MPWCA-L: A New Clustering Algorithm to Improve Stability and QoS in MANETs †

Vincent Bricard-Vieu
LE2i, University of Burgundy,
Dijon, France
e-mail: vincent.bricard-vieu@free.fr

Hafssa Benaboud
LRI, Performance Evaluation Team,
Faculty of Science, Rabat, University Mohammed V-Agdal
Rabat, Morocco
e-mail: benaboud@fsr.ac.ma

Noufissa Mikou
LE2i, University of Burgundy,
Dijon, France
e-mail: noufissa.mikou@u-bourgogne.fr

***Abstract*— Mobile Ad hoc Networks (MANETs) have a dynamic topology due to the mobility of their nodes. This mobility makes routing more complicated. Moreover, the well known routing protocols are not able to offer QoS. To manage MANETs, clustering techniques can be used. The principle of clustering is to organize the network into a hierarchical structure. This last makes the hierarchical routing and optimizes bandwidth by minimizing the amount of information exchanged to maintain routing tables. Clustering techniques have several advantages. However, a reconfiguration of the system is inevitable because the association and dissociation of nodes, to and from clusters, perturb the stability of the network topology. In order to keep the topology stable as long as possible, cluster-heads are used and elected. These nodes form a dominant set and determine the topology and its stability. In this paper, we give an overview of proposed clustering algorithms up to this date and we present three clustering algorithms, MPWCA, WCA-L and MPWCA-L, that we have proposed in previous works. These algorithms are based on an on-demand distributed clustering algorithm for MANETs. Simulations are performed to evaluate the stability of the dominant set and the QoS in terms of packet delivery rate and overhead provided by our algorithms. We compare the results of the proposed approaches with the well-known clustering technique, WCA, using different performance metrics. Results show that our algorithms, and especially MPWCA-L, perform better than WCA.**

Keywords – MANET, Clustering Algorithms, WCA, QoS

† This paper is a revised and expanded version of paper [18] presented at the conference mentioned in references.

I. INTRODUCTION

Multi-hops wireless networks are mobile radio networks without infrastructure, which allows rapid implementation. They can also be connected to a LAN to extend the coverage of existing infrastructure. Entities may appear, disappear, and move independently of each other. The network topology is scalable. The terminals can communicate within the scope of their radio communication. A multi-hop communication scheme is necessary to allow communication of two remote correspondents. In this scheme, each terminal can be used as a router to relay communications from other terminals. The configuration of these multi-hop routes is carried out by a routing protocol. To be effective, these routing protocols should consider the intrinsic characteristics of the network (topology changing), terminals (memory size and limited computing capabilities...), and communication medium (limited bandwidth, interference...).

Today there are many routing protocols for such networks. While these protocols are effective on sparse networks or those having small or medium size, none of them can be used on large scales because they generate too much traffic control or require large routing tables. To solve the problem of large scale routing, the proposed solutions have introduced a hierarchical routing by grouping geographically close entities in clusters. The hierarchical routing uses different routing schemes within clusters and between clusters. This approach allows each entity to store all the information of its cluster and only a part of the information of other clusters, and thus, enables network scalability.

A multi-cluster wireless network should be able to dynamically adapt itself when network's configuration changes. Some nodes, known as cluster-heads, are responsible for the formation of clusters. The set of cluster-heads is called *Dominant set*. A cluster-head is responsible of resource allocation for all nodes in its cluster.

DOI 10.5013/IJSSST.a.14.05.01 ISSN: 1473-804x online, 1473-8031 print

Due to the dynamic nature of the mobile nodes, their association and dissociation, to and from clusters, perturb the stability of the network and thus, reconfiguration of cluster-heads is often unavoidable. Various clustering techniques have been proposed. Most of these methods induce a very significant overhead due to signalling messages allowing nodes to identify them as well as messages exchanged during reconstruction of the hierarchy. None of these techniques has used a mechanism for estimating path nodes to limit the impact of this overhead.

In this paper, we propose various algorithms to improve clustering techniques. Each one improves significantly one of the clustering phases (construction of the hierarchy or maintenance). The first approach decreases the overhead (caused by out-band signalling cluster-head) by reducing the frequency of signalling packets and by replacing dropped packets by the path estimation of the concerned cluster-head, using its past positions. In the second approach, we limit the cost of total reconstruction of the hierarchy.

The remainder of this paper is organized as follows. Section II recalls the principles of the clustering algorithms and discusses some related works. In section III, we give a brief description of the well-known clustering technique WCA. Section IV describes our algorithm MPWCA-L which is a combination of our two proposed algorithms MPWCA and WCA-L. Section V presents performances evaluation of these algorithms and those of WCA and gives a comparison, using simulation of a several systems with various numbers of nodes. Section VI concludes our study.

II. CLUSTERING ALGORITHMS AND RELATED WORKS

A. *Mobile Adhoc Clustering Algorithms Principle*

Clustering is to divide the network into groups of entities called clusters, giving the network a hierarchical structure (Fig. 1). Each cluster is represented by a special node called cluster-head. This node is elected as a cluster-head according to a specific metric or combination of metrics such as user name, degree, mobility, weight, density and others. The cluster-head acts as a local coordinator in the cluster.

Cluster-heads can be used to:

- Control the scheduling of the channel,
- Control energy consumption,
- Maintain frame synchronization,
- Improve reuse of code and time,
- Serve as a regional broadcaster.

A clustering algorithm is based on the following steps:

Construction of the hierarchy: In this phase, the cluster-heads are appointed (elected) and they relate with ordinary nodes located in their neighbourhood in order to form clusters. Election phase uses heuristics like the largest/ lowest ID in the neighbourhood, the greater degree of connectivity (i.e. the largest number of neighbours), the geographic area, the transmit power or the moving speed (e.g. use the less mobile nodes), or by using a weight for each node representing a different combination of criteria.

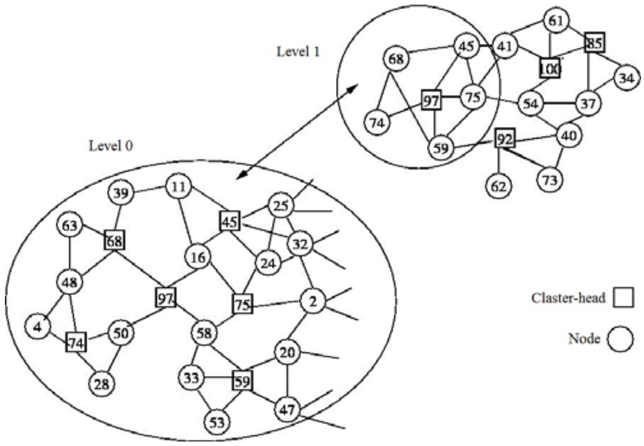

Figure 1. Clustering principle in Ad Hoc networks

Communication between cluster-heads: In a cluster, each ordinary node is two hops away with its neighbours. Moreover, the cluster-heads are not directly connected, so gateways nodes are also elected and used for communications between cluster-heads, unless those using two different transmission powers: a small for intra cluster communications, and a great for communication with other cluster-heads.

Maintenance of cluster-heads: in order to adapt to frequent changes in network topology, the elected cluster-heads and ordinary nodes attached to them are dynamically updated.

B. *Related Work*

In mobile ad hoc networks, LCA (Linked Cluster Architecture) [1], [2], [4] is an early work on clustering. Gerla and his team have developed several protocols for clustering [6], [7], [8], [13]. In these approaches, the clustering is to classify the nodes of the network in a hierarchical manner in equivalent classes using different criteria (e.g. address, geographic area).

Several heuristics have been proposed to choose cluster-heads in ad hoc networks. Authors in [14] cited the well known algorithms and presented their limitations in section 2 of their paper. They include: Highest-Degree heuristic [5], [6], Lowest-ID heuristic [1], [2], [4], and Node-Weight heuristic [11], [12]. The Lowest-ID and the Highest-Degree were the two clustering algorithms which were based on the LCA.

Authors in [15] presented a comprehensive survey of proposed clustering algorithms up to 2005. Authors classified them based on their objectives in table 2 of their paper. We recall here the table summarizing the six clustering schemes that we find interesting for readers.

DS-based clustering: Finding a (weakly) connected dominating set to reduce the number of nodes participating in route search or routing table maintenance.
Low-maintenance clustering: Providing a cluster infrastructure for upper layer applications with minimized clustering-related maintenance cost.
Mobility-aware clustering: Utilizing mobile nodes' mobility behaviour for cluster construction and maintenance and assigning mobile nodes with low relative speed to the same cluster to tighten the connection in such a cluster.
Energy-efficient clustering: Avoiding unnecessary energy consumption or balancing energy consumption for mobile nodes in order to prolong the lifetime of mobile terminals and a network.
Load-balancing clustering: Distributing the workload of a network more evenly into clusters by limiting the number of mobile nodes in each cluster in a defined range.
Combined-metrics-based clustering: Considering multiple metrics in cluster configuration, including node degree, mobility, battery energy, cluster size, etc., and adjusting their weighting factors for different application scenarios.

Discussion of these clustering schemes is given in [15]. We are interested in Combined-metrics-based clustering. The well known algorithm in this category is WCA which we present in the following section.

III. WEIGHTED CLUSTERING ALGORITHM

The Weighted Clustering Algorithm (WCA) [14] works differently of most of algorithms proposed in literature. It is only invoked on demand by isolated nodes. Moreover, to determine the cluster-head nodes, WCA considers several parameters: the ideal number of nodes that a cluster can handle, the mobility (speed of nodes), the distance between a node and its neighbours and the battery power. WCA assigns weights to these different parameters.

The cluster-head election procedure is invoked at system activation, and also when the current *dominant set* is unable to cover all the nodes. Every invocation of the election algorithm does not necessarily mean that all the cluster-heads in the previous *dominant set* are replaced by new ones. If a node detaches itself from its current cluster-head and links itself to another one, then involved cluster-heads update their member list instead of invoking the election algorithm. A detailed description of cluster-heads election in WCA is given in [14].

All nodes monitor continuously the signal strength of a *Hello* messages received from the cluster-head. When the node moves away from its cluster-head, the signal strength decreases. In this case, the mobile has to notify its current cluster-head that it is no longer able to attach itself to it. The node tries to handover to a neighboring cluster which cluster-head is the first found in its list. If the node goes into a region not covered by any cluster-head, then the WCA election procedure is invoked and a new dominant set is obtained.

Unfortunately, a high communication overhead is induced by periodic *hello* messages, used for control and the broadcasted messages during elections. We proposed our algorithms in order to avoid this overhead. The following sections present these algorithms and compare their performance to those of WCA.

IV. MPWCA-L DESCRIPTION

Mobility **P**rediction-based **W**eighted **C**lustering **A**lgorithm using **L**ocal cluster-heads election, MPWCA-L, is a combination of our two algorithms MPWCA and WCA-L. We have shown in previous works ([16] and [17]) that MPWCA and WCA-L improve the stability of the dominant set and the quality of service. Fortunately, these two algorithms act at different levels in the clustering scheme and thus, they are independent. We can then combine both of them to form a hybrid algorithm, MPWCA-L, which could improve more and more performances in comparison to those of the WCA.

To give a description of MPWCA-L, it is wise to recall the two algorithms on which it is based. In the following, a brief description of MPWCA and WCA-L is given.

A. MPWCA Description

As mentioned in section III, the overhead induced by WCA control messages is very high, since it uses a large part of bandwidth for building and maintaining the dominant set (discovery of neighbours, election process, signal strength monitoring), which cannot be used for useful data transmissions.

To reduce this overhead, we can reduce the number of hello messages by increasing the duration between them. However, due to nodes mobility, the topology is always changing and ordinary nodes might miss information about the position of their cluster-head. This leads to a link failure, and consequently to an inadvertent reconstitution of the hierarchy that could have been avoided if the isolated node was able to handover a few moments earlier.

To limit the impact of this lack of information about the position of their cluster-head, ordinary nodes try to estimate the position using the past trajectory of the cluster-head. So, we propose a distributed mobility prediction-based mechanism using the past movements of the cluster-heads to replace the missing information given by frequent Hello Messages. A detailed description of the MPWCA is already given in [16]. For the convenience, we recall here its operation.

Our estimation algorithm starts after the election of the cluster-heads, when the ordinary nodes are monitoring the signal strength of packets from their cluster-head. It works as follow:

Step 1.
The cluster-head periodically sends its position, in Cartesian coordinates (x, y, z), and its speed in Hello messages. When an ordinary node receives the Hello message, it stores information about its cluster-head into a list named past information list.

Step 2.

If an ordinary node has less than two elements in its past information list in time interval between two Hello messages, it waits for the next hello message (step1). Otherwise, it uses the past information list to estimate the current position and speed of its cluster-head and to store it in a list named prediction list. Time interval between two Hello Messages could be very large. So, the ordinary node could estimate position and speed several times. In this case, the previous information will be stored as past information list, and the new estimation will be stored in the prediction list. The computing of an estimated position is given in [16].

Step 3.
How does the ordinary node decide if it should stay or not in its current cluster ?

- The ordinary node computes the distance to the position of its cluster-head. If it has just received a Hello message, the position of the cluster-head is a trusted position. Otherwise, the position of the cluster-head is an estimated one.
- The ordinary node compares this distance to the transmission range, which is the same for all nodes. If the distance is less than transmission range, the ordinary stays in its cluster. Otherwise, it tries to handover to another neighbouring cluster-head.
- If the ordinary node cannot find another cluster-head in its neighbourhood, two cases are discussed:
  (a) The position of the cluster-head is a trusted position, given by a Hello message. In this case, the ordinary node is isolated and we must invoke the election algorithm to obtain a new dominant set.
  (b) The position of the cluster-head is an estimated position. We should update the dominant set by invoking the election algorithm, but it induces a high computational cost and a high overhead. Assuming that we might have false estimations, we prefer avoid unneeded updates of the dominant set. So, the ordinary node stays in its current cluster, waiting for the next estimation or the next Hello message.

Step 4.
Each time the ordinary node receives a new Hello Message from its cluster-head (step 1), it clears its prediction list. The past information list has a finite size (in our experiments, we choose to keep at most 10 positions). When this list is full, the oldest information is deleted in order to insert the new information. When a new election occurs, both the past information list and the prediction list are cleared.

B. WCA-L Description

As shown in [16], our algorithm MPWCA performs better than WCA. However, the improvement did not correspond to what we expected. In fact, MPWCA only acts on the maintenance phase and we observed that such phase represents only a third of the considered duration. This motives us to propose additional improvement that potentially acts on the election phase. This improvement gave birth to our algorithm which we named WCA-L (for WCA with Local cluster-heads election) [17].

In WCA and MPWCA, election occurs when an ordinary node is not able anymore to use its cluster-head as a forwarder and its handover fails. In WCA-L, however, we avoid this election by considering that an isolated node becomes itself cluster-head and forms its own cluster.

Nevertheless, such behaviour could lead to a network with members which are all cluster-heads. To avoid this situation, we propose to invoke election when at least two cluster-heads are one-hop neighbours as shown on Fig. 2. In this case, the cluster-heads in competition inform ordinary nodes belonging to their respective clusters that election will start.

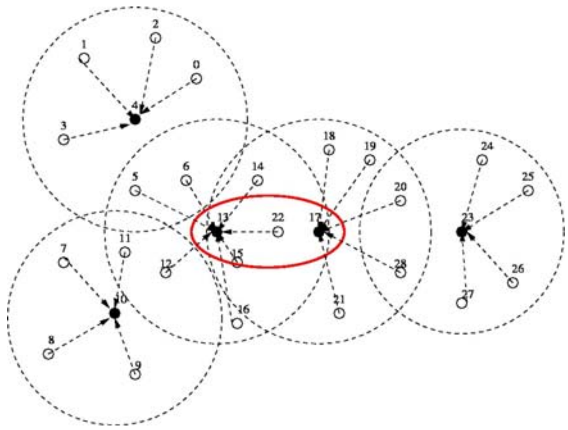

Figure 2. Two cluster-heads are neighbors

Then, as shown in Fig. 3, ordinary nodes try to handover to another cluster. Note that only gateways are successful. Finally, only the remaining ordinary nodes and the concurrent cluster-heads continue with the election process which is the same as the one described in WCA except that the election in WCA-L is localized to concurrent clusters (see Fig. 4) and does not affect the entire network.

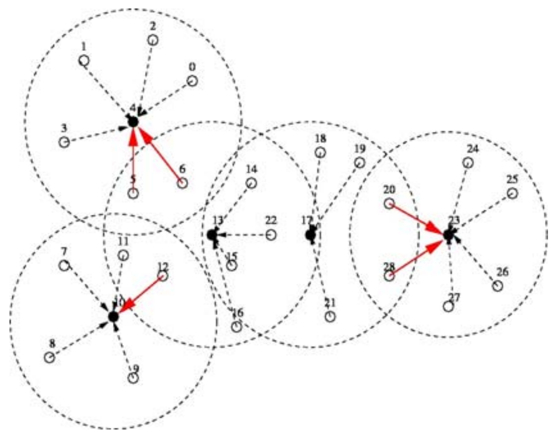

Figure 3. Ordinary nodes try to handover

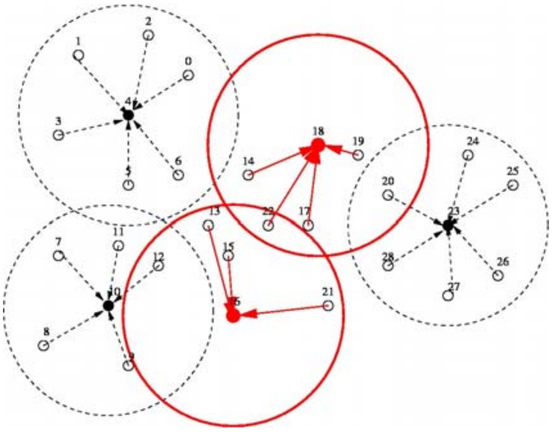

Figure 4. Election process is localized to nodes belonging to the concurrent clusters

V. Random Modeling of Ad Hoc Networks

The cluster-head and the node are mobile (random trajectories). The cluster-head sends at regular intervals (e.g. every second) a hello message which allows the node to verify if it is still in the transmission range of its cluster-head. If not, (i.e. if the node moves to another transmission range), then this leads to a link break. At this time, the node will try to perform a handover. Assume that we know the cluster-head position and the node position at reception time of the n^{th} hello message. Could we have information on their positions at the reception moment of the $(n+1)^{th}$ hello message?

In effect, if we can have these expectations, we will know in advance if such node will no longer be within the transmission range of its cluster-head, and thus, we will take measures to avoid the link break.

Suppose that the messages Hello arrive according to a Poisson process with parameter λ.

Denote by $X_n = (x_n^1, x_n^2)$ (resp. $Y_n = (y_n^1, y_n^2)$) the position of the cluster-head (resp. the position of the ordinary node) at hello message reception time.

At the node level, we have the following information when receiving the hello message: ordinary node position, cluster-head position, ordinary node speed and cluster-head speed as well as the state of connection with the cluster-head (expressed in the received signal power).

X_n and Y_n are random processes, with countable sets of indices and continuous state spaces.

Since the hello messages arrive according to a Poisson process, the times between X_{n+1} and X_n, as well as between Y_{n+1} and Y_n, are independent and exponentially distributed. The cluster-head position (resp. ordinary node position) at the arrival time of the hello message $n+1$ depends only on its position at the arrival time of the hello message n and not of previous positions.

Indeed, let $P_{n,n+1}$ (resp. $P'_{n,n+1}$) the movement vector of the cluster-head (resp. of an ordinary node):

$$\begin{aligned} P_{n,n+1} &= (x_{n+1}^1 - x_n^1,\, x_{n+1}^2 - x_n^2), \\ P'_{n,n+1} &= (y_{n+1}^1 - y_n^1,\, y_{n+1}^2 - y_n^2). \end{aligned} \tag{1}$$

We have to estimate the position P_{n+1} versus P_n, P_{n+1} being obtained by translating the position P_n of the movement vector.

$$P_{n,n+1} = \frac{1}{n-1}\sum_{i=1}^{n-1} P_{i,i+1} \tag{2}$$

where $P_{i,i+1} = (x_{i+1}^1 - x_i^1 ; x_{i+1}^2 - x_i^2)\, for\ i = 1..(N-1)$

Since $P_{i;i+1}$ are vectors, we have also

$$P_{1,n+1} = \sum_{i=1}^{n-1} P_{i,i+1} \tag{3}$$

and then
$$P_{n,n+1} = \frac{1}{n-1} P_{1,n+1} \tag{4}$$

We therefore obtain:

$$\begin{cases} X_{n+1} = \left[x_n^1 + \dfrac{x_n^1 - x_1^1}{n-1},\, x_n^2 + \dfrac{x_n^2 - x_1^2}{n-1} \right] \\ Y_{n+1} = \left[y_n^1 + \dfrac{y_n^1 - y_1^1}{n-1},\, y_n^2 + \dfrac{y_n^2 - y_1^2}{n-1} \right] \end{cases}, n \geq 2 \tag{5}$$

$\{X_n\}$ et $\{Y_n\}$ are then discrete time Markov process with continuous state space.

A. Distance between the node and the cluster-head

We consider a two-dimensional space and we introduce some metric that calculates the distance between the node and the cluster-head:

$d(X_n, Y_n) \leq d_{\max}$ means that the link is not broken;

$d(X_n, Y_n)$ and $d_{\max}$ are known. We have:

$$d(X_n, Y_n) = \sqrt{(y_n^1 - x_n^1)^2 + (y_n^2 - x_n^2)^2} \tag{6}$$

Suppose that the position of cluster-head (X_n) and the one of node (Y_n) represent the centres of the circles and that X_{n+1} and Y_{n+1} are on respective circles (Fig. 5). Denote by:

- W_n inter-arrival between hello messages $n+1$ and n.
- V^h (resp. V^e) speed of cluster-head (resp. speed of node). Suppose these speeds are constants.
- θ_n^h (resp. θ_n^e) angle determined by the direction of the cluster-head (resp. of the node) and the axe Ox, as shown in Fig. 5.

So,

$$x_{n+1}^1 = x_n^1 + V^h\, W_n \cos\theta_n^h,$$
$$x_{n+1}^2 = x_n^2 + V^h\, W_n \sin\theta_n^h \qquad (7)$$

And also

$$y_{n+1}^1 = y_n^1 + V^e\, W_n \cos\theta_n^e,$$
$$y_{n+1}^2 = y_n^2 + V^e\, W_n \sin\theta_n^e \qquad (8)$$

We compute $P(d(X_{n+1}, Y_{n+1}) > d_{\max})$ the probability of link break. We consider the worst case, where the ordinary node and its cluster-head move in directions which are diametrically opposite as shown in Fig. 5.

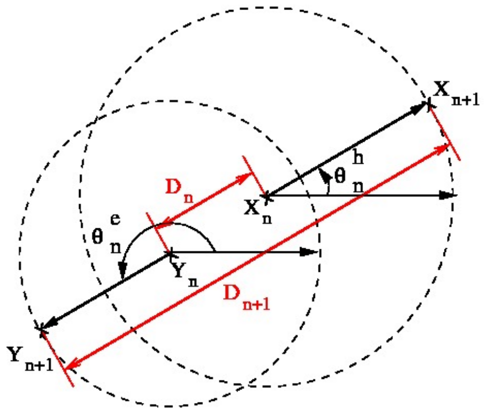

Figure 5. Distance between an ordinary node at position Y_{n+1} and its cluster-head at position X_{n+1} upon reception of the *(n+1)*th hello message

Upon arrival of the $(n+1)^{th}$ hello message, the distance between the cluster-head and the node is the largest (denoted by D_{n+1}), namely:

$$D_{n+1} = d(X_n, Y_n) + radius\ of\ the\ first\ circle$$
$$+ radius\ of\ the\ \mathrm{sec}ond\ circle$$
$$+ d(X_n, Y_n) + V^h.W_n + V^e.W_n$$

We have:

$$P(D_{n+1} > d_{\max}) = P\left(W_n > \frac{d_{\max} - d(X_n, Y_n)}{V^h + V^e}\right) \qquad (9)$$

Let

$$\alpha = \frac{d_{\max} - d(X_n, Y_n)}{V^h + V^e}$$

$$P(W_n > \alpha) = \exp(-\alpha\lambda) \qquad (10)$$

We want to find an optimal value for λ where probability of link failure is very small. For that, we need to calculate the transition probabilities of Markov chains $\{X_n\}$ and $\{Y_n\}$.

B. Transitions Probabilities

Let $X = (x^1, x^2)$ and $Y = (y^1, y^2)$ two points in the plane, transition probability of the cluster-head positions is given as follows:

$$P(X_{n+1} = X / X_n = Y) = \begin{cases} \dfrac{1}{2\pi\sqrt{\dfrac{(x^1-y^1)^2+(x^2-y^2)^2}{(V^h)^2}}} \\ \times\, \lambda e^{-\lambda\sqrt{\frac{(x^1-y^1)^2+(x^2-y^2)^2}{(V^h)^2}}} \end{cases} \qquad (11)$$

We can easily show that:

$$\int_{x^1}\int_{x^2} P(X_{n+1} = X / X_n = Y) = 1 \qquad (12)$$

Similarly, we can calculate the transition probability of ordinary nodes positions that gives the same result as in (11) by changing V^h by V^e.

$$P(Y_{n+1} = X / Y_n = Y) = \begin{cases} \dfrac{1}{2\pi\sqrt{\dfrac{(x^1-y^1)^2+(x^2-y^2)^2}{(V^e)^2}}} \\ \times\, \lambda e^{-\lambda\sqrt{\frac{(x^1-y^1)^2+(x^2-y^2)^2}{(V^e)^2}}} \end{cases} \qquad (13)$$

It is clear that the necessary condition for ergodicity is checked (irreducibility and aperiodicity). In fact, $\forall\ X = (x^1, x^2),\quad \forall\ Y = (y^1, y^2)$
$P(X_{n+1} = X / X_n = Y) > 0$ and $P(Y_{n+1} = X / Y_n = Y) > 0$.

C. Steady State

The process $\{X_n\}$ and $\{Y_n\}$ are random walks (in discrete time and continuous state space). There is no well-established approach; the calculation of steady state is extremely complex. However, probability theory offers a possibility of using the factorial methods. The studied random variable can be divided into two parts, one is exponential (in our case, it is W_n, the time between the *(n+1)*th and the n^{th} hello messages).

To measure W_n, we can use the duration connectivity between a cluster-head and a node (between *(n+1)*th and the n^{th} hello messages) which we note by D_c.

Compute of Dc:

Assume that the node and its cluster-head are within the same transmission range upon reception of the n^{th} hello message. We look for the time *t* during which they remain in this transmission range, which means that the distance between the ordinary node and its cluster-head at time *t* is less than *R* (in this case, at $t = 0$ they were in the position where the n^{th} hello message found them). This amounts to solve the following equation:

$$\begin{pmatrix} \left(x_n^1 + V^e t \cos\theta_n^e - (y_n^1 + V^h t \cos\theta_n^h)\right)^2 \\ + \left(x_n^2 + V^e t \sin\theta_n^e - (y_n^2 + V^h t \sin\theta_n^h)\right)^2 \end{pmatrix} < R^2$$

$$\Leftrightarrow$$

$$\begin{pmatrix} \left(x_n^1 - y_n^1 + t(V^e \cos\theta_n^e - V^h \cos\theta_n^h)\right)^2 \\ + \left(x_n^2 - y_n^2 + t(V^e \sin\theta_n^e - V^h \sin\theta_n^h)\right)^2 \end{pmatrix} < R^2$$

We have,

$$(b+at)^2 + (d+ct)^2 < R^2$$

$$\Leftrightarrow (a^2+c^2)t^2 + 2(ab+cd)t + (b^2+d^2) - R^2 < 0$$

where,

$$\begin{cases} a = V^e \cos\theta_n^e - V^h \cos\theta_n^h \\ b = x_n^1 - y_n^1 \\ c = V^e \sin\theta_n^e - V^h \sin\theta_n^h \\ d = x_n^2 - y_n^2 \\ R = Transmission\ radius\ of\ the\ cluster - head \end{cases}$$

We solve the following trinomial,

$$(a^2+c^2)t^2 + 2(ab+cd)t + (b^2+d^2) - R^2 = 0$$

Simply, we use the determinant

$$\Delta = 4\,[(a^2+c^2)R^2 - (ad-bc)^2]$$

As ordinary node is within transmission range of its cluster-head, we additionally have $b^2 + d^2 - R^2 < 0$. Then $\Delta > 0$ and solutions are:

$$t_1 = \frac{-2(ab+cd) + 2\sqrt{(a^2+c^2)R^2 - (ad-bc)^2}}{2(a^2+c^2)}$$

$$t_2 = \frac{-2(ab+cd) - 2\sqrt{(a^2+c^2)R^2 - (ad-bc)^2}}{2(a^2+c^2)}$$

$\frac{b^2+d^2-R^2}{a^2+c^2} < 0$, we conclude that one of them is positive.

We have then,

$$D_c = \frac{-(ab+cd) + \sqrt{(a^2+c^2)R^2 - (ad-cb)^2}}{a^2+c^2} \qquad (14)$$

The compute of stationary probability of cluster-head position and ordinary node position will allow computing the duration of connectivity distribution and also, to obtain the optimum value of the average inter-arrival Hello messages, λ, which minimizes the probability of link failure (10).

In following section, we give a simulation study and we discuss our results. We will show that our algorithms (MPWCA,WCA-L and MPWCA-L) performs better than WCA in terms of number of updates of a dominant set, number of handovers of a node in a cluster, packet delivery rate and overhead.

VI. PERFORMANCE EVALUATION: SIMULATION

A. Simulation Model

For our simulations, we use GloMoSim [9]. GloMoSim is a discrete event parallel environment based on PARSEC (PARallel Simulation Environment for Complex systems) [10]. We have developed some models using PARSEC language and integrated them in GloMosim.

We simulate three systems of 25, 50 and 75 nodes respectively on a (500m x 500m) area. Clustering study leads us to consider an optimal number of neighbours so that clusters are neither overloaded nor under-loaded. In the literature, this value is chosen in an arbitrary manner. In our work, using [3], we find an optimal value and derive the optimal transmission range versus the number of nodes and the system area. We obtain for 25 nodes (resp. 50 and 75) 130m (resp. 92m and 75m).

The nodes can randomly move in all possible directions with speed varying uniformly between 0 and one parameter representing the maximum value of the speed.

To measure the packet delivery rate we consider constant bit rate data traffic, with a packet inter-arrival time of 100ms. Data packet length is 512 kbytes.

When all cluster-heads are chosen, they start sending hello messages with a period of 3s. Then, ordinary nodes start predictions (**step 2** to **step 3** described in section IV). In our experiments, we suppose that two predictions are made before receiving the next hello message. After that, the prediction list is cleared and the algorithm is then in **step 1**.

B. Results and Analysis

To measure the performance of our system, we consider four metrics:

- number of updates of the dominant set,
- number of handovers between two clusters,
- average packet delivery rate,
- overhead

These metrics are studied as a function of the maximum speed of the nodes.

In our simulations, we choose values 1 m/s, 5 m/s, 10 m/s, 25 m/s and 40 m/s for the maximum speed of nodes. For the estimation, we choose to replace 2 Hello messages by 2 estimations. The nodes move randomly and uniformly in all possible directions.

From Fig. 6, we can see that the number of handovers increases while the speed increases. We can also observe that MPWCA, WCA-L and MPWCA-L allow a lower number of handovers than WCA.

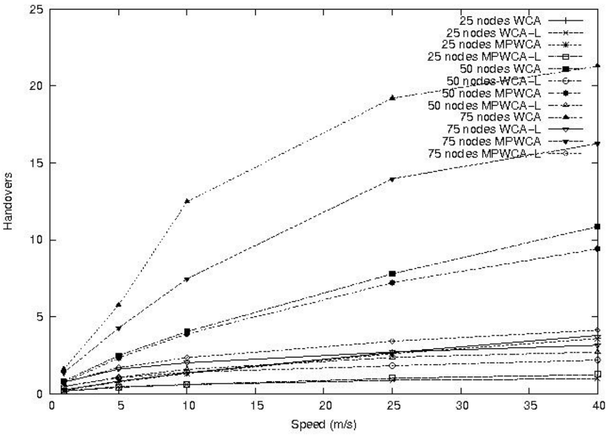

Figure 6. Number of handovers vs maximum speed

Due to the mobility, nodes do not always stay in the same cluster but we can notice a very interesting result: using WCA-L and MPWCA-L, the number of handovers increases slower than WCA. This means that our algorithms make the network more stable than WCA because the speed has a lower impact on the performance with WCA-L and MPWCA-L than with WCA.

Like handover, the number of updates of the dominant set increases while the speed increases, due to mobility, as shown on Fig. 7. We can also see that our algorithms give better results for this metric, since WCA involves more updates of the dominant set than our algorithm, and the cost of these updates is high in terms of resources allocation such as CPU and bandwidth.

However, we can see that WCA-L and MPWCA-L outperform WCA for low density (25 nodes). This can be explained by a failure of the mobility model used for simulation. As explained in [11], the Random Waypoint tends to concentrate nodes in the center of simulation area. Thus, cluster-heads are more often neighbours in WCA-L and MPWCA-L than ordinary nodes are isolated in WCA and, considering the optimal transmission range for 25 nodes, there are more local election in WCA-L and MPWCA-L than global election in WCA.

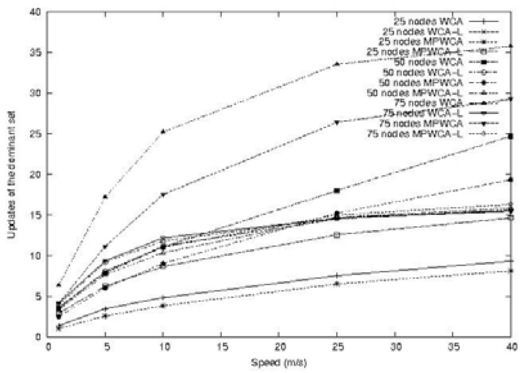

Figure 7. Number of updates of the dominant set vs maximum speed

Fig. 8 shows that the average packet delivery rate decreases when the speed increases. When the speed increases, the links between two nodes more often break, then there are more packet losses and thus, fewer packets delivered to the destination.

We also observe that results in terms of packet delivery rate are similar for 50 nodes and our algorithms give better results for 75 nodes, but WCA-L and MPWCA-L outperform WCA for 25 nodes.

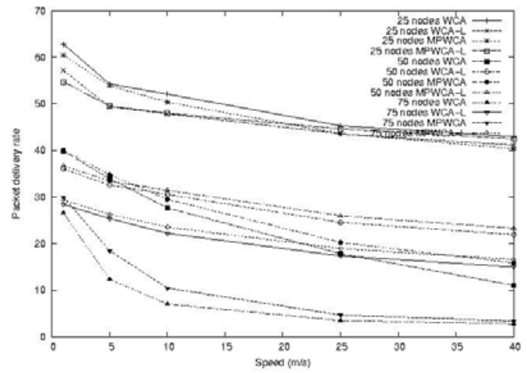

Figure 8. Number of updates of the dominant set vs maximum speed

To explain these results we should take into account that a local election concerns a bigger proportion of nodes in 25 nodes than in 75 nodes. Considering that each cluster-head has in average 5 neighbours, if we suppose that 2 cluster-heads are concurrent, a local election will concern about 10 nodes. This value represents 40% of 25 nodes and only 13% of 75 nodes. So a local election has a high cost for small networks, and this cost tends to be similar to the cost of a global election. Hence, since we have a higher number of updates of the dominant set in the case of 25 nodes, due to a failure of the mobility model, this degrades the average packet delivery rate in WCA-L and MPWCA-L, in comparison with WCA.

For 75 nodes, the low delivery rate for speeds greater than 10 m/s can be explained by the fact that there are a higher number of updates of the dominant set, thus involving a frequent reconstruction of the routes for each destination. Note that the average delivery decreases slower in WCA-L and MPWCA-L than in WCA which is a very important characteristic because we want to keep stability while speed is increasing.

Since we replace Hello messages by predictions, we reduce the overhead induced by these signaling packets so the unused bandwidth can be allocated to data packets, thus increasing the packet delivery rate. Once again, we can observe that the delivery rate decreases slower with WCA-L and MPWCA-L than with WCA.

Fig. 9 show the overhead percentage, considering the number of control packets divided by the total number of packets sent. We can see that overhead increases while speed increases but the increase speed in MPWCA-L is lower than those in WCA. Except for 25 nodes, we observe that our algorithms perform better than WCA.

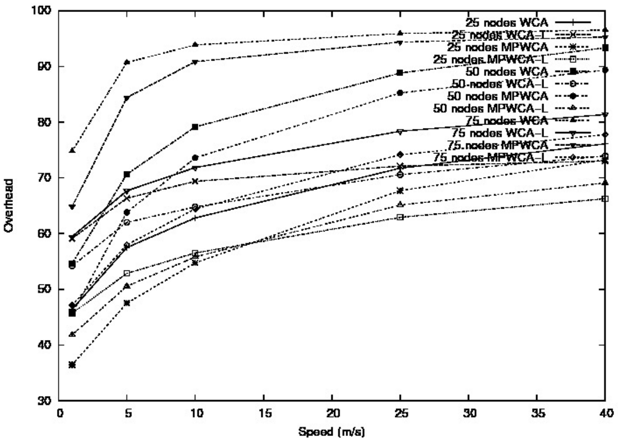

Figure 9. Number of updates of the dominant set vs maximum speed

We can resume our results considering the gain of our algorithms versus WCA in the worst case (25 nodes) and in the best case (75 nodes) in the table below:

Metric	MPWCA		WCA-L		MPWCA-L	
	Worst case	Best case	Worst case	Best case	Worst case	Best case
Updates of the dominant set (%)	14	58	-20	23	-52	152
Handovers (%)	3	67	67	38	22	463
Average delivery rate (%)	-9	98	-13	38	-13	541
Overhead (%)	3	4	3	26	2	16

Regarding these gains, we can observe that the combination of mobility prediction and local election (MPWCA-L) highly improves the performances in comparison to WCA, especially for a high number of nodes (75).

VII. CONCLUSION

In this paper, we gave first an overview of clustering algorithms in mobile ad hoc networks and we showed limits of their performances because of the overhead induced by control messages such as Hello messages. We proposed then, a new algorithm which we called MPWCA-L to limit this overhead. This algorithm combines both the prediction mechanism of MPWCA, and the local election process of WCA-L. More specifically, we increased the interval between two messages, and during this long time, nodes try to estimate the movement of their cluster-head and then anticipate handovers, to avoid link breaks. In addition, we considered that an isolated ordinary node can become itself cluster-head and form its own cluster and we proposed to restrain the range of an election to an area where at least two cluster-heads are one-hop neighbours.

This paper gave the MPWCA-L description in describing our two algorithms MPWCA and WCA-L. Simulations are also performed to evaluate the stability of the dominant set and the QoS in terms of packet delivery rate and overhead provided by our algorithms. We compared the results of the proposed approaches with the WCA using different performance metrics. Results showed that our algorithms, and especially MPWCA-L, perform better than WCA.

REFERENCES

[1] D.J. Baker and A. Ephremides. A distributed algorithm for organizing mobile radio telecommunication networks. In 2nd international Conference on Distributed Computing Systems, pages 476-483, Paris, France, April 1981.

[2] D.J.Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm. IEEE Transactions on Communications, 29(11) :pp. 1694–1701, 1981.

[3] Hideaki Takagi and Leonard Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. IEEE transactions on communications, 32(3), pp. 246-257, march 1984.

[4] 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, Vol. 75(1), 1987, pp. 56–73. Also available as technical report in september 1988 in
http://www.dtic.mil/dtic/tr/fulltext/u2/a200928.pdf

[5] A.K. Parekh. Selecting routers in ad-hoc wireless networks. In Proceedings of the SBT/IEEE International Telecommunications Symposium, august 1994.

[6] M. Gerla and J. Tsai. Multicluster, mobile, multimedia radio network. ACM/Baltzer Journal of Wireless Networks, 1(3), pp. 255-265, 1995.

[7] Chunhung Richard Lin and Mario Gerla. Adaptive clustering for mobile wireless networks. IEEE Journal of Selected Areas in Communications, 15(7) , pp. 1265–1275, 1997.

[8] Ching-Chuan Chiang, Hsaio-Kuang Wu, Winston Liu, and Mario Gerla. Routing in clustered multihop mobile wireless networks with fading channel. In Proceedings of IEEE Singapore Internationale Conference on Networks, pp. 197–211, april 1997.

[9] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In Workshop on Parallel and Distributed Simulation, pp. 154-161, 1998.

[10] R. Bagrodia and R. Meyer. PARSEC: A Parallel Simulation Environment for Complex System. Technical report, UCLA, 1998.

[11] S. Basagni, Distributed clustering for ad hoc networks, in: Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, June 1999, pp. 310–315.

[12] S. Basagni, Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks, in: Proceedings of Vehicular Technology Conference, VTC, Vol. 2, 1999-Fall, pp. 889–893.

[13] Mario Gerla, Taek Jin Kwon, and Guangyu Pei. On demand routing in large ad hoc wireless networks with passive clustering. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), 2000.

[14] 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), 5, pp. 193-204, April 2002.

[15] JANE Y. YU AND PETER H. J. CHONG, A SURVEY OF CLUSTERING SCHEMES FOR MOBILE AD HOC NETWORKS, IEEE Communications Surveys & Tutorials, VOLUME 7, NO.1, 2005, pp. 32-48

[16] Vincent Bricard-Vieu and Noufssa Mikou. QoS in MANETs using a Mobility Prediction-based Weighted Clustering Algorithm. In the Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMob 2005), pp. 390-396 August 2005.

[17] Vincent Bricard-Vieu and Noufssa Mikou, A Weighted Clustering Algorithm using local cluster-heads election for QoS in MANET. In Proceedings of GLOBECOM 2006, pp. 1-5, 2006.

[18] V. Bricard-Vieu, N.Nasser and N. Mikou, A Mobility Prediction-based Weighted Clustering Algorithm Using Local Cluster-heads Election for QoS in MANETs, Wimob 2006, pp. 24-30.

[19] Jane Yang Yu, Peter Han Joo Chong, An efficient clustering scheme for large and dense mobile ad hoc networks (MANETs), Computer Communications 30, 2006, pp. 5-16.

[20] Yan Zhang, Jim Mee Ng, Chor Ping Low, A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks, Computer Communications 32, 2009, pp. 189-202.

[21] K. Drira, H. Seba, H. Kheddouci, ECGK: An efficient clustering scheme for group key management in MANETs, Computer Communications 33, 2010, pp. 1094-1107.

[22] Hamid Ali, Waseem Shahzad, Farrukh Aslam Khan, Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization, Applied Soft Computing 12, 2012, pp. 1913-1928.