The CSI Journal on
Computer Science and Engineering
Vol. 7, No. 2 & 4 (b), 2009
Pages 24-33
**Regular Paper**

# A New Self-Organizing Mobile Ad Hoc Network Using Case Base Reasoning

Reza Assareh[1] Ali Assarian[1] Arash Dana[2] Ahmad Khadem Zadeh[3]

[1]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2]Department of Electrical Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran
[3]ICT Research Institute, Tehran, Iran

## Abstract

This paper proposes an intelligent weight based clustering algorithm for mobile ad hoc networks. The proposed algorithm takes into account transmission power, battery power, $T_{nb}$ (connection duration with neighbors), $D_{nb}$ (Average of distance with neighbors in $T_{nb}$), and the degree of a node for forming clusters. New Parameters of $T_{nb}$ and $D_{nb}$ are adopted as two metric to elect ClusterHead (CH) in this paper. Using the first parameter increases the stability of cluster architecture. Using the second parameter causes the ClusterHeads to consume less power for communication with their neighbors. We provide the node with the capability of learning, by using Case-Based Reasoning (CBR). In accordance with the learning capability of the nodes, the $T_{nb}$ and $D_{nb}$ parameters are intelligently calculated and an optimum clustering algorithm is presented. The main goals of this article are increase of stability of clusters and decrease consumed power during interaction between clusterhead and its neighbors. Through simulations we have compared the performance of our algorithm with that of the Lowest-ID (LID), Highest Degree (HD) and Weighted Clustering Algorithm (WCA) algorithms in terms of number of reaffiliations and power consumed in total network. Results obtained from simulations proved that the proposed algorithm achieves the goals.

**Keywords:** MANET, Case-Based Reasoning, Clustering, Stability.

## 1. Introduction

A mobile ad hoc network (MANET) is a collection of wireless mobile nodes that dynamically form a network without the need for any pre-existing network infra-structure or central control. Communication between any two nodes in a MANET might require the packets to traverse multiple hops. To provide communication throughout the network, the mobile hosts act as routers and cooperate to handle various network functions, such as traffic routing.

Routing in MANET is challenging because of the dynamic nature of the network topology. Dividing mobile nodes to groups named clusters is a well-known method to reduce the amount of routing data exchanges and consequently, to save the communication bandwidth and energy. The clustering algorithm and the selection criterion of clusterhead are crucial to a clustering ad hoc network.

Clusterheads act as local coordinators, and in addition to support packets routing and forwarding, they may resolve channel scheduling, perform power measurement/control, the maintain time division frame synchronization.

Various algorithms have been presented for the election of CHs and each of these algorithms has taken into account special parameters as metric of this election. Some of these parameters are battery power, number of neighbors, node mobility, and distance from neighbors, etc.

Clustering algorithms impose great traffic load upon the network, increase process of nodes and as a result more

battery power is consumed in the network. As a result the election of CHs must be in a way that the cluster stability can be preserved as much as possible and repetitive clustering is not necessary. To attain these goals, it should be known what parameters cause the CHs to be eliminated.

A cluster disappears due to two reasons:

- Node movement to the outside of its CH coverage area.
- Descend CH battery power of a predefined threshold.

To solve the first problem, the CH must remain in the covering area of its neighbors, for a longer time. To solve the second problem; firstly, CH must have the more remaining battery power; secondly, it must have the less average distance with the neighbors to consume less battery power for connection to them.

In this paper, two new parameters named neighborhood duration ($T_{nb}$) and average distance from neighbors in neighborhood duration ($D_{nb}$) are presented. Using the first parameters causes the nodes to remain in the coverage area of their own CH longer. Using the second parameters causes the CHs to consume less power for communication with their neighbors.

Case-based reasoning is an approach to problem solving and learning where a new problem is solved by finding a set of similar previously cases, and reusing them in the new problem situation solving process.

The amounts of $T_{nb}$ and $D_{nb}$ are computed due to providing the nodes with the learning capability and through retaining and retrieving the previous cases. Case-based reasoning provides the learning capability for each node. By using CBR method, each node learns to be neighbor with which nodes, how long this neighboring lasts and accounts its average distance with neighbor nodes.

In this paper, we propose a weighted clustering algorithm for clustering in wireless ad hoc networks. By using this algorithm, each node is selected as a cluster head or a cluster member. Result of the simulations performed on this algorithm shows that the average duration of time in which a clusterhead remains is its leadership status has been improved over other clustering algorithm. This shows that this algorithm creates more stable clusters in comparison to other algorithms.

The rest of this paper is organized as follows. In Section 2, we review on existing clustering algorithms proposed. The basic definitions of case-based reasoning are presented in section 3. Section 4 presents the proposed algorithm for mobile ad hoc networks. The analyzed performance of the proposed algorithm is given in Section 5. Finally, Section 6 concludes this paper.

## 2. Related Works

There have been several proposals for the election of clusterheads in mobile ad hoc networks, which include:

Highest-Degree (HD) algorithm [1,2], exclusively uses location information for cluster formation: the highest degree node in a neighborhood, i.e. the node with the largest number of neighbors is elected as CH. Basically, each node either becomes a clusterhead or remains as an ordinary node (cluster member). The reaffiliation count of nodes is high due to node movements.

The Lowest-Identifier algorithm [3, 4, and 5] chooses the node with the minimum identifier (ID) as a clusterhead. For this algorithm, the system performance is better compared to the Highest-Degree heuristic in terms of the throughput. 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.

The Weighted Clustering Algorithm (WCA) [6] employs combined-metrics-based clustering: a number of metrics, including node degree, CH serving time (to estimate residual energy capacity) and moving speed, are taken into account to calculate a weight factor $W_v$, for every node v. As a result, the overhead induced by WCA is very high. If a node moves into a region that is not covered by any clusterhead, then the cluster set-up procedure is invoked throughout the whole system.

The proposed WBACA [7] takes into account the transmission power, transmission rate, mobility, battery power and the degree of a node for forming clusters. Unlike the Lowest-ID algorithm, which finds only the local minima of IDS and the Weighted Clustering Algorithm (WCA), which finds the global minima of weights the proposed WBACA finds the local minima of weights for the clustering process.

In LIDAR algorithm [8], CHs are initially elected based on the time and cost-efficient lowest-ID method. During clustering maintenance phase though, node IDs are re-assigned according to nodes mobility and energy status, ensuring that nodes with low mobility and sufficient energy supply are assigned low IDs and, hence, are elected as CHs.

The DWCA [9] algorithm is an enhanced version of WCA to achieve distributed clustering set up and to extend lifetime span of the system. DWCA consists of the clustering set up and clustering maintenance phases. DWCA chooses locally optimal clusterheads and incorporates power management at the clusterheads.

In [10], willingness factor (W) is adopted as a metric to elect CH. The willingness factor W describes the sole mobility of a node and the relative mobility of a group of nodes, which are jointly represented by stable factor S and the number of relatively stable nodes M. Namely, W can be expressed as a function of S and M, i.e. W=f(S, M).

In reference [11] it is presented a novel service discovery algorithm based on node clustering. Nodes within a cluster may sleep to save energy during idle period time. A cluster head node is always active and answers discovery requests on behalf of other nodes to achieve low discovery latencies. Simulation experiments show energy savings of up to 66% compared to an approach where all nodes are permanently active while the discovery latencies were not increased.

In reference [12] it is proposed a novel graph-based mobility model, which provides a more realistic movement than the random walk model by reflecting the spatial constraints in the real world.

In three algorithms of [7], [8], [13], [14] and [15] the nodes' mobility has been used as metric and a node which is chosen as clusterhead, must have less mobility. While for having stable clusters the relative mobility of each node compared with its neighbors must be taken into account. In fact, the node which is chosen as the clusterhead must be more similar to its neighbors not only in terms of speed but also in terms of mobility direction. In algorithm [10] only

speed of nodes in comparison with each other has been taken into consideration and it has nothing to do with concerning mobility direction.

[16] Proposed a new mobility-aware technique for stable clustering in MANETs. This mobility criterion predicts the variability of the neighborhood of nodes over time. The highest degree clustering algorithm [10] is combined with this method to minimizing number of clusters in the network. The accuracy of mobility prediction method which used in [16] is weak because it doesn't consider mobility parameters such as speed and mobility direction to its estimation. Another drawback of this algorithm is ignoring battery power of mobile nodes to CHs selection.

In this paper, we predict node mobility in the future based on node mobility in the past using a learning method. Our approach is more promising due to high accuracy of estimation in term of mobility metrics.

# 3. Case Based Reasoning

Case-based reasoning is a mature subfield of artificial intelligence. As Riesbeck and Schank [17] describe case-based reasoning, "input a problem, find a relevant old solution, and adapt it."

Case-based reasoning is a recent approach to problem solving and learning that has got a lot of attention over the last few years. CBR is a problem solving paradigm that in many respects in fundamentally different from other major AI approaches. Instead of relying solely on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, CBR is able to utilize the specific knowledge of previously experienced, concrete problem situations(cases). A new problem is solved by finding a similar past case, and reusing it in the new problem situation. A second important difference is that CBR also is an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems.

Case representation and indexing are choose an appropriate representation format for a case [18], and build a case index to facilitate the retrieval of similar cases in the future.

We used CBR as a prediction method to improving clustering process in case of stability and power consumption.

## 3.1. Why Case Base Reasoning?

However, over the last few years an alternative reasoning paradigm and computational problem solving method has increasingly attracted more and more attention. Case-based reasoning (CBR) solves new problems by adapting previously successful solutions to similar problems. CBR is attracting attention because it seems to directly address the problems outlined above. Namely:

- CBR does not require an explicit domain model and so elicitation becomes a task of gathering case histories.
- Implementation is reduced to identifying significant features that describe a case, an easier task than creating an explicit model.
- By applying database techniques largely volumes of information can be managed.
- CBR systems can learn by acquiring new knowledge as cases thus making maintenance easier.

## 3.2. Case Base Reasoning Cycle

The general model of a CBR process is often referred to as the CBR-cycle (Figure1). This cycle generally comprises four activities:

- Retrieve the case(s) from the case-base whose problem is most similar to the new problem.
- Reuse the solutions from the retrieved cases to create a proposed solution for the new problem.
- Revise the proposed solution to take account of the problem differences between the new problem and the problems in the retrieved cases.
- Retain the new problem and its revised solution as a new case for the case-base if appropriate.

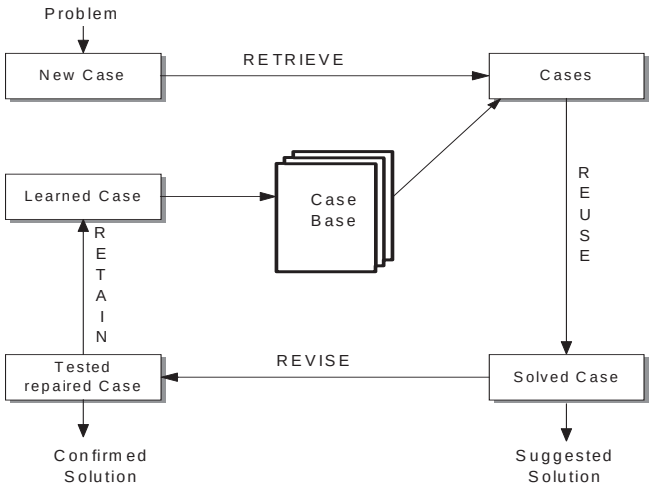


Figure 1. Case Base Reasoning cycle

# 4. The Proposed Algorithm

In this paper, the Parameters of $T_{nb}$ and $D_{nb}$ are two new criteria for choosing clusterhead. In this part, these two parameters are calculated by using CBR and then a stable clustering method will be presented in accordance with the computed parameters.

## 4.1. Definition of $D_{nb}$ & $T_{nb}$

To enhance stability of clusters we must find out problems that decrease the cause of stability and as a result cause a cluster to disappear. By solving 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 CH must have the following conditions:

- It must have a higher 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 weight factors. To meet the 2nd condition a node is chosen as a clusterhead, which has less distance with its neighbors during neighborhood duration (by using $D_{nb}$). $D_{nb}$ is average distance between clusterhead i and its neighbors. The smaller $D_{nb}$ (i) is, the less transmission power the node i requires for interaction and communication with its neighbors and as a result it consumes less battery power.

The second parameter which causes the clusters to be unstable is the nodes mobility. Node mobility has a speed and direction. $T_{nb}$ (i) is the average neighborhood duration of node i with its neighbors. The $T_{nb}$ has direct relation with stability of clusters.

## 4.2. Calculation the $T_{nb}$ and $D_{nb}$ Parameters Using Case Base Reasoning

The CBR approach for calculation of these amounts is presented as follows:

Case Representation: The clustering approach presented in this paper is based on the availability of position information via a reliable position locating system. Nodes execute a distributed algorithm of localization in order to build a global coordinates system [19].

Each node creates a control message named situation message through receiving its coordinates. This message describes in the Figure2.

| Node ID | Coordinates of the node |
|---|---|

Figure 2. Situation Message

Each node produces this message and sends it to its neighbors (only its neighbors) alternatively with $T_c$ alternation period. If the node A receives the situation message from the node B at the first time ($t_0$) and the second time ($t_1$), it will create a case for the node B. The case structure includes the properties in accordance with the Table1.

Table 1. Case structure

| |
|---|
| Situation: |
| f1: $(X_A(t_0),Y_A(t_0))$ Coordinate of node A in $t_0$<br>f2: $(X_B(t_0),Y_B(t_0))$ Coordinate of node B in $t_0$<br>f3: $(X_A(t_1),Y_A(t_1))$ Coordinate of node A in $t_1$ ($t_1=t_0+T_c$)<br>f4: $(X_B(t_1),Y_B(t_1))$ Coordinate of node B in $t_1$ ($t_1=t_0+T_c$) |
| Solution: |
| f5: $T_{nb}$ (A,B) Neighborhood duration A and B<br>f6: $D_{nb}$ (A,B) Average distance between A and B in $T_{nb}$ (A,B) |

The node A creates a local array and a local counter for all of its neighbors and checks its neighboring with them at the next steps ($t_i+1=t_i+T_c$). While node B is still a neighbor of node A, the Euclidean distance between the node A and B at step i ($D_i(A,B)$), is calculated and saved in the local array and local counter is increased a unit. If the node A loses its neighboring with the node B, then it calculates the amounts of $D_{nb}(A,B)$, $T_{nb}(A,B)$ according to equations 1 and 2 .

$$D_{nb}(A,B) = \frac{\sum_{i=0}^{N} D_i(A,B)}{N+1} \quad (1)$$

$$T_{nb}(A,B) = \text{local counter} * T_c \quad (2)$$

Then node A inserts $T_{nb}(A,B)$, $D_{nb}(A,B)$ in the case situation, where $T_{nb}(A,B)$, $D_{nb}(A,B)$ respectively denote the total neighboring time and average distance between node A and B.

Case Retrieving: The similarity of current problem with each case inside the case library is calculated according to the following formula:

$$sim(p,c) = 1 - \frac{dist(p,c)}{\max Range} \quad (3)$$

Where sim(p,c) is local similarity function, which computes the normalized similarity degree of each property of the current problem with its retrieved case property in the case library, and p,c are the values for features in the input and retrieved cases respectively. maxRange is the maximum possible distance between two nodes. disti(p,c) is the Euclidean distance function between ongoing problem coordinate and case coordinate as following :

$$dist_i(p,c) = \sqrt{(x_{ci}-x_{pi})^2 + (y_{ci}-y_{pi})^2} \quad i = 1,2,3,4 \quad (4)$$

This approach involves the assessment of similarity between stored cases and the new input case, based on matching a weighted sum of features. The biggest problem here is to determine the weights of the features $w_i$ . The limitation of this approach includes problems in converging on the correct solution and retrieval times. In general the use of this method leads to the retrieval time increasing linearly with the number of cases. Therefore this approach is more effective when the case base is relatively small. An algorithm for calculating nearest neighbor matching is introduced as equation (5), where w is the importance of a feature, sim is the similarity function and Pi ,Ci are the values for feature i in the input and retrieved cases respectively.

$$\rho(p,c) = \frac{\sum_{i=1}^{n} w_i \times sim_i(p_i,c_i)}{\sum_{i=1}^{n} w_i} \quad (5)$$

Similarities are normalized to fall within the range [0, 1], where zero is dissimilar and one being an exact match.

Through using the general similarity function, the similarity of the current problem to all the cases is compute and ten cases, which are more similar to the current problem, are chosen.

$D_{nb}$ (A, B) and $T_{nb}$ (A, B) are calculated as following:

$$T_{nb}(A,B) = \frac{\sum_{j=1}^{10} \rho_j \times T_{nb}(A,B)_j}{\sum_{j=1}^{10} \rho_j} \quad (6)$$

$$D_{nb}(A,B) = \frac{\sum_{j=1}^{10} \rho_j \times D_{nb}(A,B)_j}{\sum_{j=1}^{10} \rho_j} \quad (7)$$

j = 1 to 10

Case Retaining: In the retaining stage, it is time to check whether a case must be added in the case library or discard. To determine an answer, the similarity of this case to the all the cases inside a case base, is computed. If the average of these similarities exceeds a predefined threshold a new case is added to the case library otherwise it will be discarded.

Ultimately, the node A must calculate the average neighborhood duration ($T_{nb}$) and the average distance from all of its neighbors during the neighborhood duration ($D_{nb}$). Assume that the node A's neighbor are nodes E, D, C and B. The node A calculates two parameters $T_{nb}$ (A) and $D_{nb}$ (A) as follows:

$$T_{nb}(A) = \frac{T_{nb}(A,B) + T_{nb}(A,C) + T_{nb}(A,D) + T_{nb}(A,E)}{4} \quad (8)$$

$$D_{nb}(A) = \frac{D_{nb}(A,B) + D_{nb}(A,C) + D_{nb}(A,D) + D_{nb}(A,E)}{4} \quad (9)$$

## 4.3. Cluster Formation

In order to maintain an ad hoc network organized, the network can be divided in partitions demonstrated clusters of nodes. A cluster is a group of similar things (e.g. nodes in telecommunication systems), with similar characteristics. Then, protocol-based on a cluster system consist of a set of rules applying cluster partition, where nodes have autonomously organized themselves to forms clusters.

Each cluster contains a clusterhead zero or more ordinary nodes and one or more gateways. Clusterheads are nodes whose main functions are transmissions and allocation of resource within the cluster, for example, it might issue tokens to potential transmitters, emit busy tones when a transmission in progress, or assign slots to specific transmitters and sessions. Gateways connect adjacent clusters. A gateway may directly connect two clusters by acting as a member of both, or it may directly connect two clusters by acting as a member of one and forming a link to a member of the other.

Figure3 shows an example of the clustered network.

The clustering algorithms to be discussed are completely distributed and adaptive. Hence, they are all suitable for highly dynamic wireless ad hoc networks.

Election of the clusterheads is based on the weight values of the neighbor nodes. Each node calculates its weight value based on the following factors:

- $T_{nb}$: Average neighborhood duration
- $D_{nb}$: Average distance from the neighbors in neighborhood duration
- Degree difference (Δ): the difference between the actual number of neighbors and the number of nodes that a clusterhead can handle ideally.
- transmission power ($T_{RX}$)
- battery power (B)

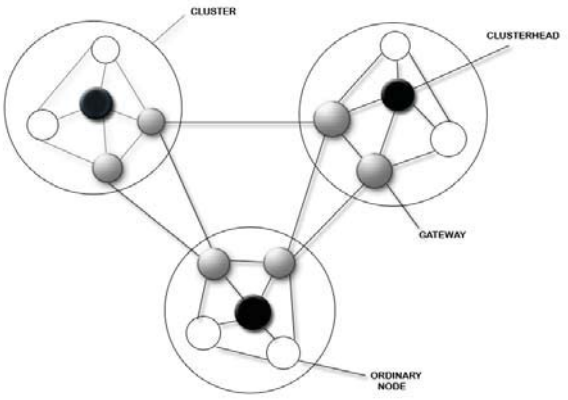


Figure 3. Example of cluster of ad hoc network

The weight calculation procedure consists of seven steps as described below:

**Step1**. Broadcast a Situation message to the neighbors.

**Step2**. Find degree, which is the number of neighbors of a node v ($d_V$).

**Step3**. Compute the degree-difference, $\Delta_v = | d_V - \sigma |$, for node v. where $d_V$, is the number of neighbors of a node v and σ is a pre-defined threshold for the number of nodes that a clusterhead can handle ideally.

**Step4**. Compute Transmission power of each node v. ($T_{RX}$)

**Step5**. Compute $T_{nb}$ (v) of each node v using CBR. (In accordance with the part 3.2)

**Step6**. Compute $D_{nb}$ (v) of each node v using CBR. (In accordance with the part 3.2)

**Step7**. Calculate the composed weight $W_V$ for node v:

$W1(v) = w1*T_{nb}(v) + w2/D_{nb}(v) + w3*B_v + w4/\Delta_v + w5*T_{RXv}$

Where, w1, w2, w3 and w4 are the weighting factors. After finding its weight value, each node broadcasts the weight value using a Weight_value message to its one-hop neighbors only. Only the node with the largest weight among neighbors can declare to become a CH.

At this time, it broadcasts a CH message as shown in Figure4, which indicates it is a CH.

| Node ID | battery power | transmission power | Degree difference |
|---|---|---|---|

Figure 4. CH Message

If the node u receives a CH message from node v and decides to join cluster v, it sends a JOIN (u, v) message to declare that it will join cluster v and become an ordinary node.

If the node u receives CH messages from the nodes v, w, and z, it will compute a second weight for them.

The node u computes the second weight for the node v as follows:

$W2(v)=\alpha1*T_{nb}(u,v)+\alpha2/D_{nb}(u,v)+\alpha3*B_v+\alpha4/\Delta_v+\alpha5*T_{RXv}$

In this equation:

α1, α2, α3, α4, and α5 are the weighting factors.

Bv is the battery power of the node v.

Δv is the degree-difference of the node v.

$T_{RXv}$ is transmission power of the node v.

$T_{nb}$ (u, v) is the neighborhood duration of the nodes u and v.

$D_{nb}$ (u, v) is the average distance of the node u with the node v during the neighborhood duration with it.

These two parameters have been computed for calculation of the first weighted causes by the node and it is not necessary to calculate them again. These three parameters are computed through CH message.

Using the second weight causes the node u chooses best node as its CH. This CH must have better physical characteristics and secondly it must be more stable and have less relative distance in comparison with the node u. The node u, computes the second weight for each of the nodes v, w and z and saves it along with node id in a list (list Γ) in a declining way. The node u sends a Join_Req message to the first node of the list Γ.

As soon as each of the CHs receives the Join_Req message, it will send the Join_Ack message to the node u if its Δ is opposed to 0. Otherwise it will send the Join_Refuse message to the node u and the node u will send the Join_Req message to the next node existing in the list Γ.

If the node u does not find any clusterhead for itself it will choose itself as the clusterhead and broadcasts the CH message.

When an ordinary node u moves to the outside of its cluster boundary, it is required to find a new clusterhead to affiliate with. It will broadcast a CH_Req message. After receiving this message each of the clusterheads such as v will broadcast CH message and Situation message to the node u. In this mode the best clusterhead is chosen by the node u in accordance with the second weight. In this case, $T_{nb}$ and $D_{nb}$ are not existent in advance and must be calculated in accordance with situation message.

If node u finds a new clusterhead, it hands over to the new one. If not, it declares itself as a clusterhead.

# 5. Experimental Results

The simulation is run using NS-2 and MATLAB. For getting better simulation result, first generated traffic in NS-2 then import traffic files into MATLAB simulator.

## 5.1. Evaluation of Case Base Reasoning System in Proposed Algorithm

When applying CBR systems for problem solving, there is always a trade-off between the number of cases to be stored in the case library and the retrieval efficiency. The larger the case base, the greater the problem space covered.

The increase of the number of the cases inside the casebase is useful as long as it decreases the amount of the errors of the system and enhances its efficiency.

To calculate the number of the ideal cases and to evaluate the CBR efficiency, two experiments were conducted.

At first, 11000 cases were produced.

In the first step, 800 cases out of 11000 cases were chosen randomly as the trainset and placed in the case library and the 10200 remaining cases were placed as the testset. Then each of the cases inside the Testset was considered as an ongoing problem and then a solution was produced by compositional adaptation through retrieved cases from trainset. The amount of errors of the CBR system for each case is computed comparing the solution with the real one. The amount of the computational error for each ongoing problem solving according to the (10) is computed.

$$e=\frac{\frac{\left|T_{nb}^{retrieved}-T_{nb}^{real}\right|}{\max Range1}+\frac{\left|D_{nb}^{retrieved}-D_{nb}^{real}\right|}{\max Range2}}{2} \quad (10)$$

Average error for testset is also computed through the equation (11).

$$e_{avg}=\frac{\sum_{i=1}^{n}e_i}{n} \quad (11)$$

Where n is the testset number.

The Performance of the CBR system is computed as follows:

$$performance=1-e_{avg} \quad (12)$$

In the first trial 800 cases were selected randomly to constitute the case library, and 10200 problem situations were then tested. In the next trial, the number of the cases in case library was increased to 1600 and 9400 problem situations was this time tested. This process of increasing the cases and decreasing the number of problem situations was continued until the stage where the cases number and problem situations number become 10400 and 600 respectively.

Table2 briefly illustrates results of simulation for each epoch of experiment.

Table 2. Value of experimental result.

| step | Trainset# | Testset# | performance | Retain# |
|---|---|---|---|---|
| 1 | 800 | 10200 | 0.2 | 76 |
| 2 | 1600 | 9400 | 0.22 | 85 |
| 3 | 2400 | 8600 | 0.31 | 92 |
| 4 | 3200 | 7800 | 0.4 | 105 |
| 5 | 4000 | 7000 | 0.56 | 121 |
| 6 | 4800 | 6200 | 0.68 | 126 |
| 7 | 5600 | 5400 | 0.74 | 115 |
| 8 | 6400 | 4600 | 0.76 | 118 |
| 9 | 7200 | 3800 | 0.79 | 125 |
| 10 | 8000 | 3000 | 0.82 | 112 |
| 11 | 8800 | 2200 | 0.81 | 107 |
| 12 | 9600 | 1400 | 0.83 | 105 |
| 13 | 10400 | 600 | 0.84 | 101 |

Projecting the results of Table2 onto the curve of Figures5 and 6, it is easily seen that this curve reaches a

saturation characteristic around the point of 5000 (as the number of cases), satisfying an average performance of 0.7, which is quite satisfactory. The storage property of the system when increasing the cases in case base will be constant at the saturation point (5000 case#).

Figure5 shows the performance of the CBR system according to the cases inside the case base in the trials.

In accordance with the Figures5 and 6 the performance of the system has promoted from 84% (during the first test) to 91% (during the second test).

Figure7 shows the computation time cost according to the number of cases inside the case library. In accordance with the Figure7 the computation time will be increase due to the increasing of number of cases inside the case library.

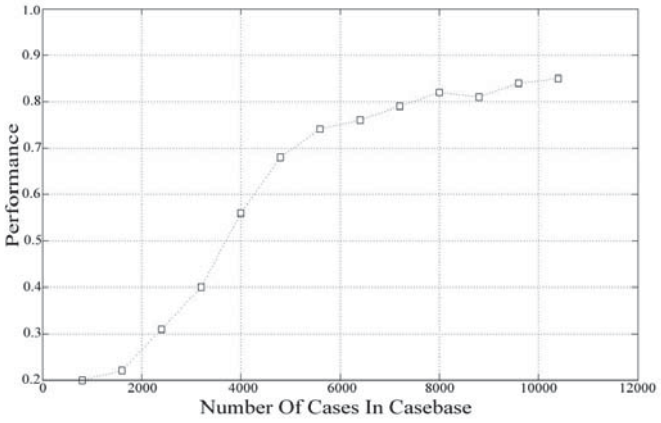


Figure 5. Performance of CBR system without retaining cases

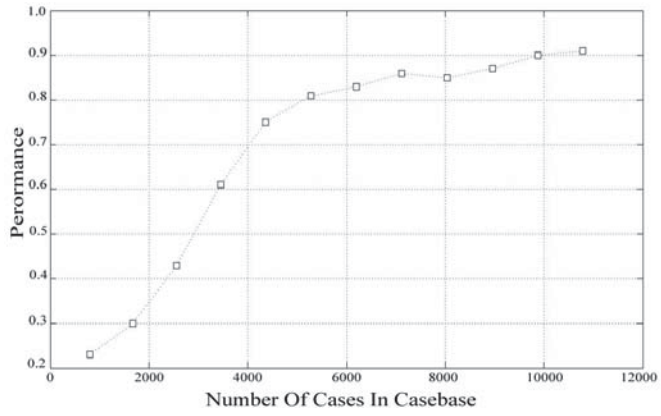


Figure 6. Performance of CBR system with retaining cases

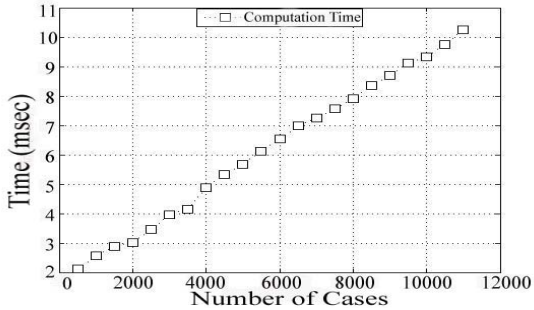


Figure 7. Computation time according to the number of cases inside the case library

## 5.2. Performance of Proposed Algorithm

The measured performance of the proposed algorithm is compared with that of Highest Degree (HD), Lowest-Identifier (LID) and Weighted Clustering Algorithm (WCA).

A system of 25 mobile nodes is considered in an 800m*800m area. Radio transmission range for each node is set to 250 meters. Nodes movement used model is according to the modified random waypoint model [20]. The routing protocol is AODV.

A source node selects its destination randomly and sends traffic for it. In this simulation two traffic model are used:

- Constant Bit Rate (CBR) traffic through UDP with a rate of 2 packets per second. Each packet carries 512 bytes of data payload.
- Self Similar (SS).

In our experiments, the values of weighting factors used in the first and second weight calculation are set to:

$w_1 = 0.2$, $w_2 = 0.3$, $w_3 = 0.2$, $w_4 = 0.1$, $w_5 = 0.2$,
$\alpha_1 = 0.3$, $\alpha_2 = 0.3$, $\alpha_3 = 0.2$, $\alpha_4 = 0.1$ and $\alpha_5 = 0.1$.

To determine the best values of $w_1$ to $w_5$, we assumes that $\alpha_1$ to $\alpha_5$ are constant, then we implement the experience randomly for each value of
$W = \{w_1, w_2, w_3, w_4, w_5 | w_1+w_2+w_3+w_4+w_5=1 \& w_i = x*0.1$ that $x \in N_0\}$, 100 times.

The best values of $w_1$, $w_2$, $w_3$, $w_4$ and $w_5$ are the W that its average results are best. Then we Use these values for $W_i$, calculated the best values of $\alpha_i$ same as above.

Also the ideal degree for each clusterhead, $\sigma$, is set to 3.

To measure the performance of our algorithm, the following two metrics in considered:

- Total network power consumption
- The number of reaffiliations

Each simulation is performed in the duration of 100s and each result is obtained from the average of 2500 simulation runs.

The result of the number of reaffiliations due to node velocity is depicted in Figures8 and 9. In Figure8 the traffic is based on constant bit rate traffic model and in Figure9 the Self Similar traffic model has been used.

The Figures 8 and 9 show the number of Reaffiliations will increase due to the increase of maximum speed of the nodes.

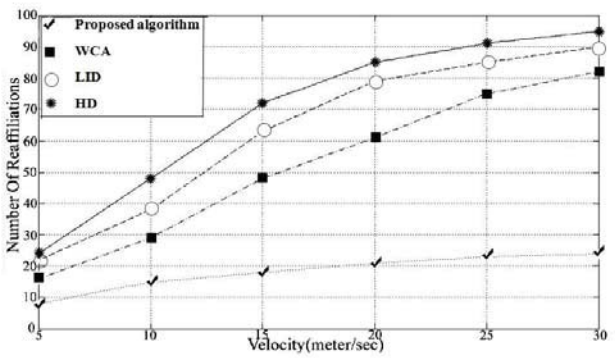


Figure 8. Number of reaffiliations with Constant Bit Rate traffic

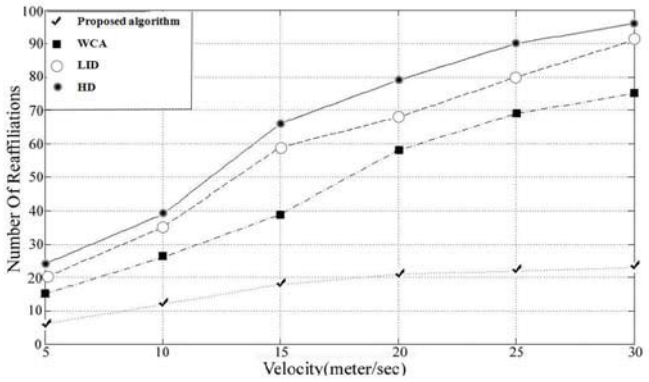


Figure 9. Number of reaffiliations with Self Similar (SS) traffic

This increase is much less in the proposed algorithm in compare to other previous algorithms (LID, HD and WCA) and this is due to using of $T_{nb}$ parameter as one of the weight factors. Using this parameter causes the neighborhood duration of the ordinary nodes with their CHs, to increase and as a result the number of reaffiliations will decrease.

According to two last figures, the type of traffic model has not so much more effective on the number of reaffiliations. The Figures10 and 11 show the amount of power consumed by the entire network during the simulation time.

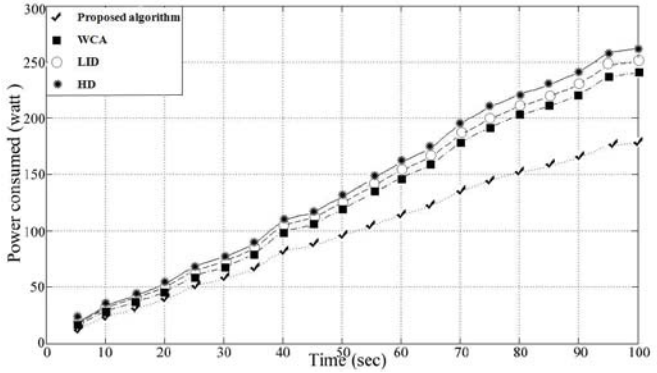


Figure 10. Power consumption with Constant Bit Rate traffic

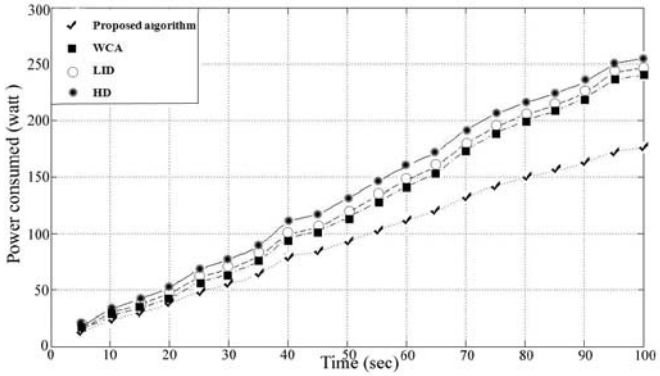


Figure 11. Power consumption with Self Similar (SS) traffic

These two figures show that the amount of power consumed by the whole network increases due to passage of simulation time. This increase is much less in the proposed algorithm in compare to other previous algorithms (LID, HD and WCA) due to using $D_{nb}$ parameter as one of the weight factors. Using this parameter causes average distance of the ordinary nodes with their CHs to decrease during the neighborhood duration and as a result interaction between them and their clusterheads requires less power. According to the Figures10 and 11, we conclude that the type of traffic model is not very effective on the Power consumption in total network.

The experimental results show that our proposed algorithm not only is more stable rather than evaluated clustering algorithms (low reaffiliations rate) but also it consume less battery power in compare to others. These improvements obtained by definition and estimation two new metrics; $T_{nb}$ and $D_{nb}$.

# 6. Conclusion

Clustering can provide large-scale MANETs with a hierarchical network structure to facilitate routing operations. A distributed clustering algorithm was proposed for wireless ad hoc networks. The simulation results of the proposed algorithm showed better stability of the system in comparison to other algorithms. Another feature of this algorithm is its simplicity.

In this paper, two parameters named $T_{nb}$ and $D_{nb}$ were defined. Using the first parameter causes the nodes to remain in the coverage area of their CH longer and using the second parameter causes the CHs to consume less power for interaction with their neighbors. Furthermore two weights were defined. The first weight causes the best node to be chosen as the CH and the second weight causes each of the ordinary nodes to choose the best CH from among their neighbors. The proposed algorithm was compared and contrasted with algorithms of HD, LID, WCA in terms of the number of reaffiliations and amount of the power consumed by the entire network. The results showed that in both cases the efficiency of our algorithm is better than the mentioned algorithms.

# Acknowledgment

Sincere gratitude is hereby extended to following who never ceased in helping until this paper is structured:
Kouhyar Tavakolian, Ph. D Candidate, Centre for Integrative Bio Engineering Research (CiBER), Simon Fraser University, Canada, Kouhyart@sfu.ca.

# References

[1] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *in Proceedings of IEEE*, pp. 56-73, 1987.

[2] D. B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," *Workshop on Mobile Computing and Applications*, 1997.

[3] D. J. Baker, and A. Ephremides, "A distributed algorithm for organizing mobile radio telecommunication networks," *in 2nd International Conference on Distributed Computer Systems*, Paris, France, IEEE, pp. 476-483, 1981.

[4] D. W. Aha, *Case-Based Learning Algorithms,* In: Bareiss, pp.147-158, 1991.

[5] C. K. Riesbeck, and R. C. Schank, *Inside Case-based Reasoning,* 1989.

[6] D. J. Baker, and A. Ephremides, "A distributed algorithm for organizing mobile radio telecommunication networks," *in 2nd International Conference on Distributed Computer Systems,* Paris, France, IEEE, pp. 476-483, 1981.

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

[8] S. K. Dhurandher, and G. V. Singh, "Weight Based Adaptive Clustering in Wireless Ad Hoc Networks," *Personal Wireless Communications,* 2005.

[9] W. Choi, and M. Woo, "A Distributed Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *International Conference on Internet and Web Applications and Services/Advanced,* IEEE, 2006.

[10] M. Gerla, and J. T. C. Tsai, "Multicluster, mobile, multimedia radio network," *ACM/Baltzer Journal of Wireless Networks,* pp. 255-265, 1995.

[11] G. Schiele, C. Becker, and K. Rothermel "Energy-Efficient Cluster-based Service Discovery," *In: Proceedings of the 11th SIGOPS European Workshop (SIGOPSEW04);* Leuven, Belgium, September, 2004.

[12] J. Tian, J. Hähner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-Based Mobility Model for Mobile Ad Hoc Network Simulation," *In: IEEE Computer Society Press (ed.): Proceedings of the 35th Annual Simulation Symposium:* San Diego, California, April, 2002.

[13] K. Liu, J. Su, J. Zhang, F. Liu, and C. Gong "A novel stable cluster protocol for mobile ad hoc networks," *in Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications,* IEEE, 2005.

[14] H. Ali, W. Shahzad, and F. A. Khan, "Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization," *App l. Soft Comput. J,* 2011.

[15] A. Sampath, C. Tripi, and M. Sabu, "An ACO Algorithm for Effective Cluster Head Selection," *Journal of Advanes in Information Technology,* vol. 2, no. 1, 2011.

[16] C. Konstantopoulos, D. Gavalas, and G. Pantziou, "Clustering in mobile ad-hoc networks through neighborhood stability-based mobility prediction," *Computer Networks,* vol. 52, pp. 1797-1824, 2008.

[17] D. J. Baker, and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Transactions on Communications COM-29,* pp. 1694–1701, 1981.

[18] A. Dana, A. Khadem Zadeh, M. E. Kalantari, and K. Badie, "A Traffic Splitting Restoration Scheme for MPLS Network Using Case-Based Reasoning," *Proc. of the 9th IEEE Asia Pacific Conference on Communication (IEEE APCC 2003),* Malaysia, September 2003.

[19] R. Mellier, J. F. Myoupo, I. Sow, Univ. de Picardie-Jules Verne, Amiens, "GPS-Free Geocasting Algorithms in Mobile Ad hoc Networks," *Wireless and Mobile Communications, 2006 ICWMC '06. International Conference on Publication,* July, 2006.

[20] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," *INFOCOM 2003. Twenty-Second Annual Joint Conferences of the IEEE Computer and Communications Societies.* IEEE, April, 2003.

**Reza Assareh** received his B.Sc and M.Sc. in Computer Engineering from Islamic Azad University respectively in 2004 and 2007. He is Ph.D student in Computer Engineering at Islamic Azad University, Research & Science Branch, Tehran, Iran right now. Assareh is a lecturer at Islamic Azad University Shahr-e-Rey Branch. His research interests are in the areas of clustering and power consumption in ad-hoc wireless network, anti-collision protocols in RFID system and intelligent control mechanism (Cellular Automata and Learning Automata).
**E-mail**: reza_assareh@yahoo.com

**Ali Assarian** received his B.Sc and M.Sc. in Computer Engineering from Islamic Azad University respectively in 2004 and 2007. He is Ph.D student in Computer Engineering at Islamic Azad University, Research & Science Branch, Tehran, Iran right now. Assarian is a lecturer at Islamic Azad University East Tehran Branch. His research interests are in the areas of clustering and power consumption in ad-hoc wireless network, anti-collision protocols in RFID system and intelligent control mechanism (Cellular Automata and Learning Automata).
**E-mail**: ali_assarian@yahoo.com

**Arash Dana** received his B.Sc and M.Sc.in Electrical Engineering from Islamic Azad University respectively in 1993 and 1996 and Ph.D in Electrical Engineering from Islamic Azad University, Research & Science Branch, Tehran, Iran in 2003. Dr. Dana is a lecturer at Tehran universities. His research interests are in the areas of quality of service provisioning in high speed networks, wireless communications with special emphasis on design of reliable and efficient wireless networks, ad-hoc wireless network, self-similarity and traffic modeling.
**E-mail**: a_dana@iauctb.ac.ir

**Ahmad Khadem Zadeh** was born in Meshed, Iran, in 1943. He received the B.Sc. degree in applied physics from Ferdowsi University, Meshed, Iran, in 1969 and the M.Sc., Ph.D. degrees respectively in Digital Communication and Information Theory & Error Control Coding from the University of Kent, Canterbury, U.K. He is currently the Head of Education & National Scientific and Informational Scientific Cooperation Department at Iran Telecom Research Center (ITRC). He was the head of Test Engineering Group and the director of Computer and Communication Department at ITRC. He is also a lecturer at Tehran Universities & he is a committee member of Iranian Computer society and also a committee member of the Iranian Electrical Engineering Conference Permanent Committee. Dr. Khadem Zadeh has been received four distinguished national and international awards including Kharazmi International Award, and has been selected as the National outstanding researcher of the Iran Ministry of Information and Communication Technology.

**E-mail**: zadeh@itrc.ac.ir

**Paper Handling Data:**

Submitted: 08.06.2009
Received in revised form: 24.10.2012
Accepted: 04.11.2012
Corresponding author: Reza Assareh,
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.