CBPMD: A New Weighted Distributed Clustering Algorithm for Mobile Ad hoc Networks (MANETs)

[1]Ayman Bassam Nassuora and [2]Abdel-Rahman H. Hussein

Jarash University, Jarash (Jordan)

E-mail: nassuora@yahoo.com, abed_90@yahoo.com

Abstract

Clustering approach is an important research topic for MANETs and widely used in efficient network management, hierarchical routing protocol design, network modeling, Quality of Service, etc. Many researchers' recent focus has been on clustering management which is one of the fundamental problems in mobile ad hoc networks. The main objective of clustering in mobile ad-hoc network environments is how can an optimal clusterhead be elected and how can the optimal number of clusters be achieved through division without degrading the whole network's performance. In this paper, we propose new weighted distributed clustering algorithm, called CBPMD. It takes into consideration the parameters: connectivity (C), residual battery power (BP), average mobility (M), and distance (D) of the nodes to choose locally optimal clusterheads. The goals of this algorithm are maintaining stable clustering structure with a lowest number of clusters formed, to minimise the overhead for the clustering formation and maintenance and to maximise the lifespan of mobile nodes in the system. Simulation experiments are conducted to evaluate the performance of our algorithm in terms of the number of clusters formed, reaffiliation count and numbers of clusterhead changes. Results show that our algorithm performs better than existing ones and is also tuneable to different kinds of network conditions.

Keywords – Mobile Ad-Hoc Networks, Weighted Distributed Clustering Algorithm, CBPMD algorithm.

1. Introduction

Scalability is of particular interest to ad hoc network designers and users and is an issue with critical influence on capability and capacity. Where topologies include large numbers of nodes, routing packets will demand a large percentage of the limited wireless bandwidth and this is exaggerated and exacerbated by the mobility feature often resulting in a high frequency of failure regarding wireless links. To overcome such barriers to success and address the issues of scalability and maintenance of MANETs it is essential, "to build hierarchies among the nodes, such that the network topology can be abstracted. This process is commonly referred to as clustering and the substructures that are collapsed in higher levels are called clusters."[1, 2]. Increasing network capacity and reducing the routing overhead through clustering brings more efficiency and effectiveness to scalability in relation to node numbers and the necessity for high mobility. The manager node- CH (Clusterhead) - in clustering has responsibility for many functions such as cluster maintenance, routing table updates, and the discovery of new routes. However, the recurrent changes faced by the clusterhead can lead to losing stored routing information, route changes between node pairs and ultimately impacts on the overall performance of the routing protocol because of cluster structure instability [3]. For these reasons this paper will focus on how to elect a clusterhead to keep the stability of network topology.

In addition, an efficient and effective method of updating a cluster structure in response to underlying network topology change is imperative because without cluster stability the performance of a clustering scheme in a dynamic, mobile environment is affected – usually adversely. It is preferable to have fewer invocations of the algorithm. Recurrent membership updating and especially, repeated clusterhead alteration unfavorably affect the cluster stability and can interfere with the performance guarantee of protocols residing on the cluster structure, such as dynamic routing and channel assignment. Hence, frequent updating is expensive because it introduces extra clustering-related control overheads, invalidation of existing routes and even a ripple effect of re-clustering, which causes cluster structure re-building over the entire network [4, 5].

With proposed algorithms and protocols contributing towards the achievement of stability and reliability of clustering in mobile ad hoc networks, solutions have been provided in two main areas: clustering and topology management. Stability management crucially begins from an optimal partitioning of the network into clusters. Once these are established, routes between node pairs can be upheld through evaluating and reviewing the route lifetime based on the node mobility pattern. Network topology and scalability is also maintained through the identification of vital nodes (CH) to take on a role in the network operations, such as routing. The goals of this algorithm are maintaining stable clustering structure with a lowest number of clusters formed, to minimise the number of invocations for the clustering formation/maintenance and to maximise the lifespan of mobile nodes in the system.

In this paper, we proposed new weighted distributed clustering algorithm called CBPMD. The proposed algorithm is an extended version of Weighted Clustering Algorithm (***CEMCA***) [6]. The merits of CBPMD algorithm are described as follows. First of all, a novel weight function was introduced that can be used to select suitable and optimal clusterheads based on these parameters: connectivity (C), residual battery power (B), average mobility (M), and distance (D) and addressed explicitly for how to normalise these parameters. Secondly, satisfying the load balancing between the clusters and reducing the number of clusters formed by specifying the maximum and minimum number of nodes that a clusterhead can ideally handle followed. Thirdly, each mobile node starts to measure its weight after n (small integer in order to minimise memory requirement) successive ‘hello messages’, where the result specifies the accurate value for the mobility and battery power. Fourthly, the nodes with the largest local weights are elected as clusterheads. Finally, as the nodes have various battery powers to start with, a more accurate metric would be to measure the power currently available at the node. The main goals in this research were achieved using the weighted distributed clustering algorithm CBPMD. It was used to elect optimal clusterheads and divide optimal number of clusters without degrading the whole network performance, satisfying the load balancing between clusters, reducing the communication overhead and minimising the explicit control messages caused by cluster maintenance, maximising the stability of clustering to improve network life time.

The remainder of this paper is organized as follows. In Section 2, we review relevant several clustering algorithms proposed previously and its limitation. Section 3 presents the proposed algorithm for ad hoc networks. Discussion of the simulation results and analysis presented in section 4. Finally, Section 5 concludes this paper.

2. RELATED WORK

Recently, a number of clustering algorithms have been proposed and based on some of criteria to choose clusterhead such as speed and direction (mobility), battery power, distance, and the number of neighbors of a given node. These works present advantages but some drawbacks as a high computational overhead for both clustering algorithm execution and update operations (maintenance). In this section, an overview of the existing clustering algorithms for mobile ad hoc networks is presented.

Entropy-Based Weighted Clustering Algorithm [7] Entropy based clustering overcomes the drawback of WCA‘s high reaffiliation rates that contribute to higher communication overhead (Chatterjee et al., 2002) and forms a more stable network. It uses an entropy based model (originally founded in thermodynamics Second law) whereby measurement of ―the level of disorder in a closed but changing system in which energy can only be transferred from an ordered state to a disordered state shows that the higher the entropy, the higher the disorder and the lower the availability of the system‘s energy to do useful work.‖ (Definition: BusinessDictionary.com, 2010) evaluates route stability in ad hoc networks and the election of a clusterhead. By evaluating this dynamic a better indication of the stability and mobility of the ad hoc network can be achieved.

The Connectivity, energy & mobility driven weighted clustering algorithm (CEMCA) [6] To elect a cluster, the combination of particularly significant metrics is considered with attention focused on the extremes of the metrics rather than their existence or distribution per se. Characteristics such as the lowest node mobility, the highest node degree, the highest battery energy and the best transmission range are factored together. Total distribution of the algorithm throughout the network ensures every node having equal opportunity to assume the role of clusterhead. CEMCA is composed of two main stages. Following the election of the cluster head members are then grouped into a cluster. Election of clusterhead is based on calculated normalized values (normalized to 1) of mobility, degree and energy level for each node. Through broadcast to neighbors to compare node quality the best is selected for clusterhead. The selected clusterheads define their neighbors at two hops maximum to form their cluster membership and they store all member information just as all nodes record their clusterhead identifier. This information exchange permits the routing protocol to function both within a cluster and between the cluster groups.

The **Weighted Clustering Algorithm (WCA)** [8] obtains 1-hop clusters with one cluster-head. The election of the cluster-head is based on the weight of each node. It takes four factors into consideration and makes the selection of cluster-head and maintenance of cluster more reasonable. The four factors are node degree (number of neighbors), distance summation to all its neighboring nodes, mobility and remaining battery power. Although WCA has proved better performance than all the previous algorithms, it lacks a drawback in knowing the weights of all the nodes before starting the clustering process and in draining the CHs rapidly. As a result, the overhead induced by WCA is very high. However, how to normalize the factors is not addressed explicitly and the cumulative time of a node already serving as a clusterhead cannot accurately reflect the current level of battery power because a busy node may almost run out of power and it has never been a clusterhead.

3. THE PROPOSED MODEL

WEIGHTED DISTRIBUTED CLUSTERING ALGORITHM (CBPMD)

This section focuses on the factors used to select a clusterhead. First discussed are the philosophy, and the basics of clustering algorithm before presenting more detailed information.

3.1 BASIS OF OUR METRIC COMPONENTS

The literature reviewed from previous research[2,3,9], reveals that most clustering algorithms, apart from the combined metric scheme, consider only one of the factors when addressing clusterhead selection, such as the node degree, connectivity, mobility, remaining battery power and so on. As one is problem solved, another problem created and most of the approaches for finding and electing the clusterheads fail to produce an optimal solution with respect to battery usage, load balancing, stability, and MAC functionality. This research proposes an improved clustering algorithm inspired by the fundamental idea of combined metric. But, before computing a metric, there are several questions that need to be answered "what purpose does clustering serve", "how can the generated overhead be minimized"," how can the stability of clustering and maximization of network life time be improved" and "how can the load balancing between clusters be satisfied". In the solution offered, the focus is better network management, in several areas (security, administration, transmission management, routing…etc). On the supposition that clusterheads will collaboratively ensure management tasks, deciding how well a node is suited for being a clusterhead, its connectivity, distance, mobility and battery power must be taken into account. The following points briefly describe metrics considered in the proposed clustering algorithm:

- The ***clustering algorithm*** (formation or maintenance) is not invoked if the nodes don't exist from the transmission range of their master clusterheads.
- The ***weighting factor*** is a generic parameter used in the decision of selecting a clusterhead. The node having the greatest weight is elected as clusterhead.
- The ***max value*** denotes the upper limit of the number of nodes simultaneously capable of support by a cluster-head. In other words, specifying a pre-defined limit on the number of nodes that a clusterhead can ideally handle, thus ensuring that none of the clusterheads are overloaded at any given time. A high system throughput can be achieved by limiting or optimizing the *degree* of each cluster.
- The ***min value*** denotes the lower limit of the number of nodes belonging to a given cluster before invoking to the merging algorithm. The min value therefore, may avoid the inherent complexity of the management of greater numbers of clusters.
- The ***residual battery power*** can be efficiently utilized within certain transmission ranges, i.e., it takes less power for nodes to communicate with others if they share close proximity. A clusterhead consumes more battery power than an ordinary node since a clusterhead has extra responsibilities to carry out for its members.
- ***Mobility*** is a crucial element in deciding the clusterheads. In order to avoid frequent clusterhead changes, it is advantageous to elect a slow-moving clusterhead. When a clusterhead moves fast, the nodes may detach from it resulting in a *reaffiliation*, which occurs when an ordinary node moves out of a cluster and joins another existing cluster and as such, the amount of information exchange is limited between the node and the new corresponding clusterhead as local and comparatively small. The subsequent information

update demanded in the event of a change in the clusterheads is much more than merely reaffiliation.

- The ***communication ability*** *(connectivity)* of a clusterhead is more efficient and effective with neighbors that have closer proximity to it within the transmission range. As the nodes move away from the clusterhead, signal attenuation from the increasing distance can be detrimental to the communication.
- Finally, these parameters play a decisive role in the proposed model and are described in more detail in the following section.

3.2 THE METRIC COMPONENTS AND COMBINED WEIGHT

- **Connectivity Metric (C)**

The first parameter is the connectivity. Neither nodes with the highest connectivity nor the lowest should be elected as clusterheads. The former will be congested and their battery power will drop rapidly and the latter will have a low cluster size and the advantages of clustering will be unable to be exploited. The problem in the first case can be solved by determining the number of nodes that can simultaneously be supported by a clusterhead. This gives a measure of connectivity and is denoted by C, as

$$C = \sum_{j=1, j\neq i}^{N} L_{i,j}(t) \ldots\ldots\ldots\ldots (1).$$

Let N be the number of nodes in the system.
Let $L_{i,j}(t)$ be the indication that whether node i is a neighbor of node j, at time t; $L_{i,j}(t) = \{0,1\}$.
Where, $L_{i,j}(t) = 0$; If node i is not a neighbor of node j at time t,
$= 1$; If node i is a neighbor of node j at time t.
It is assumed that all links between any two neighbors are bi-directional.

Why choose this parameter: This particular parameter helps to decrease the unique distribution of clusterheads (i.e. the average ratio of number of clusterheads to the total number of nodes). It can also help identify topologically disadvantageous nodes (i.e. nodes nearby partition borders can be assumed to leave the partition sooner than more centrally located ones). Therefore, a node with the largest connectivity can perform well as clusterhead.

- **Residual Battery power metric (RBP)**

The second parameter is the residual battery power. Mobile nodes in a MANET usually depend on battery power supply; therefore prolonging a network’s lifespan by reducing energy consumption is an attractive proposition and as CH as team leader and administrator carries extra responsibilities and performs more tasks compared with ordinary members it is likely to “die” early because of excessive energy consumption. These ‘deaths’, or ‘dies’, diminish the effectiveness of the network; a deficiency of mobile nodes due to energy depletion may cause network partition and communication interruption. Hence, it is vital to balance the energy consumption among nodes to avoid node failures, especially when the network density is comparatively sparse. Also, the *battery power* (set residual battery power of the node i as RBP_i) can be resourcefully used within an optimum transmission range, i.e., nodes within close proximity will require less power to communicate with other nodes. Hence, residual battery power is a better measure than consumed battery power [10] or the cumulative time during which the node acts as clusterhead [6]. But long-term service as a clusterhead can cause reduction in the battery power and hence RBP_i of the current clusterhead is to be calculated periodically. The objective is to avoid this detriment where the total collapse of the current network topology may result and reduce the number of clusterhead elections and cluster formations. Finally, a node with high residual battery power RBP_i can perform well as clusterhead for a longer duration.

Why choose this parameter: Playing the leading role, in the communication process, means that the energy consumption of clusterheads is greater than that of ordinary nodes. An "early death" of a clusterhead necessitates additional clusterhead elections, increasing the traffic of control packets in the network. Therefore, a node with sufficient battery power to survive a predicted term is to be selected as clusterhead to reduce the amount of overhead incurred during clusterhead re-election and to avoid the premature demise of nodes.

- **Average mobility metric (M)**

The third parameter is the average mobility of the node. In most of the currently offered schemes addressing this property, the estimation of average node mobility *M* demands a GPS card with adequate precision to be mounted on every mobile node. This paper presents an alternative method for measuring *M* which relaxes mobile nodes from such requirement. Each node *i* measures its own average mobility M_i, used to calculate the weighted function value, in formula (4). This is achieved through contrasting the topology information it obtains during successive Hello messages (HMs). Mobile nodes maintain a short 'Neighbor Record Table' (NRT); NRT rows comprise vectors representing the IDs of neighboring nodes, where each NRT row refers to different HM. Calculated M_i value actually represents the values recorded by *i* during the latest *n* HMs (where *n* is a small integer in order to minimize memory requirement):

$$M_{i,n} = \frac{1}{n-1}\sum_{i=n}^{1}(i-1)\left|NRT_{t(i+1)} - NRT_{t(i)}\right| \dots\dots\dots\dots (2).$$

Where *t* denotes the current time. The coefficient $(i-1)$ increases the weight of recent over older node movements on calculated $M_{i,n}$ values since the former are regarded as more reliable indicators of future mobility trends.

Fig. 1 (a-d) illustrates how mobile node with ID = 4 and 6 moves on the plane; as a result of that movement (and the movement of other network nodes), its neighboring nodes (i.e. those within its transmission range) differ at the end of every HM. For this particular example, the 'neighborhood record' of node #4 at the end of four successive HMs: are NRT_1 = {1, 21, 5, 10, 7}, NRT_2 = {1, 21, 5, 11}, NRT_3 = {1, 5, 11}, NRT_4 = {1, 21, 5, 9}. Hence, the average mobility of node #4 within this period of time is given by:

$$M_{4,4} = \frac{1}{4-1}[\sum_{i=4}^{1}(i-1)\left|NRT_{t(i+1)} - NRT_{t(i)}\right|$$

$$= \tfrac{1}{3}[\,3 * |NRT_4 - NRT_3| + 2 * |NRT_3 - NRT_2| + 1 * |NRT_2 - NRT_1| + 0]$$

$$= \frac{1}{3}[\,3 * 3 + 2 * 1 + 1 * 3] = 4.667$$

Also, the 'neighborhood record' of node # 6 at the end of four successive HMs: are NRT_1 = {2, 11}, NRT_2 = {2, 7, 8}, NRT_3 = {2, 7, 9},NRT_4 = {2, 7}. Hence, the average mobility of node #4 within this period of time is given by:

$$= \frac{1}{4-1}\left[\sum_{i=4}^{1}(i-1)\left|NRT_{t(i+1)} - NRT_{t(i)}\right|\right.$$

$$= \frac{1}{3}[\,3 * |NRT_4 - NRT_3| + 2 * |NRT_3 - NRT_2| + 1 * |NRT_2 - NRT_1| + 0]$$

$$= \frac{1}{3}[\,3 * 1 + 2 * 2 + 1 * 3] = 3.333$$

From the above result, node #6 is more stable than node # 4 because the average mobility of node #6 less than node #4. Table 1 and 2, presents the NRT of node #4 and node # 6 respectively, illustrating how its neighborhood changes and how its average mobility is evaluated over those four successive HMs.

Table 1: Instance of the neighbor node table of node #4

Hello Message #	Neighboring Nodes	NRT_i
1	1 21 5 10 7 - -	0
2	1 21 5 - - 11 -	3
3	1 - 5 - - 11 -	1
4	1 21 5 - - - 9	3

Table 2: Instance of the neighbor node table of node #6

Hello Message #	Neighboring Nodes	NRT_i
1	2 11 - - -	0
2	2 - 7 8 -	3
3	2 - 7 - 9	2
4	2 - 7 - -	1

Why choose this parameter: The main characteristic of the ad hoc network is its dynamic topology, therefore, there must be an adaptation of the algorithm to support this topology. It is essential that the clusterhead experiences the least possible change when it moves, so a slowly moving node as a clusterhead is chosen otherwise the cluster may be broken. For example, when a node moves quickly, it can detach from its neighbors with ensuing reduction in the node degree and moreover, it is possible that this node may move into another transmission range, i.e. another cluster. Therefore, nodes with lower mobility are favored for the role of clusterheads as there will be fewer changes in clusterheads either by replacement or re-education.

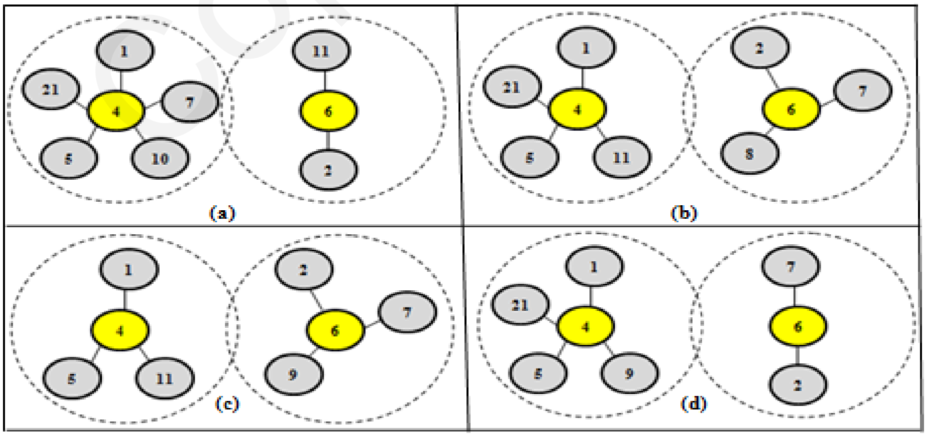


Figure 1: Neighboring nodes of node with ID = 4 and 6 during four successive Hello message

- **Distance metric (D)**

The fourth parameter is the distance between node and others within transmission range. It's better to elect a clusterhead with the nearest members. This might minimize node detachments and enhances clusters' stability. For a node i, D is computed as the cumulative mean square distance to neighbors divided by the total number of neighbors as shown in formula (3):

$$D = \frac{1}{d_i} \sum_{j \in N(i)} \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \; \ldots\ldots\ldots (3).$$

Where the neighbors of each node *i* (i.e., nodes within its transmission range) which defines its degree, d_i , as

$$d_i = |N(i)| = \sum_{j \in V, i \neq j} \{dist(i,j) \leq T_x\}$$

Where (X_i, X_j) and (Y_i, Y_j) are the coordinates of the node *i* and *j* respectively. It is assumed that this method for computing D is more efficient than the one used in ***CEMCA*** [6] where D is just the cumulative distance to neighbors. In fact, a node with a high number of neighbors close to it can have a distance superior than the one of a node with very few neighbors which is far from it.

Why choose this parameter: The motivation of D is primarily linked to energy consumption. It is understood that more power is necessary to communicate to a greater distance. Following this, it could be suggested that to use the sum of the squares (or higher exponent) of the distances would be more expedient because the increased demand on power required to support a link linearly is more severe.

- **Combined Weight (W)**

Based on the above parameters about residual battery power, average mobility, connectivity and distance, it is obvious that a node j is the best candidate for a clusterhead among all its neighbors, if its M_j is the lowest, its RBP_j is the highest, its C_j is the highest, and its D_j is the lowest. In other words, a node with the highest weight is the best candidate for a clusterhead when we combine these four metrics together as the weight, which is calculated in formula (4). But these metrics have different units, as

- The mobility can theoretically vary between zero and infinity, a normalized translation is needed. One way to do it is,
$$M \rightarrow e^{-M}$$
- The residual battery power can theoretically between 0 and max power, a normalized translation is needed. One way to do it is,
$$RBP \rightarrow \frac{RBP}{Max\_Power}$$
- The connectivity can theoretically vary between zero and *N – 1*, a normalized translation is needed. One simple way to do it is,
$$C \rightarrow \frac{C}{N-1}.$$
- The distance can theoretically vary between 0 and $(d_v * T_x)$, a normalized translation is needed. One way to do it is
$$D \rightarrow e^{-\left(\frac{D}{d_v * T_x}\right)}$$

Using the result from the above, the combined weight W_i for each node *i* is,

$$W_i = W_1 * e^{-M_i} + W_2 * \frac{RBP_i}{Max\_Power} + W_3 * \frac{C_i}{N-1} + W_4 * e^{-\left(\frac{D_i}{d_i * T_x}\right)} + \left(\frac{i}{i+1} * 0.00000001\right) \ldots.(4)$$

Where $W_1 + W_2 + W_3 + W_4 = 1$, and i is the ID number of node ($1 \leq i \leq N$).

In formula (4), the weighting factors W_1, W_2, W_3 and W_4 are set according to the different scenarios in the applications.

The last part ($\frac{1}{i+1} * 0.00000001$) of the weight definition is used to make each weight unique (i.e. no two nodes will have the same ID number). Therefore, no two nodes will have the same weight even if the value of the left part of the weight definition is the same.

The nodal block diagram Fig. 2 is based on the above description. Being a distributed system, no centralized node is involved in clustering. The decisions of each node are based on the information received from its neighbors.

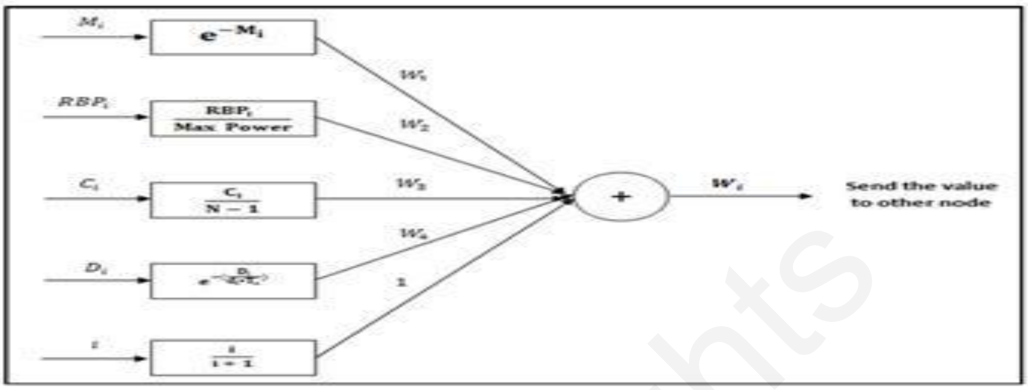


Figure 2: Block diagram for node i

4. CLUSTERING SETUP PHASE (FORMATION CLUSTERS)

Here, more detail is offered on the clustering setup phase. Just as the clustering phase has two elements so does the cluster formation itself; first, with the election of clusterhead and second, the formation of members to the clusters. The different phase steps are described below:

Phase 1: Election of the clusterhead (CH)

The position of nodes has to be located and a sequence of events to determine this for the ultimate election of the clusterhead ensues. Thus it is essential for each node to broadcast a Hello message to notify its presence to all of its neighbors in the same transmission range; a Hello message contains its ID and position value. During this phase each node compiles its neighbor list based on the receipt of Hello messages. Election of the clusterheads is based on the weight values of the neighbor nodes; each node calculates its weight value based on the metrics discussed in above and after finding its weight value, each node broadcasts the weight value using a *Weight_Info ( )* message to its one-hop neighbors only. Following the collection of *Weight_Info ( )* messages from the neighbors, each node builds a set, *S*, which contains the IDs and the weight values of itself and its neighbors. With this information the node then broadcasts its weight to its neighbors in order to compare the better among them. After this, the node that has the largest weight is chosen as a cluster head.

Phase 2: Formation of the cluster members' set

This stage is the final step of the algorithm where the construction of the cluster members' set is presented. Each clusterhead neighbor is defined at one-hop maximum and these nodes form the members of the cluster. Next, all information about its members is stored by each clusterhead, and all nodes record the cluster head identifier. This exchange of information allows the routing protocol to function both within and between the clusters.

An Illustrative Example

The clustering process is best explained with the help of figures 3-7. All numeric values, as obtained from executing CBPMD on the 10 nodes as shown in fig. 3, are tabulated in table 3. Figure 3 demonstrates the initial configuration of the nodes in the network with individual node ids; dotted circles with equal radius represent the fixed transmission range for each node. Broadcast messages from the nodes which are within its transmission range can be heard by another. Nodes sharing one-hop links in fig. 4 [nos.3 and 8, 6 and 5 and 5, 7 and 9 in the illustration] identifies these nodes as neighbors of each other and its neighbors for each node shown in fig. 5. Fig. 6 shows how a node with largest weight is selected as the clusterhead. The solid nodes [2, 3 and 7] represent the clusterheads elected for the network. Fig. 7 shows the initial clusters formed by execution of the clustering algorithm. We suppose the value of weight as the following W_1=W_2=W_3=W_4= 0.25, Max Value (Cluster Size) = 4, Min Value=1

The previous example addressed the following points:

- Weight measurement following receipt of HM broadcasts. Each mobile node starts to measure its weight after n (small integer in order to minimize memory requirement) successive ‘hello messages’.
- CH selection based on weight. When a node receives multiple messages to connect with different CH, it selects the CH with the largest weight.
- Cluster size limitations. When the clusterhead chooses its members, if the total number of neighbors exceeds the cluster size limit, CH can only select the number of members that are equal to the cluster size limit and nearest in distance.

5. Performance Analysis

A simulation model is developed using discrete event simulation to evaluate the inherent stability, reliability, and efficiency of the election clusterhead (CBPMD) algorithms. This simulator is written in NS2 to evaluate the performance of our clustering algorithm and compare them to existing clustering algorithms performances, show fig. 8 as a snapshot from our simulator.

5.1 Simulation Environment and Parameters

In our simulation experiments, three different network sizes are taken into account, 20, 40, and 60 mobile nodes, and the transmission range was varied between 20 and 100 m. Initially, each mobile node is assigned a unique node ID, a random x-y position, a random mobility speed, and a random power level greater than 95% of its maximum battery power. At every time unit, the nodes are moved randomly according to the random waypoint model in all possible directions in 200 X 200 m^2 square space with velocity distributed uniformly between 0 and maximum speed along each of the coordinates. This behavior is repeated for the duration of the simulation and each simulation scenario is run for enough time to reach and collect the desired data at the steady state. Several runs of each simulation scenario are conducted (each run representing random initial parameters) to obtain statistically confident averages. We assumed a predefined threshold for each clusterhead which can handle (i.e. cluster size) at most 10 nodes, and min value is 1. The final result is the average of 20 simulation results. The simulation parameters have been listed in table 4.

Table 3: Results from CBMD algorithm.

Node ID	Connectivity	Mobility	Distance	Residual BP	W
1	0.111	0.050	0.100	0.901	0.290
2	0.333	0.368	0.117	0.934	0.438
3	0.333	0.135	0.089	0.967	0.381
4	0.111	0.135	0.067	0.976	0.322
5	0.444	0.018	0.108	0.881	0.363
6	0.222	0.135	0.083	0.936	0.344
7	0.222	0.135	0.100	0.958	0.354
8	0.222	0.050	0.083	0.840	0.299
9	0.333	0.018	0.122	0.915	0.347
10	0.111	0.368	0.133	0.898	0.378

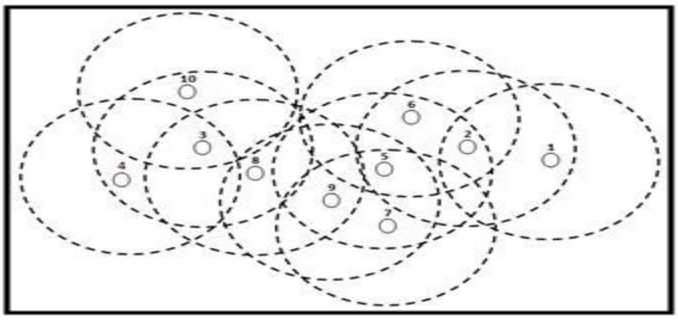


Figure 3: Initial configuration of nodes

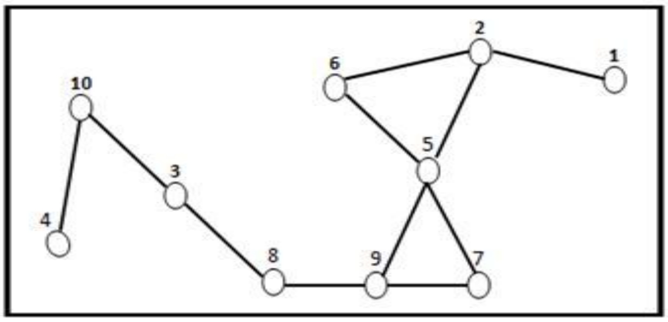
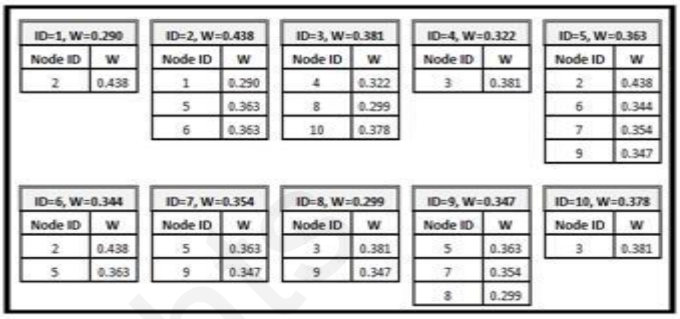


Figure 4: Neighbors determined



ID=1, W=0.290	
Node ID	W
2	0.438

ID=2, W=0.438	
Node ID	W
1	0.290
5	0.363
6	0.363

ID=3, W=0.381	
Node ID	W
4	0.322
8	0.299
10	0.378

ID=4, W=0.322	
Node ID	W
3	0.381

ID=5, W=0.363	
Node ID	W
2	0.438
6	0.344
7	0.354
9	0.347

ID=6, W=0.344	
Node ID	W
2	0.438
5	0.363

ID=7, W=0.354	
Node ID	W
5	0.363
9	0.347

ID=8, W=0.299	
Node ID	W
3	0.381
9	0.347

ID=9, W=0.347	
Node ID	W
5	0.363
7	0.354
8	0.299

ID=10, W=0.378	
Node ID	W
3	0.381

Figure 5: Identified neighborhood for each node

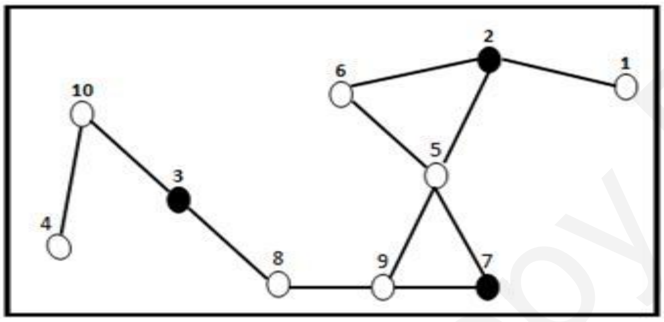


Figure 6: Clusterhead (CH) identified

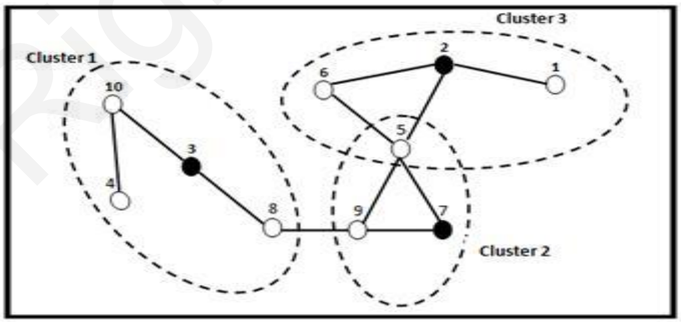
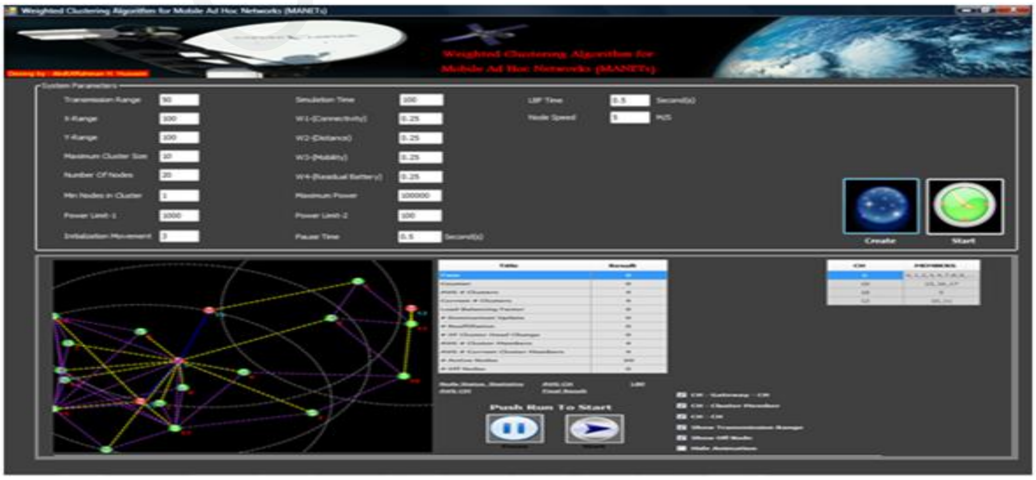


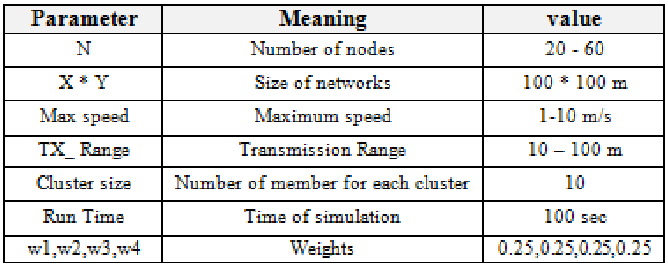
Figure 7: Clusters identified



Transmission Range 50
X-Range 100
Y-Range 100
Maximum Cluster Size 10
Number Of Nodes 20
Min Nodes in Cluster 1
Power Limit-1 1000
Initialization Movement 3
Simulation Time 100
W1-(Connectivity) 0.25
W2-(Distance) 0.25
W3-(Mobility) 0.25
W4-(Residual Battery) 0.25
Maximum Power 100000
Power Limit-2 100
Pause Time 0.5 Second(s)
LBP Time 0.5 Second(s)
Node Speed 5 M/S

Figure 8: Snapshot from simulator

Table 4: Simulation Parameter



Parameter	Meaning	value
N	Number of nodes	20 - 60
X * Y	Size of networks	100 * 100 m
Max speed	Maximum speed	1-10 m/s
TX_ Range	Transmission Range	10 – 100 m
Cluster size	Number of member for each cluster	10
Run Time	Time of simulation	100 sec
w1,w2,w3,w4	Weights	0.25,0.25,0.25,0.25

5.2 Performance Metrics

The following metrics are used to evaluate the performance of our algorithm:

- **Average number of cluster formation, also known as average number of clusterheads**: This defines the average number of logical partitions formed in the network with the mobile nodes. One of the important ways to judge the efficiency of a cluster management algorithm and give an idea about the suitability of network division will be through the number of CHs it creates.

- **Number of Reaffiliation**: Refers to the disassociation of cluster member from its clusterhead and associating itself to another cluster without affecting the corresponding clusterhead(s).One of the important ways to judge the overhead of maintenance will be through the count of reaffiliation. A higher reaffiliation count means higher control traffic overhead since all active routes to the node need to be updated. However, a lower number of reaffiliation implies a better cluster stability.

- **Number of dominant Set updates:** The dominant set updates takes place if a node is no longer associated to any of the cluster in the network (i.e. it no longer be a neighbor of any existing clusterheads). The frequent updating of dominating set incurs high overhead in updating the information. Therefore, a good clustering algorithm reduces the dominant set updates to avoid unnecessary overheads. This metric will give us an idea about the clusters stability.

<u>Results for varying transmission range</u>:

Here we show simulation results for varying transmission range (20-100 m), fixed speed 5 m/sec and varying densities (20, 40, and 60).

Fig. 9 shows the variation of the average number of clusters with respect to the transmission range. We notice a large number of clusters for small transmission range because clusters cover small area and a reduced number of clusters for large transmission range because cluster cover a very large area. However, for Tx >= 80, results in (5, 6, and 7) for different node numbers (20, 40, and 60) respectively. Thus, we satisfy the load balancing between the clusters and then a suitable division of the network. Fig. 10 shows the variation of the reaffiliation with respect to the transmission range. We notice the number of reaffiliation decrease as the transmission range increases, since the nodes, in spite of their random movement, tend to stay inside the large area covered by the clusterhead. Fig. 11 shows the number of dominant set updates with respect to the transmission range. For transmission range increases, the number of dominant set updates decreases because the nodes stay within their cluster in spite of their movements.

Results for varying speed (mobility):

Here we show simulation results for varying speed (1-10 m/s), fixed transmission range 50 m and varying densities (20, 40, 50, and 60).
Fig. 12 depicts the variation of the average number of clusters with respect node speed. With increase in speed, cluster members change clusters frequently; however, the average number of clusters shows a low variation in value. With a low variation in number of clusters, with fixed maximum number of nodes in a cluster, the network would be easily maintainable. This is because the routing tables required maintaining communication across the clusters would be easily maintainable with a lower variation of clusters with cluster size limits. We observe that the average number of cluster varies in a small interval almost [3, 5] for 20 nodes, [5, 7] for 40 nodes, [6, 8] for 50 and 60 nodes. Consequently, we can see that the number of clusters remains stable whatever the speed or the density of nodes in network. Fig. 13 shows the reaffiliation per unit time with respect to the speed. We notice that reaffiliation increase with the increase of node speed. This is because nodes with higher speed quit rapidly their cluster to reach another one. Fig. 14 shows that the numbers of dominant set update are increases as node speed increases. The number of dominated set updates which is an indicator of the overall cluster structure stability. We observe that our algorithm gives a stable number of clusters inspite of speed.

Fig. 15 compares the reaffiliation per unit time of CBPMD and CEMCA for varying speed. We observe that both algorithms give low reaffiliation rate. For low mobility CEMCA and CBPMD similar results. However, while varying node speed between 2 and 10, CBPMD gives more stable clusters. When the speed increased (2-10), our algorithm produced 27.53% and 38.7% less reaffiliation than CEMCA.

Fig. 16 shows a comparison of reaffiliation per unit time for varying transmission. We notice that for a transmission range between 80 and 100 m both algorithms CEMCA and CBPMD similar results. However, CBPMD gives lower reaffiliation than CEMCA between 30-70m and so more stable clusters. When the transmission range increased (30-70), our algorithm produced 57.15% and 63.68% less reaffiliation than CEMCA.

6. Conclusion

Motivated by the benefits and challenges of clustering in mobile ad hoc networks, this paper has presented a new clustering algorithm, called CBPMD, to select optimal clusterheads that aims to achieve stability, reliability, and low maintenance. It has the flexibility of assigning different weights and takes into account a combined metrics to form clusters automatically. We assumed a predefined limit for the number of nodes to be held by a cluster-head, so that it does not degrade the MAC function and be able to improve the load balancing in the same time.
We conducted simulation that shows the performance of the proposed clustering in terms of the average number of clusters formation, reaffiliation count, and clusterhead change. We also compared our results with the CEMCA. The simulation results show that our clustering algorithm have a better performance on average and gives a convenient decomposition of the network with balanced clusters.

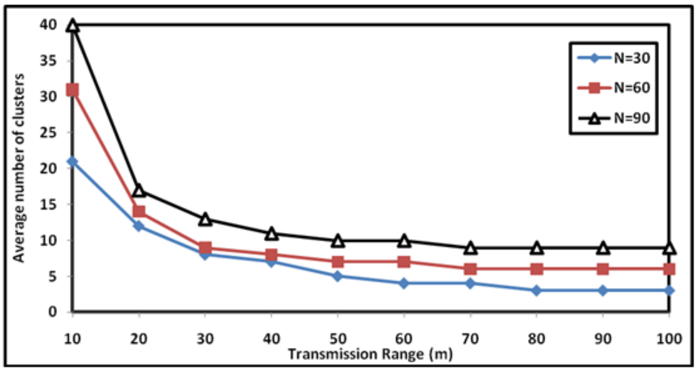


Figure 9: Impact of transmission range on the average number of clusters

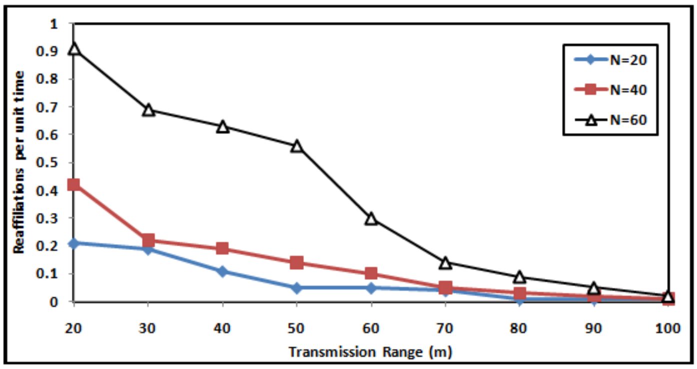


Figure 10: Impact of transmission range on reaffiliation per unit time.

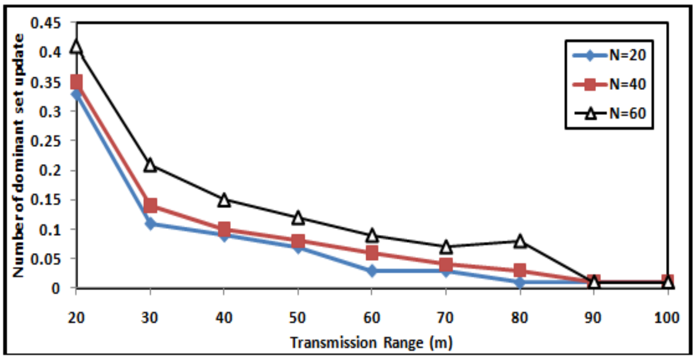


Figure 11: Impact of transmission range on number of dominant set update.

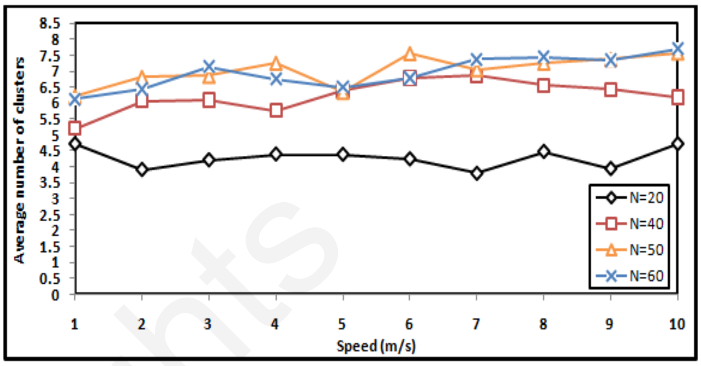


Figure 12: Impact of speed on the average number of clusters.

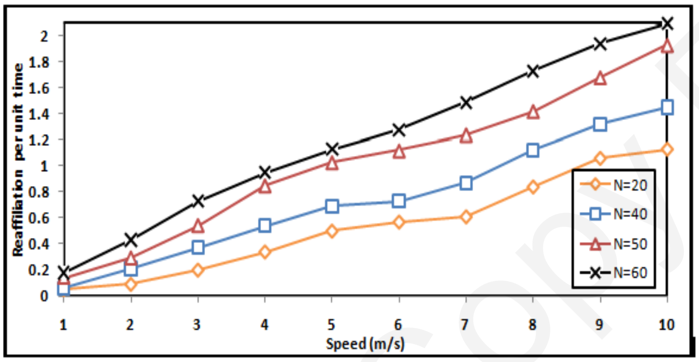


Figure 13: Impact of speed on reaffiliation per unit time.

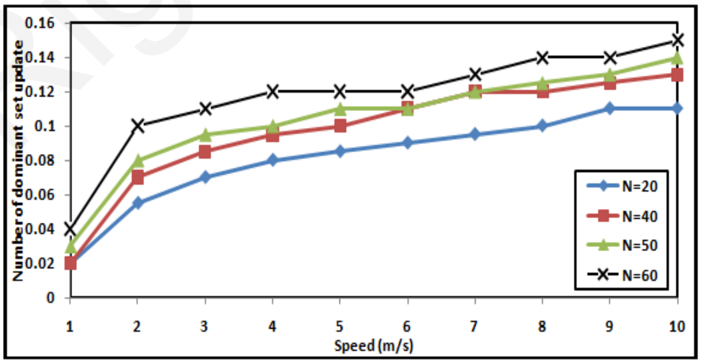


Figure 14: Impact of speed on the number of dominant set update.

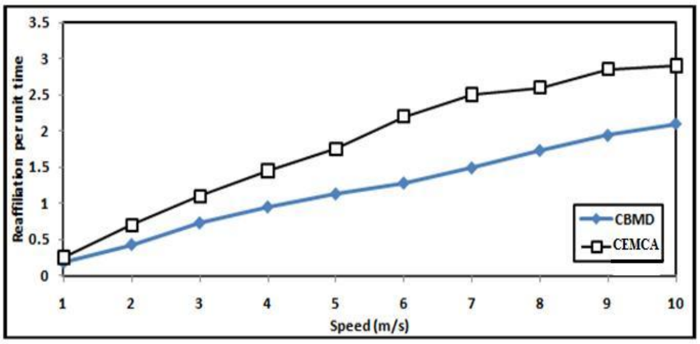


Figure 15: Comparison of reaffiliation per unit time for varying speed, N=60 and T_x=50 m

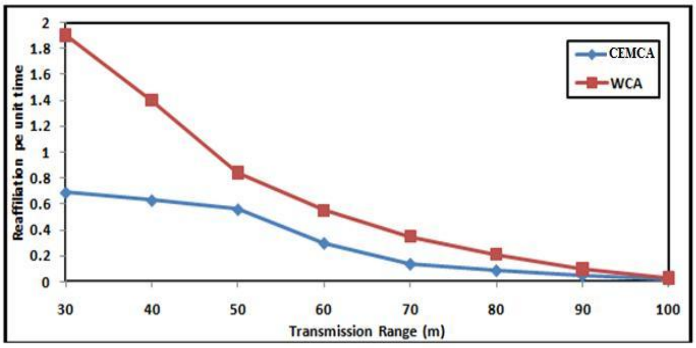


Figure 16: Comparison of reaffiliation per unit time for varying transmission range, N=60 and speed=5 m/s

7. Future Work

In the future, we intend to test CBPMD algorithm with other mobility models such as random walk model, random direction model, or reference region group Mobility model. Also, the development of an efficient routing protocol based on our clustering algorithm CMBD will be the subject of future study.

8. References

[1] Angione, G., Bellavista, P., Corradi, A., & Magistretti, E. (2007). A k-hop Clustering Protocol for Dense Mobile Ad-Hoc Networks. *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06).* Lisboa, Portugal.

[2] Agarwal, R., & Mahesh, M. (2009). Survey of clustering algorithms for MANET. Distributed, Parallel, and Cluster Computing .

[3] Chinara, S., & Rath, S. (2009). A Survey on One-Hop Clustering Algorithms in Mobile Ad Hoc Networks. Journal of Network and Systems Management , 17, Pages: 183 - 207 .

[4] El-Bazzal, Z., Kadoch, M., Agba, B., Gagnon, F., & Bennani, M. (2006). A Flexible Weight Based Clustering Algorithm in Mobile Ad hoc Networks. International Conference on Systems and Networks Communication (ICSNC'06), (Page(s): 50-56) .

[5] El-Bazzal, Z., Kadoch, M., Agba, L., Gagnon, F., & Bennani, M. (2006). An Efficient Management Algorithm for Clustering In Mobile Ad Hoc Network. International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems. Terromolinos, Spain: ACM, (Page(s): 25-31).

[6] Tolba, F., Magoni, D., & Lorenz, P. (2007). Connectivity, energy & mobility driven Weighted clustering algorithm. in proceedings of IEEE GLOBECOM 2007.

[7] Wang, X., & Bao, F. (2007). An Entropy-Based Weighted Clustering Algorithm and Its Optimization for Ad Hoc Networks. Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007), (Page(s): 56-70).

[8] Chatterjee, M., Das, S., & Turgut, D. (2002). WCA: A Weighted Clustering Algorithm (WCA) for mobile ad hoc networks. Cluster Computing (pp. 193-204). Kluwer Academic.

[9] AbdelRahman H. Hussein, Sufian Yousef, and Omar Arbeyyat(2010). “A Novel Weighted Distributed Clustering Algorithm for Mobile Ad hoc Networks”, The International Journal of ACM Jordan, Volume 1, No. 3, September 2010, ISSN: 2078-7952.

[10] Choi, W., & Woo, M. (2006). A Distributed Weighted Clustering Algorithm for Mobile Ad Hoc Networks. Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, (pp. 73-78). Guadeloupe, French Caribbean.