An improved adaptive weighted clustering algorithm based on time interval grade in Mobile Ad Hoc networks

Huyin Zhang[1], Jing Wang[1, 2]*, Fang Xu[1], Ning Xu[1], Zhiyong Wang[1], Xuejun Zhou[2], Haitao Lin[2], Peng Yu[2], Yuanyuan Zhou[2]

[1]*Computer Institute of Wuhan University, Wuhan, Hubei province, China*

[2]*Department of Information and Network Technology, Electronic Engineering School, Naval University of Engineering, Wuhan, Hubei, China*

**Corresponding author's-e-mail: wang_jing28@126.com*

Received December 2014, www.cmnt.lv

Abstract

Mobile Ad Hoc Networks (MANETs) are self-configuring dynamic networks of mobile devices connected by wireless links without any fixed infrastructure or centralized administration. In order to achieve stable clusters, the cluster-heads (CHs) maintaining the cluster should be stable with minimum re-affiliation times and number of changes on CHs, with maximal throughput of the clustering formation and maintenance. An improved adaptive weighted clustering algorithm based on time interval grade (IATIGWCA) in MANETs is proposed. Each node can be assigned an adaptive role and set its status value through their Hello messages in the formation procedure of clusters, and an appropriate CH of a cluster is elected by the calculation the total weight which comprising four factors: degree difference, average Euclidean distance, average relative speed and consumed battery power. In the maintenance procedure of clusters, the duration of clustering maintenance is set to 2 grades which are Little Time Slot and Big Time Slot in order to improve the efficiency of clustering and decrease the times of computation of the total weight of every node. The simulation results show that the selection of numbers of CHs and numbers of clusters in the stage of the formation of clusters is an optimal solution which brings higher throughput, less re-affiliation times, less number of changes on CHs and longer residence time of cluster in IATIGWCA than LID and WCA.

Keywords: clustering, cluster-head (CH), time interval grade, clustering maintenance

1 Introduction

Mobile Ad Hoc Networks are called MANETs, comprising a great deal of mobile devices that form the wireless networks without any fixed infrastructure or centralized administration. In fact, MANETs are dynamic because of the mobility of device nodes. Every node interconnects with each other and performs as a router or a package forwarder.

With the continuous increase of the size of MANETs, flat routing schemes do not scale well in terms of performance. The routing tables and topology information become more and more tremendous. In order to enhance the low bandwidth utilization and reduce the high overhead of using routing schemes in large networks, some kind of organization structure of nodes is required, and thus grouping a number of nodes into an easily manageable set which is called "Cluster". Clustering (grouping a number of nodes into a cluster) is the most popular method to impose a hierarchical structure in MANETs. In clustering algorithm, CHs are responsible for the designation of the members of the clusters and maintenance of the topology of the network, and then selecting a suitable node in a cluster as a Cluster-Head (CH) is so important. Herein, CHs act as local coordinators and handle various network functions. The clusters are able to store minimum topology information; each CH acts as a temporary base station within its cluster and communicates with other CHs. A clustering scheme should be adaptive to changes with minimum clustering management overhead, which is incurred by changes in the network topology. Due to the mobility of device nodes, their affiliation and disaffiliation from clusters perturb the stability of the network and the reconfiguration of CHs is unavoidable. It is a serious issue that the frequent changes of CHs adversely affect the performance of other mechanisms or protocols such as scheduling, routing and resource allocation [1].

An efficient clustering algorithm must adapt itself to frequently and unpredictable topology changes in MANETs. It must generate stable clusters as much as possible to reduce their update times, which can result in updates of other information such as routing, security and management information. Otherwise, recomputation of CHs and frequent information exchange among the participating nodes will bring high computation overhead. Therefore, a proper maintenance scheme of the process of clustering should be designed, and the overhead of electing CHs and maintenance should be least, and high frequency of reaffiliation (the process of joining a new cluster) could be decreased as much as possible.

In this paper, we propose an improved adaptive weighted clustering algorithm based on time interval grade (IATIGWCA) in MANETs to maintain stable clusters by electing a node with strong battery power as a CH to decrease the re-affiliation as much as possible, minimizing the number of clusters, and minimizing the number of changes on CHs and maximizing the throughput for the clustering formation and maintenance.

The rest of this paper is organized as follows. In Section II, we review previous related several clustering algorithms proposed. Section III presents the proposed algorithm for MANETs. The conditions of experimental setup and simulation results of the proposed algorithm are given in Section IV. Finally, we give conclusions of this paper in Section V.

2 Related work

In order to enhance network manageability, channel efficiency and energy economy, a several classic clustering algorithms in wireless ad hoc networks have been investigated in the past. Probably the most crucial point when dealing with clustering is the criterion how to choose the CHs. The number of CHs and average number of changes on CHs strongly influences the communication overhead and latency. And some classic clustering mechanisms have been proposed, namely, Lowest-ID [2], Highest-Connectivity [3], Distributed Mobility-Adaptive Clustering Algorithm [4], Distributed Dynamic Clustering Algorithm [5], Node-Weight Clustering Algorithm [4], and Weighted Clustering Algorithm (WCA) [1]. And they are as described below.

A. Lowest-ID Clustering Algorithm (LID):

Each node is assigned a distinct ID and broadcasts the list of IDs of nodes that it can hear including its own ID periodically. A node, which only hears nodes having higher IDs than its own ID, becomes a CH. And the lowest-ID node that a node hears is its CH, unless the lowest-ID specifically gives up its role as a CH. If a node can hear two or more CHs, it is a Gateway. Then, the other nodes are ordinary nodes.

B. Highest-Connectivity Clustering Algorithm:

Each node broadcasts the list of IDs of nodes that it can hear including its own ID. If a node is elected as a CH if it is the most highly connected node of all its uncovered neighbor nodes. In case of a tie, the lowest-ID prevails. And if a node which has not elected its CH yet, is an uncovered node, otherwise it is a covered node. When a node has already elected another node as its CH, it gives up its role as a CH.

Due to resource limitations, a CH may not be able to handle a large number of nodes, and resulting in more number of changes on CHs can lead to instability of the cluster.

C. Distributed Mobility-Adaptive Clustering Algorithm (DMAC):

This algorithm determines the CHs not based on the node ID, but based on the nodes generic weights W, which is a positive real number. Ideally, the W captures the mobility and reliability of a node and characterizes the preferences on which a node is suitable as CH. An example for a mobility based metric for clustering is given in [6].

Each node determines its role as either an ordinary node or a CH on its own. Only local information is required. That is, the role of a node is determined by using its node ID and its node weight, as well as the weights of all its neighbors, and, for cases where ties occur, their node IDs. In order to allow for fast communication between two nodes, the DMAC algorithm requires every node to be connected to at least one CH. Moreover, no two CHs can be one-hop neighbors. This is used to ensure that the CHs are well spread out in the network. A node that is added to the network starts an initialization algorithm that determines its role in the network. The decision is based on its own weight and its neighbors' weight. If the new node has a neighboring CH with a higher node weight, it decides to be an ordinary node, joins the cluster corresponding to that CH and sends a Join-message. Otherwise, it decides to become a CH itself and sends out a message of CH. This helps the neighbor nodes in getting the information about the existence and role of a new node.

D. Distributed Dynamic Clustering Algorithm (DDCA):

The (α, t) criterion of DDCA describes a probabilistic bound α on the availability of paths in the corresponding cluster over a certain time t. A new node seeking a cluster to join sends out a JoinRequest message. If it does not receive any responses, it basically builds a new cluster and becomes a CH. If the new node upon sending a JoinRequest receives one or more JoinResponse messages before its JoinTimer runs out, it decides to join the cluster with the highest (α, t) value. This value has to exceed the minimum required value (α, t) threshold, otherwise the JoinResponse is ignored. And the cluster strength, seen by a node m and given by the (α, t) value, gives the measure of the availability of a path from the node m over some initial hop node n to the CH of the corresponding cluster.

An important property of DDCA is that new nodes tend to join existing cluster. New nodes don't first of all seek to become CHs, hence the clustering algorithm is very stable. However, it is disadvantageous in this algorithm that the suitability of nodes to be a CH is not considered in the CH election process.

E. Node-Weighted Clustering Algorithm (NWC):

In this approach, each node is assigned weights based on its suitability of being a CH. A node is chosen to be a cluster head if its weight is higher than any of neighbor's weight; otherwise, it joins a neighboring cluster head. The smaller ID node id is chosen in case of a tie. Making an assumption is that the network topology does not change during the execution of the algorithm. To verify the performance of the system, the nodes were assigned weights which varied linearly with their speeds but with negative slope. Since node weights were varied in each simulation cycle, computing the cluster heads becomes very expensive.

F. Weighted Clustering Algorithm (WCA):

The Weighted Clustering Algorithm (WCA) was originally proposed by M. Chatterjee et al [1]. In order to decide how well suited a node is for being a CH and maintenance of cluster is more reasonable, four factors are taken into account. And the four factors are node degree (V1), distance summation to all its neighboring nodes (V2), mobility (V3) and remaining battery power (V4) respectively.

So, the clustering problem is converted into an optimization problem since an objective function is formed. The weight of a node is calculated as

$$Wnode = w1 \times V1 + w2 \times V2 + w3 \times V3 + w4 \times V4$$

Keeping in view the above factors, the combined weight for each node is calculated. The node with the smallest weight is then chosen as the CH. One of the drawbacks of this algorithm is that it uses the concept of global minima. All the nodes in the network have to know the weights of all the other nodes before starting the clustering process. This process can take a lot of time. Also, two CHs can be one-hop neighbors, which results in the clusters not necessarily being spread out in the network, and the overhead induced by WCA is very high.

3 Proposed algorithm: an improved adaptive time interval grade-based weighted clustering algorithm (IATIGWCA)

A. Objectives.

In order to design and implementing the IAIGWCA algorithm, the core objectives are underlying points:

1. The clusters of MANET should be efficient and stable, and some hello messages between CHs and cluster members are needed to maintain the routing structure and avoid congestion in the network.

2. The selection of numbers of CHs and numbers of clusters in the stage of the formation of clusters is an optimal solution which brings higher throughput versus some classic algorithms (LID and WCA) [7].

3. In the stage of the maintenance of the clusters, the CHs have to be re-elected in the specified time interval. So the re-affiliation times of the clusters should be less, and the residence time of cluster should be longer.

B. The formation procedure of clusters.

We consider that MANET is represented by an undirected graph $G=(V,E)$ where V is the set of nodes and E is the set of bi-directional communication links. In the formation phase of clusters, V is divided into a collection of subsets $\{V_1,V_2,...,V_n\}$, where $V=\bigcup_{i=1}^{n}V_i$. And every subset V_i induces a connected sub graph of G, and is called cluster set which can overlap.

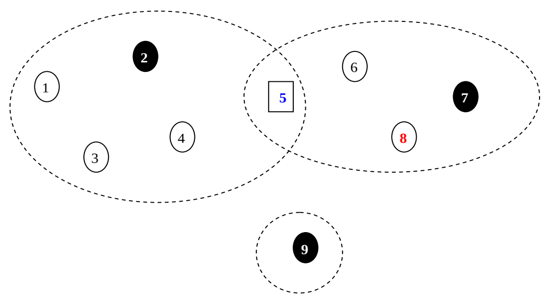

FIGURE 1 Roles assignment 3 of nodes in different clusters

Figure 1 shows the distribution of the nodes of clusters, and the difference of roles of nodes is presented. The roles of nodes in clusters are classified four types: cluster-head (CH), gateway, ordinary member and original role. For example, node 2, 7 and 9 are CHs which are responsible for the designation of the members of the clusters and maintenance of the topology of the network. Node 5 is a gateway node which is responsible for data relaying between clusters; node 8 is a new original node which has not joined in any cluster, not liking the ordinary member nodes which are including node 1, 3, 4 and 6.

Therefore, all nodes of clusters have their roles, and the role assignment algorithm of member nodes in clustering setup is given as follows.

Algorithm 1. Role assignment of member nodes in clustering setup.

Every node can receive the Hello messages from its all 1-hop neighbors.

1: **IF** node *u* which is not a cluster-head receive a MAC_{CH} **THEN**

2: Assign the status of node *u* to an ordinary member role, and its status value is 1.

3: Update the neighborhood lists of node *u*.

4: **IF** node *u* receive another MAC_{CH} **THEN**

5: Assign the status of node *u* to a gateway role, which node is responsible for data relaying between clusters, and its status value is 2.

6: Update the neighborhood lists of node *u*.

7: **END IF**

8: **ELSE IF** node *u* is a cluster-head **THEN**

9: Assign the status of node *u* to a cluster-head role, and its status value is 0.

10: Update the neighborhood lists of node *u*.

11: **ELSE IF** node *u* which is not a cluster-head does not receive a MAC_{CH} **THEN**

12: Assign the status of node *u* to an original role which has not joined in any cluster, and its status value is 3.

13: Update the neighborhood lists of node *u*.

14: **END IF**

In the algorithm 1, every node can receive the Hello messages from its neighbors. According to their Hello messages, the node can be assigned an adaptive role and set its status value in the procedure of clustering setup. MAC_{CH} is the physical address of cluster head in an ordinary cluster.

However, the election method of CH is most important part in the procedure of clustering setup. An improved adaptive time interval grade-based weighted clustering algorithm (IATIGWCA) is proposed.

Algorithm 2. Calculation the total weight of node *u* in clustering setup.

1: Compute the degree difference Δ_u of the node *u*, $\Delta_u \leftarrow |d_u - N|$, where d_u is the degree of connectivity of the node *u* by counting its neighbors, and N is a predefined threshold, i.e. the ideal number of nodes that a cluster-head can manage ideally.

2: Compute the average Euclidean distance $\overline{D_u}$ between the node *u* and its all neighbors, and a predefined moving duration of node *u* is partitioned several intervals (n Δt), $\overline{D_u} \leftarrow$

$$\frac{\sum\limits_{v\in Neighbour(u,n\Delta t)} \frac{\sum\limits_{i=1}^{n}\sqrt{(X_u(t_i)-X_v(t_i))^2+(Y_u(t_i)-Y_v(t_i))^2}}{n}}{Count(Neighbour(u,n\Delta t))}$$

, where ($X_u(t_i)$, $Y_u(t_i)$) is the position of the node *u* at the *i*-th interval

3: Compute the average relative speed $\overline{V_u}$, utilizing the same method of $\overline{D_u}$, $\overline{V_u} \leftarrow \frac{\sum\limits_{v\in Neighbour(u,n\Delta t)} \frac{\sum\limits_{j=1}^{n}|V_u(t_j)-V_v(t_j)|}{n}}{Count(Neighbour(u,n\Delta t))}$

4: Compute the consumed battery power E_u of the node *u*

5: Calculate the total weight of the node *u*, $W_u = w_1 \times \Delta_u + w_2 \times \overline{D_u} + w_3 \times \overline{V_u} + w_4 \times E_u$, where w_1, w_2, w_3, w_4 are the weighting factors and $w_1 + w_2 + w_3 + w_4 = 1$

6: Elect the node u which its total weight value is the minimum of the all cluster nodes as a cluster-head (dominant set)

In the algorithm 2, an adaptive CH of a cluster is elected through the calculation the total weight in clustering setup. Like WCA algorithm, four factors are taken into account, and the four factors are degree difference Δ_u, average Euclidean distance D_u, average relative speed V_u and consumed battery power E_u respectively. The CH has the least total weight value.

C. The maintenance procedure of clusters.

The election of CH during the formation stage of clusters may not remain the best choice for long in an emergency MANET. Due to the mobility and consumption of battery power, we have to monitor the performance of the CHs and every member node by periodically updating the relative location information *Position* and the relative speed information *Speed* as follows.

Step 1: In the maintenance procedure of clusters each node periodically broadcasts the Hello message to its neighbors, which is shown below [8].

MAC_u	MAC_{CH}	Channel	Status	Position	Speed	Priority

FIGURE 2 Format of Hello message

MAC_u : Physical address of node *u*;

MAC_{CH} : Physical address of cluster head in an ordinary cluster;

Channel: Through utilizing the different channels, data can be transmitted from node *u* to its neighbor nodes;

Status: Status information of node *u*, where value 0 means that node *u* is a cluster-head; value 1 indicates that *u* is an ordinary node in a cluster; value 2 represents that *u* is a gateway node which is affiliated with several clusters; value 3 means that node *u* has not joined any cluster and it is regarded as a new node;

Position: Relative location information of node *u*, including its x-coordinate and y-coordinate;

Speed: Relative speed information of node *u*, including its direction and value for system of coordinates;

Priority: Priority of node *u* in grade of Little Time Slot, through comparing the priority with its neighbor's, the node which has a least priority can become a new cluster-head.

Step 2: Each node builds up a neighborhood main list and a sub-list with the aid of the Hello messages sent by its neighbors, and the neighborhood lists are periodically updated according to the Hello messages received, and its explicit format are shown in Table 1 and Table 2.

TABLE 1 Neighborhood main list of node *u*

$ID_{neighbour}$	$MAC_{neighbour}$	Channel	Status	Priority
a	xx-xx-xx-xx-xx-xx	0	0	0.65
b	xx-xx-xx-xx-xx-xx	3	1	0.48
c	xx-xx-xx-xx-xx-xx	4	1	0.34

TABLE 2 Neighborhood sub-list of node *u*

$ID_{neighbour}$	i	D_i	V_i
a	0	$D_0 = \sqrt{(X_a(t_0) - X_u(t_0))^2 + (Y_a(t_0) - Y_u(t_0))^2}$	$V_0 = \lvert V_a(t_0) - V_u(t_0) \rvert$
a	1	D_1	V_1
a	2	D_2	V_2
a	…	...	...
b	0	$D_0 = \sqrt{(X_b(t_0) - X_u(t_0))^2 + (Y_b(t_0) - Y_u(t_0))^2}$	$V_0 = \lvert V_b(t_0) - V_u(t_0) \rvert$
b	1	D_1	V_1
b	2	D_2	V_2
b	…	...	...

In Table 1, the parameters, such as $ID_{neighbour}$, $MAC_{neighbour}$, Channel, Status and Priority are the same meaning as the Hello message's. And in Table 2, during every Hello message cycle i, the $Position_i$ and $Speed_i$ value are saved and calculated in D_i and V_i, therein the value of i is from 0 to N-1.

In order to improve the efficiency of clustering and decrease the times of computation of the total weight of every node, the duration of clustering maintenance is set to 2 grades which are Little Time Slot and Big Time Slot.

Phase 1: in Little Time Slot, priority of node *u* is calculated periodically. And then node *u* forwards it to its neighbor nodes. Through comparing the Priority value of node *u* and its neighborhood, the node *x* which has a least priority can be elected as a new cluster-head (dominant set), and its status value is set to 0, updating the neighborhood lists of the node *x*. i.e. algorithm 3 is executed.

For example, the priority of node *u* is 0.37, and the priorities of a, b and c are 0.65, 0.48 and 0.34 separately. Through comparison, node c has the least priority value, and it is elected as a new cluster-head.

The computation and comparison algorithm of priority of nodes in grade of Little Time Slot is proposed in algorithm 3.

Algorithm 3. Computation and comparison of priority of nodes in grade of Little Time Slot.

1: In order to improve the efficiency of clustering and decrease the times of computation of the total weight of every node, the duration of clustering maintenance is set to 2 grades which are Little Time Slot and Big Time Slot, and elect which one become a new cluster-head through comparing the priority of nodes in grade of Little Time Slot.

2: Define the priority of node *u* is PRI_u, $PRI_u \leftarrow \alpha \times E_{remainder} + \beta \times (-H(u))$, $H(u) \leftarrow -\frac{\sum_{k \in Neighbour(u)} P_k \times \log(P_k)}{\log C(Neighbour(u))}$, $P_k \leftarrow \frac{\sqrt{(X_k - X_u)^2 + (Y_k - Y_u)^2}}{\sum_{i \in Neighbour(u)} \sqrt{(X_i - X_u)^2 + (Y_i - Y_u)^2}}$, where α, β are the priority factors and $\alpha + \beta = 1$, and $E_{remainder}$ is the remaining power of node *u*, and $H(u)$ is the entropy of node *u*, and $C(Neighbour(u))$ is the cardinality (degree) of set $Neighbour(u)$.

3: IF a new arrival node *u* (status value is 3) is moving into a cluster THEN

4: Compute the priority of node *u* PRI_u, and compare with PRI_{CH}.

5: IF $PRI_u > PRI_{CH}$ THEN

6: Elect node *u* as a new cluster-head (dominant set), and its status value is 0.

7: Update the neighborhood lists of node *u*.

8: Assign the status of old cluster-head (CH_{old}) to an ordinary member role, and its status value is 1.

9: Update the neighborhood lists of old cluster-head (CH_{old}).

10: ELSE IF $PRI_u <= PRI_{CH}$ THEN

11: Node *u* join in the cluster, and the status of node *u* is assigned to an ordinary member role, and its status value is 1.

12: Update the neighborhood lists of node *u*.

13: The cluster-head (CH) continues its role.

14: END IF

15: ELSE IF the cluster has not new arrival node, or lose some member node(status value is 1 or 2) THEN

16: For each member node N_i in the $ClusterSet$ do

17: IF $PRI_{N_i} > PRI_{CH}$ THEN

18: Elect node N_i as a cluster-head of the cluster, and its status value is 0.

19: Update the neighborhood lists of node N_i.

20: Assign the status of old cluster-head (CH_{old}) to an ordinary member role, and its status value is 1.

21: Update the neighborhood lists of old cluster-head (CH_{old}).

22: END IF

23: END FOR

24: END IF

Phase 2: in Big Time Slot, total weight of node *u* is calculated periodically. Big Time Slot period is partitioned N times Little Time Slot period, and a Hello message can be broadcasted from a node in Little Time Slot period, and N depends on the specific network topology. During every Little Time Slot period i, the euclidean distance D_i between the node *u* and its all neighbors is computed,

$$D_i = \sqrt{(X_{neighbours}(t_i) - X_u(t_i))^2 + (Y_{neighbours}(t_i) - Y_u(t_i))^2},$$

and relative speed V_i between the node *u* and its all neighbors is computed, $V_i = |V_{neighbours}(t_i) - V_u(t_i)|$. Through executing algorithm 2, the node y which its total weight value is the minimum of the all cluster nodes is elected as a new cluster-head (dominant set), and its status value is set to 0, updating the neighborhood lists of the node y.

The re-election algorithm of cluster-head in grade of Big Time Slot is proposed in algorithm 4.

Algorithm 4. Re-election procedure of cluster-head in grade of Big Time Slot.

1: In grade of Big Time Slot, according to Algorithm 1, the total weight of every node is compared with its cluster-head (CH)'s for a predefined long duration.

2: **For** each node N_j in the $ClusterSet$ **do**

3: **IF** $W_{N_j} < W_{CH}$ **THEN**

4: Re-elect N_j as a new cluster-head (dominant set), and its status value is 0.

5: Update the neighborhood lists of node N_j.

6: Assign the status of old cluster-head (CH_{old}) to an ordinary member role, and its status value is 1.

7: Update the neighborhood lists of old cluster-head (CH_{old}).

8: **ELSE IF** $W_{N_j} >= W_{CH}$ **THEN**

9: Don't perform the re-election and the cluster-head (CH) continues its role.

10: **END IF**

11: **END FOR**

4 Experimental setup and results

A. Simulation environment and parameters.

In order to evaluate the performances of the proposed clustering algorithm (IAGWCA), we implemented the proposed algorithm in NS2 [9]. We performed different simulation scenarios, and through the simulation to evaluate the performance metric of different classic algorithm, such as average number of clusters, average number of changes on CHs, re-affiliation times, average number of clusters, throughput and average cluster residence time. We use a commonly used mobility model such as the Random Way Point model [10] that represents a particular case of the Random Trip model [11]. In the Random Way Point model, each node chooses its direction and its speed according to a fixed time interval [12]. The move of the node is done during a time Δt with some pauses. However, the possibility of no pause represents the Random Walk model. Other basic parameters used in our experiments are summarized in the Table 3.

TABLE 3 Simulation parameters

Parameter	**Value**
Number of nodes	20-120
Size of network	500m×500m
Speed of the nodes	2-15Km/h
Transmission range	30-300m
Pause Time	1 sec
Little Time Slot (Hello Interval)	4 sec
Big Time Slot	40 sec
Frequency band	5.4GHz
Size of cluster	12 nodes
Time of simulation	400 sec

According to the experimental environment, in the calculation of the algorithm 2, we can set w1=0.2; w2=0.3; w3=0.3; w4=0.2, in order to compute the appropriate total weight of each node in clustering setup.

B. Metrics and simulation results.

The following metrics serve as the ways of our performance evaluations, contrasting the Lowest-ID Clustering Algorithm (LID), Weighted Clustering Algorithm (WCA) and the proposed clustering algorithm (IAGWCA).

a. The average number of clusters:

$$\frac{\text{number of nodes}}{\text{average number of nodes in a cluster}};$$

b. The average number of changes on CHs:

$$\frac{\text{total number of changes on CHs}}{\text{number of CHs}};$$

c. The re-affiliation times: times of re-election procedure of cluster-head;

d. The throughput: data transmitting rate of all clusters;

e. The average cluster residence time:

$$\frac{\text{total stable time of all clusters}}{\text{number of clusters}}.$$

Figure 3 shows the variation of the average number of clusters with respect to the transmission range from 30m to 200m. The results are indicated for three different values of number of nodes (30 nodes, 70 nodes and 120 nodes). We observe that the average number of clusters is decreased when the transmission range increases, and the reason is that the node with a higher transmission range means it covers a larger area in wireless network.

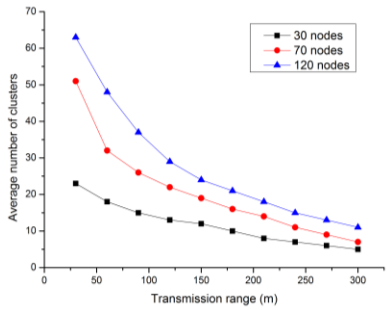

FIGURE 3 Average number of clusters versus transmission range

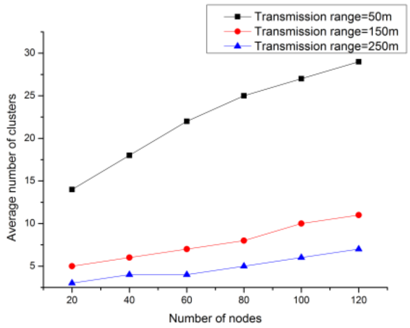

FIGURE 4 Average number of clusters versus number of nodes

Figure 4 presents the relationship between the average number of clusters and the number of nodes on the three different conditions of the transmission range (50m, 150m and 250m). As is shown in Figure 3, a bigger number of nodes can increase the average number of clusters, and the average number of clusters is less dynamic when the transmission range increases.

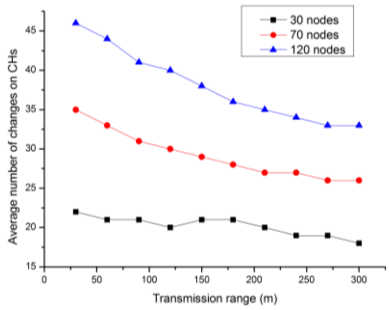

FIGURE 5 Average number of changes on CHs versus transmission range

In Figure 5, we can observe that the average number of changes on CHs decreases and the clusters, in which the nodes tend to remain in the range of their neighbors become more and more stable when the transmission range begins to be larger. And the number of nodes becomes more and more, while the influence of transmission range on average number of changes on CHs is less and less.

Figure 6 depicts that the average number of clusters increases when the maximum speed of node begins to be larger. Our proposed algorithm exhibits a lower sensitivity to the maximum speed of node than the LID and WCA. The improved resilience against node velocity fluctuations is attributed to the quantitative consideration of the average relative node-mobility measure (speed) and the neighbourhood list of every node.

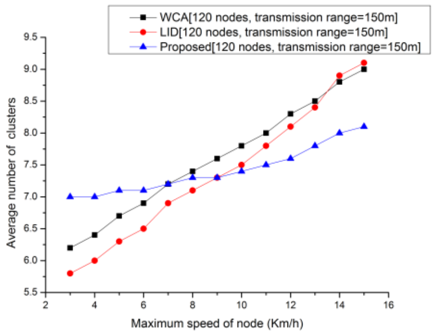

FIGURE 6 Average number of clusters versus maximum speed of node

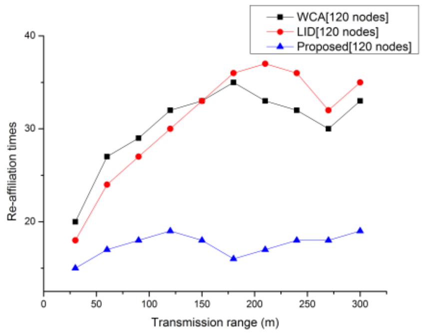

FIGURE 7 Re-affiliation times versus transmission range

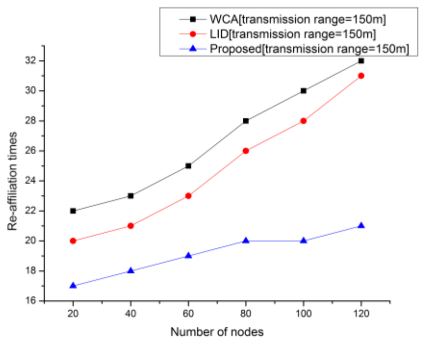

FIGURE 8 Re-affiliation times versus number of nodes

The re-affiliation times for three algorithms increases when the transmission range begins to be larger as described in Figure 7. And the times of LID and WCA are almost twice than the proposed which exhibits a lower sensitivity to the transmission range. This is due to the nodes of clusters in IAGWCA is more stable than the others. In Figure 8, we can observe the re-affiliation time is a direct proportion function of the number of nodes. The same phenomena are that the times of the proposed is lower than the others.

Figure 9 plots the throughput for three algorithms achieved in bits/second as a function of the maximum speed of node. As is shown, the proposed is found to achieve a better throughput than LID and WCA. It is clear that the maximum speed of node increases there is a reduction in the throughput of the network.

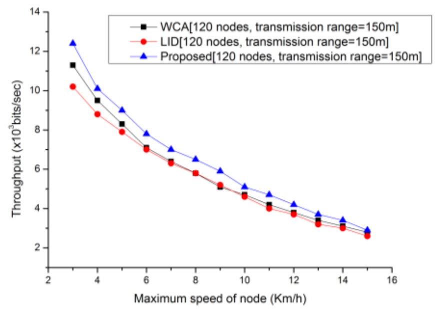

FIGURE 9 Throughput versus maximum speed of node

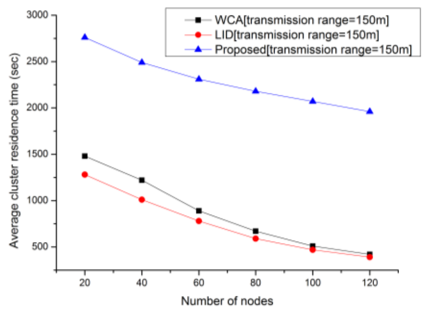

FIGURE 10 Average cluster residence time versus number of nodes

In Figure 10, the average cluster residence time for three algorithms is described versus number of nodes. It can be seen that when number of nodes increases, for three algorithms the average cluster residence time is decreased, and it confirms that the average cluster residence time of proposed is almost twice than the other two.

5 Conclusions

This paper has presented an improved adaptive weighted clustering algorithm based on time interval grade (IATIGWCA) in MANETs to maintain stable clusters. In the formation procedure of clusters, according to their Hello messages, each node can be assigned an adaptive role and set its status value. And through the calculation the total weight an appropriate CH of a cluster is elected. In the calculation the total weight, four factors are taken into account, such as degree difference, average Euclidean distance, average relative speed and consumed battery power. And in the maintenance procedure of clusters, the duration of clustering maintenance is set to 2 grades which are Little Time Slot and Big Time Slot in order to improve the efficiency of clustering and decrease the times of computation of the total weight of every node. Simulation results indicated that the selection of numbers of CHs and numbers of clusters in the stage of the formation of clusters is an optimal solution which brings higher throughput in IATIGWCA than LID and WCA. Furthermore, the re-affiliation times of the clusters and the average number of changes on CHs can be less, and the residence time of cluster should be longer in the maintenance procedure of clusters.

Acknowledgments

Wang Jing thanks for being supported by the National Natural Science Foundation of China under Grant No. 61272454 and the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20130141110022.

References

[1] Chatterjee M, Das S K, Turgut D 2002 WCA: A Weighted Clustering Algorithm for Ad hoc Networks *Clustering Computing* **5** 193-204
[2] Ephremides A, Wieselthier J E, Baker D J 1987 A Design Concept of Reliable Mobile Radio Networks with Frequency Hopping Signaling *Proceedings of the IEEE* **75**(1) 56-73
[3] Gerla M, Tsai J 1995 Multicluster, mobile, multimedia, radio network ACM-Baltzer Journal of Wireless Networks **1**(3) 255-65
[4] Basagni S 1999 Distributed Clustering for ad hoc networks In Proceedings of International Symposium on Parallel Architectures, Algorithms, and Networks (IS-PAN) Perth/Fremantle Australia 310-5
[5] McDonald A B, Znati T F 1999 A mobility-based framework for adaptive clustering in wireless ad hoc networks *IEEE Journal on Selected Areas in Communications* **17**(8) 1466-87
[6] Basu P, Khan N, Little T D C 2001 A mobility Based Metric for Clustering in Mobile Ad Hoc Networks *In Proceedings of IEEE ICDCS 2001 workshop on Wireless Networks and Mobile Computing Phoenix AZ* 412-8
[7] Rohini S, Indumathi K 2011 Consistent Cluster Maintenance Using Probability Based Adaptive Invoked Weighted Clustering Algorithm in MANETs *In Proceedings of the National Conference on Innovations in Emerging Technology-2011 Kongu Engineering College India* **17&18** 37-42
[8] Wu K, Zhong Z, Hanzo L 2010 A Cluster-head Selection and Update Algorithm for Ad Hoc Networks *In Proceedings of IEEE Globecom 2010 Miami FL* 6-10
[9] The Network Simulator-ns-2. http://www.isi.edu/nsnam/ns/
[10] Navidi W, Camp T 2004 Stationary distributions for the random waypoint mobility model *IEEE Transactions on Mobile Computing* **3**(1) 99-108
[11] Boudec J Y L, Vojnovic M 2005 Perfect simulation and stationarity of a class of mobility models In Proceedings of the IEEE Infocom: 24th Annual Joint Conference of the IEEE Computer and Communications Societies **4** 2743-54
[12] Lin G, Noubir G, Rajaraman R 2004 Mobility models for ad hoc network simulation In Proceedings of IEEE Infocom: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies **1** 1433-9

Authors

Zhang Huyin, 02.1962, Wuhan, China

Current position, grades: professor, PhD supervisor in Computer Institute of Wuhan University in China.
University studies: Computer Institute of Wuhan University, Wuhan, Hubei province, China.
Scientific interest: architecture of computer and computer network.
Publications: 15 papers, 6 books.
Experience: 18 research projects.

Wang Jing, 08.1980, Wuhan, China

Current position, grades: PhD the fourth grade.
University studies: Computer Institute of Wuhan University, Wuhan, Hubei province, China.
Scientific interest: wireless network and architecture of computer.
Publications: 5 papers, 3 books.