nodes

In this section, we describe the proposed algorithm after describing the basic steps of the algorithm and the necessary factors.

3.1 Basics

A mobile adhoc network has been consisted of some nodes, which are randomly distributed in an operational environment. At first, the nodes do not have any data about the existence of other nodes within the network. We can assume each node as a vertex of a graph linked to each other. In clustering, the primary objective is to find the efficient number of clusters, supposing clusters connectivity and similar to efficient dividing of graph a NP-hard issue. In clustering one, node is elected to be the clusterhead and the others in transmission range with this clusterhead, are selected as cluster members and new connection between clusterhead and each cluster member is made in the

next step. Effective election of the number of clusterheads is the main important issue, which further work could be made on it in the future. Fuzzy system consists of three parts (Dastgheib et al. 2011): fuzzification, inference engine and defuzzification. Fig. 1 shows the fuzzy system factors are used in this paper.

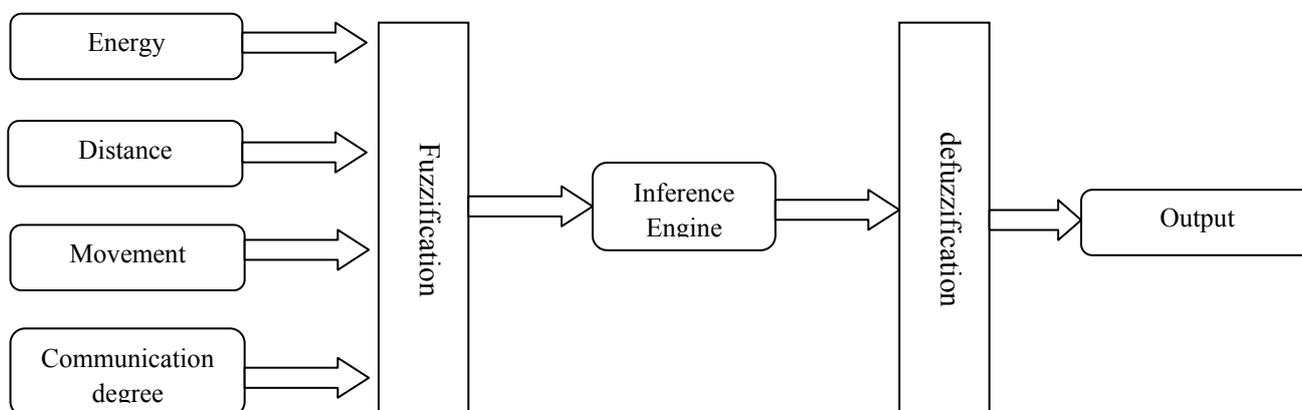


Fig. 1. Fuzzy system

Fuzzification:

In this stage, input variables by using of membership functions are converted into linguistic values to determine the membership degree.

Inference engine:

The rule based consists of series of regulations as IF-THEAN form; here we have $4^3 = 64$ deferent rules. Fig. 2 shows some samples of used rules in this research. Input fuzzy parameters apply to existing rules in the rules base and all rules adaptable to input parameters are activated then the output of all activated rules aggregate together.

Defuzzification:

The inputs of this phase are those linguistic values and the outputs are non-fuzzy values. We use defuzzification formula as follows,

$$W = \frac{\sum_{l=0}^{80} \left[\left(\sum_{i=0}^3 k_i \right) \times O_l \right]}{\sum_{j=0}^{80} O_j}. \quad (1)$$

3.2 The basic factors

In this plan, we focus our attention on factors such as remaining energy of each node as well as the connectivity degree of the same node (the number of neighbors) and the distance of node to its neighbors and then its mobility degree. These factors are briefly discussed as follow:

Remaining energy

We know that each mobile node has a battery with limited power and energy saving for batteries is essential for node survival and network maintenance. On the other hand, as the clusterhead is a focal node within the cluster and other cluster members dispatch data to clusterhead, the energy consumption of clusterhead rises. Therefore, if there is no attention to the remaining energy of nodes

while electing the clusterheads, this may cause dyeing out one of cluster nodes and also consuming more energy for new clusterhead re-election.

Connectivity degree:

The numbers of node neighbors, which are in transmission range of node are defined as connectivity degree of the node. This is important because if a node with a higher degree is elected to be clusterhead, it can decrease the number of clusters as well reduce the energy consumption of whole network.

Distance of node:

As the distance between two nodes within the transmission range grows, they need more energy to send/receive data. Therefore, in clusterhead election, considering the distance of candidate node with other members in range is vital.

Mobility degree:

If clusterhead moves fast, the connection among nodes will break down more quickly and nodes need to elect a new clusterhead, which result in re-affiliation of network. Then the mobility degree of nodes will be a basic factor in electing clusterheads.

4. Proposed algorithm

The proposed algorithm considers four basic factors discussed in previous part described as fuzzification. As Table 1 shows, each factor is being presented by three linguistic values. Figs. 3-6 respectively show the membership functions of remained energy, neighbors' number and distance and mobility factors. The figures show the membership functions for 30 nodes available in a 100×100 m² operational environment with initial energy of 5J and maximum speed of 5 m/s. As Table 1 shows, linguistic values low, medium and high are used for input parameters and also the values of very bad, bad, good and very good are used for output parameters that indicated by Table 2.

Table 1

The linguistic values for input parameters

Linguistic value	Input parameter
Low, Medium, High	ReE(Remaining Energy)
Low, Medium, High	Di(Distance)
Low, Medium, High	NabN(Neighbors Number)
Low, Medium, High	Mob(Mobility)

Table 2

The linguistic value for output parameters

Linguistic value	Output parameter
Very Bad, Bad, Good, Very Good	Out(Output)

Fig. 7 depicts the membership function of output unit before the defuzzification of results. The input parameters taken by a membership function with degree one and becomes a fuzzy value. For example, for a node with 2.5J remaining energy, the membership degree of the Low, Medium and High linguistic values respectively are 1, 0 and 0. So for this case Medium linguistic value is selected. Now to calculate the output result of first rule that is indicated in Table 2, the AND operator is applied to the membership values of input parameters, and the result is compared with the output membership function and the maximum value will be selected.

	Rules Database				
	ReE	Di	NabN	Mev	Oat
1	High	Low	High	Low	Very Good
2	Low	High	Low	High	Very Bad
3	High	Medium	High	Low	Good
4	Low	High	High	Low	Bad

Fig. 2. The sample rules in rules based

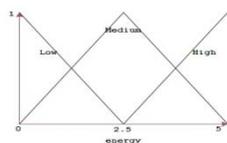


Fig. 3. Membership function of Remaining energy

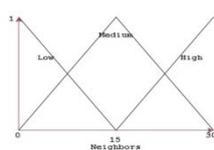


Fig. 4. Membership function of Neighbors' number

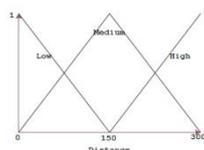


Fig. 5. Membership function of Distance

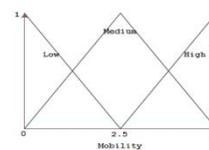


Fig. 6. Membership function of Mobility

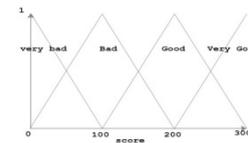


Fig. 7. Membership function for output

We assume 4 possible states for each node shown in Table 3. The nodes at first are initialized with undecided state and then after performing the algorithm they will be in one of the other three states.

Table 3

The statuses of a node

state	Code	Description
Undecided	0	The initial status simulation
Cluster head	1	Having the best score
Member	2	The other nodes
Getaway	3	Two or more CHs in its neighbor

Nodes are randomly placed and move in the environment. There are many algorithms for routing in MANET such as AODV, DSR, etc. We used routing protocol *protoname* for routing in this proposed network. First, all nodes broadcast the message "I'm free" within the network. Table 4 shows the structure of a message.

Table 4

The structure of message

sender node ID	Message Body	Sequence number	Score	Sending time
----------------	--------------	-----------------	-------	--------------

Each node with broadcasting message coincidentally set a timer and waits to receive a message from probable neighbors. After time is running up, the node checks the received packet from other neighboring nodes. If there is no other node around, this means that the node did not receive any message and then changes its state to clusterhead. When nodes receive a message from neighbors, the state of sender will be checked to see whether it is a clusterhead or not. (neighboring nodes after receiving a message from an undecided node according to its current state send the message "I'm a clusterhead" or "I'm a member" or "I'm a getaway" to that node) if there is a clusterhead around node changes its state to be a member and join to the current cluster. Accepting this situation by nodes results in reduction of energy consumption that achieved by decrease of communicative overheads due to absence of re-elections. If there are two or more clusterheads around the node, it changes its state to be a getaway and attach to a cluster with higher score. If there is no clusterhead around, the election will be held amongst neighboring nodes that are not in getaway state, then the higher-score node will be elected as clusterhead and others turn to cluster member state. Here, we avoided from engaging the getaway state in election to maintain connectivity of network that is an important parameter. In every step after defining the state of nodes, each node broadcasts its state within the network. During the simulation, re-election might be occurred as follow:

- If a node maintains high mobility and stays out of the range of current cluster change, its state is set to undecided and broadcasts with "I'm free" message.
- If the current clusterhead dies out, re-election will be held again.
- If number of a node neighbors turn to zero, the node will be a clusterhead.
- If clusterhead stays out of transmission range of its cluster members.

During the election, all 4 parameters are sent to related fuzzy functions, and then non-fuzzy output according to score of nodes are calculated and returned.

Calculating distance:

We use Eq. (2) to calculate the distance of node X_1 from X_2 . Hence, overall distance of each node to its neighbors is calculated using Eq. (3). Note that the operational environment is two-dimensional and each node is represented by a point like (X, Y) where X declares as length and Y declares as width. In addition, Dis is declared to be the overall distance of each node to neighboring nodes while n is number of these neighboring nodes.

$$d_1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

$$Dis = \sum_{i=1}^n d_i \quad (3)$$

Calculating mobility:

Eq. (2) is used for calculating mobility degree of a node. In addition, number of neighbors and remaining energy of node could be calculated and accessed from node itself. Fig. 8 shows flowchart of proposed algorithm.

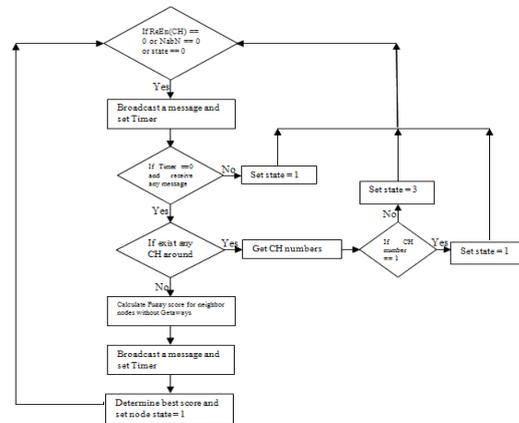


Fig. 8. Algorithm flowchart

5. Simulation

We use of NS2 (Network Simulation 2) software for simulating the proposed algorithm. This software is one of the most powerful, important and most used network simulation tool, its importance is because of being free and simple (Reineck, 2008).

5.1 The network model

The model of network is as follow:

- Operational environment $100*100 \text{ m}^2$
- The number of nodes 30, 40, 50 nodes in each phase of simulation
- The speed of nodes between 0 to 5 m/s
- The transmission range between 5 to 70 m
- The maximum amount of energy 5J

We have used 802.11b (IEEE Computer Society 2008) standard for routing in our network.

5.2 The average of clusterheads

Controlling the number of clusterheads or similarly the clusters is an important factor that reveals the efficiency of any related algorithm. As number of clusters reduces therefore intra clustering

communications reduce as well. Fig. 9 shows the average number of clusterheads that calculated for 30, 40 and 50 nodes.

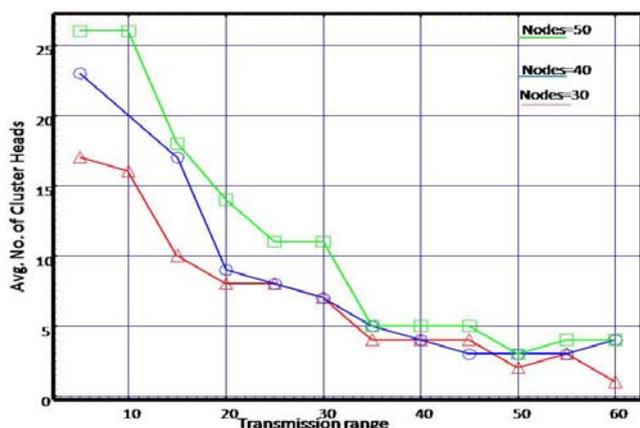


Fig. 9. Average number of cluster heads

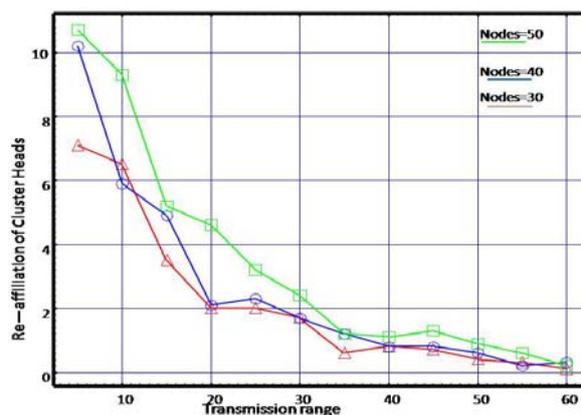


Fig. 10. Number of re-affiliation in difference transition range

5.3 Number of re-affiliation

When a node has a high level of mobility, it alternatively disconnects from one cluster and connects to another. As the number of re-affiliations of nodes stays less, the consumption of energy remains less in network. Fig. 10 shows the number of re-affiliation of nodes in the proposed network.

5.4 The number of re-election

Fig. 11 shows the number of re-election in our algorithm. The fewer number of re-election, the more stable the network is.

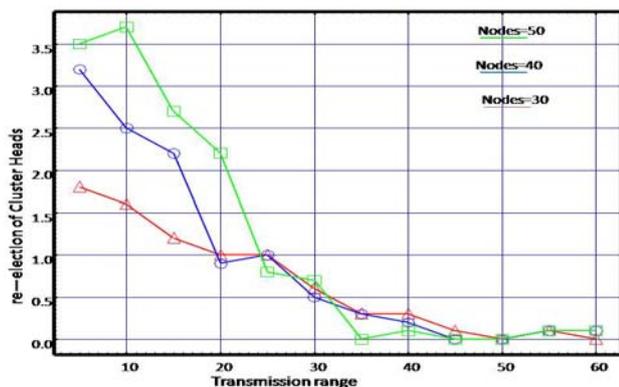


Fig. 11. Number of re-election in difference transmission range

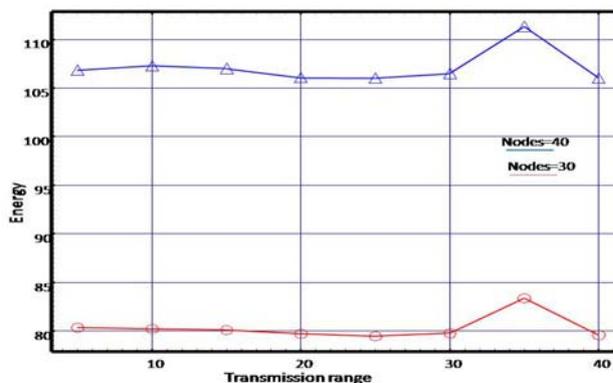


Fig. 12. Remaining energy

5.5 The amount of total energy consumption

Fig. 12 shows the amount of remaining energy in network at the end of simulation. At the beginning, the number of nodes was 30 the energy was equal to 150J and this for the 40 nodes was equal to 200J but at the end of simulation the remaining energy were in the range of 79.5 and 80.34 for 30 nodes and 105.98 and 111.27 for the 40 nodes.

6 Comparison

In this section, the proposed algorithm is applied on a network with 30 nodes and the result is compared to some important algorithms that are shown by Figs. 13-15.

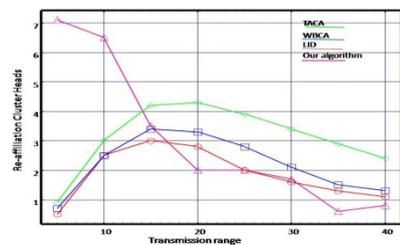


Fig. 13. Comparing re-affiliation

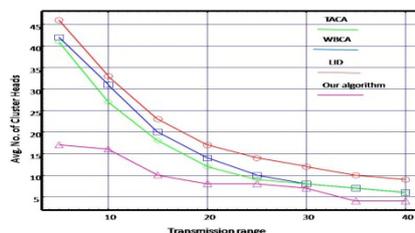


Fig. 14. Comparing average number of Cluster heads

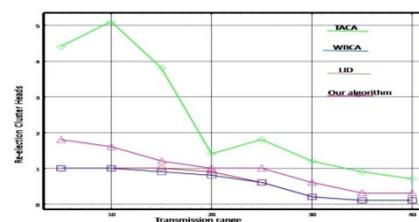


Fig. 15. Comparing re-election

7 Conclusion

In this paper, a new algorithm is proposed for nodes clustering of mobile ad hoc networks. The grade and score of each node have been calculated based on fuzzy logic and by taking distance, mobility, remaining energy and connectivity degree parameters of nodes in account. Then the algorithm compared these scores and the node with highest score was selected to be a clusterhead. In this method the re-election, re-affiliation, average number of clusterheads and remaining energy of nodes precisely considered and improved. Finally, the proposed algorithm has been compared with other existing algorithms.

References

- Adabi, S., Jabbehdari, S., & Rezaee, A. (2008). Distributed Fuzzy Score-based Clustering Algorithm for Mobile Ad-hoc Networks. *IEEE Asia-Pacific Services Computing Conference*.
- Baker, D. J., & Ephremides, A. (1981). The architectural organization of a mobile radio network via a distributed algorithm. *IEEE Transactions on Communications*, 29(11), 1694–1701.
- Basagni, S., Chlamtac, I. & Farago, A. (1997). A generalized clustering algorithm for peer-to-peer networks. In Proceedings of Workshop on Algorithm icaspect of communication (satellite workshop of ICALP)
- Basagni, S. (1999). Distributed clustering for ad hoc networks. In *Proceedings of International symposium on parallel architectures, algorithms and networks*, 310–315.
- Boleng, J., Camp, T., & Davies, V. (2002). A survey of mobility models for ad hoc network research. Wireless Communications and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research. *Trends and Applications*, 2(5), 483–502.
- Chatterjee, M., Das, S.K., & Turgut, D. (2002). WCA: A weighted clustering algorithm for mobile Ad Hoc networks. *Cluster Computing*, 5, 193–204
- Dastgheib, S. J., Oulia, H. & Sadeqi, G. M. R. (2011). An efficient approach for clustering in wireless sensor network using fuzzy logic. *International Conference on Computer Science and Network Technology*.
- Ghazanfari, M., & Rezaie, M. (2006). *An introduction to fuzzy sets theory*. Tehran: Iran University of Science and Technology press.
- Hoebeke, J., Moerman, J., Dhoedt, B. & Demeester, P. (2004). An Overview of Mobile Ad Hoc Networks: Applications and Challenges. *Journal of the Communications Network*, 3, 60-66
- Hogie, L. (2007). Mobile Ad Hoc Networks: Modeling, Simulation and Broadcast-based Applications. PhD thesis.
- IEEE Computer Society (2007). IEEE standard for Information technology telecommunications and information exchange between systems local and metropolitan area networks Specific requirements.
- Parekh, A.K. (1994). Selecting routers in ad hoc wireless network. In Proceedings of the SBT/IEEE International Telecommunication Symposium ITS.
- Reineck, K.M. (2008). *Evaluation and Comparison of Network Simulation Tools*. Master Thesis.
- Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning – I. *Information Sciences*, 8, 199–249.
- Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control*, 8, 338–353.