A Quality of Service Driven Approach for Clustering in Mobile Ad hoc Networks Based on Metrics Adaptation: Looking Beyond Clustering

Zouhair El-Bazzal, Michel Kadoch, Basile L. Agba, Mohamad Haidar, and François Gagnon

Original scientific paper

Abstract: **Recently, research topics are focusing on clustering approaches for Ad hoc networks due to their effectiveness in building a virtual backbone formed by a set of suitable clusterheads (CH) to guarantee the communications across clusters. In this paper, we propose a clustering approach to elect suitable nodes' representatives and to store minimum topology information by reducing the propagation of routing information which facilitates the spatial reuse of resource and increase the system capacity. The clusters must adapt dynamically to the environment changes, we also propose a distributed maintenance procedure that allows managing nodes' adhesion, nodes' handoff and CHs' re-election. Based on our analytical model used to estimate the quality of service (QoS) parameters, we implement an admission control algorithm to determine the number of members inside a cluster that can be accommodated while satisfying the constraints imposed by the current applications. This might effectively drive congestion avoidance on the CH and interclusters load-balancing to achieve better network resource utilization. The obtained results will help us to readjust the clustering algorithm metrics in order to provide better maintenance and QoS guarantees depending on the used applications. Through numerical analysis and simulations, we have studied the performance of our model and compared it with that of other existing algorithms. The results demonstrate better performance in terms of number of clusters, number of handoffs, number of transitions (state change) on CHs, QoS parameters, load balancing and scalability. We also observed how the connectivity and the stability are maximized when the number of nodes increases in presence of the mobility.**

***Index Terms*: Ad hoc networks, clusters, maintenance, quality of service, scalability.**

I. INTRODUCTION

Clustering is a promising approach for mimicking the operation of the fixed infrastructure and managing the resources in mobile Ad hoc networks. An Ad hoc network is characterized by a collection of wireless nodes which communicate with each other using high-frequency radio waves. These nodes arbitrarily and randomly change their locations and capabilities without the aid of any fixed infrastructure. The main objective of clustering is to elect suitable nodes' representatives, i.e. CHs and to store minimum topology information by reducing the propagation of routing information which facilitates the spatial reuse of resource and increase the system capacity. Each CH will act as a temporary base station within its zone or cluster and communicates with other CHs. Thus, packets for route finding may only spread among CHs instead of flooding among all nodes. On the other hand, the topology change information caused by movement of some nodes is limited in adjacent clusters, not in the whole network. Therefore, any clustering scheme should be adaptive to such changes with minimum clustering management overhead incurred by changes in the network topology.

To establish a cluster, traditional clustering algorithms suggest CH election exclusively based on nodes' IDs or location information and involve frequent broadcasting of control packets, even when network topology remains unchanged. Most recent work takes into account additional metrics (such as energy and mobility) and optimizes initial clustering without taking into consideration the QoS parameters like "cluster achievable throughput, cluster delay and transmissions' number per packet". In many situations, re-clustering procedure is hardly ever invoked; hence initially elected CHs soon waste their batteries due to serving the other members for longer periods of time. In addition, a topology control mechanism is required to mitigate the vulnerability of such clusters due to node joining/leaving and link failures. It aims to reduce interference and energy consumption, to increase the effective network capacity, and to reduce the end to end delay.

As election of optimal clusterheads is an NP-hard problem [1], many heuristic mechanisms have been proposed. Centralized algorithms rely on the assumption that the elected CH is responsible of the cluster's maintenance. However, these algorithms suffer from single point (CH) of bottleneck especially in highly mobile environments; hence initially elected CHs have to collect excessive amounts of information and soon reach battery exhaustion. On the other hand, distributed algorithms are more adaptive to mobility due to the fact that the maintenance is done in collaboration between all the nodes where each node relies on the local information collected from the nearby nodes. Although the distributed manner is preferred for MANET, it lacks a major drawback in achieving and guarantying a strong connectivity between the nodes. The performance changes greatly for small and large clusters and depends strongly on the formation and maintenance procedures of clusters which should operate with minimum overhead, allowing mobile nodes to join and leave

Manuscript received March 03, 2008, and revised December 14, 2008.
Authors are with the Department of Electrical Engineering, Ecole de technologie superieure, University of Quebec, Canada (email: zouhair.el-bazzal@etsmtl.ca).

1845-6421/08/8009

without perturbing the membership of the cluster and preserving current cluster structure as much as possible. On the other hand, under high traffic load conditions, admission control becomes necessary in order to provide and maintain the QoS of existing members. Our approach differs from others in that it is based on the clusters' capacity and it uses the link lifetime instead of the node mobility for the maintenance procedure. We refer this to the fact that the node mobility metric does not affect the election of a CH as much as the link stability metric does. It also provide an efficient technique to estimate and derivate the number of members inside a cluster with respect to the allowed saturation throughput, the delay and the packet error rate when the protocol DCF (Distribution Coordination Function) is used to access the channel. The estimation methodology builds on the existence of a mathematical relationship between the number of competing members, the packet collision probability encountered on the shared medium and the packet arrival rate at each member.

In this paper, we propose a weight based Efficient Clustering Algorithm (ECA) which underlay on node stability level, QoS parameters requested within a cluster as well as on sophisticated distributed maintenance procedures. Our objectives are yielding low number of clusters, maintaining stable clusters, achieving load-balancing and scalability. In this manner, we propose an analytical model to estimate the QoS parameters in order to determine the cluster size that can be accommodated while satisfying the constraints imposed by the applications. This will help us to readjust the used parameters of the clustering algorithm in order to provide better quality of service guarantees depending on the used applications. The results show that the proposed clustering model provides better performance in terms of number of formed clusters, number of handoffs, average number of transition (state change) on CHs, QoS parameter, load balancing, and scalability (connectivity) when compared to that of other weight based algorithms. The paper is organized as follows. In Section II, we review several approaches proposed previously. Section III presents the cluster formation model. Section IV presents the proposed analytical model. Section V presents the distributed maintenance model. Section VI presents the performance analysis of the model. Finally, Section VII concludes the paper.

II. Overview of the Existing Approaches

Many approaches have been proposed for the election of clusterheads ad hoc networks. Maximum Connectivity Clustering (MCC) [2] is based on the degree of connectivity. A node is elected as CH if it is the highest connected node. This is not suitable in dynamic network topologies where the degree of connectivity changes rapidly. The Highest-Degree [3] uses the degree of a node as a metric for the selection of clusterheads. The degree of a node is the number of neighbors each node has. The node with maximum degree is chosen as a clusterhead; since the degree of a node changes very frequently, the CHs are not likely to play their role as clusterheads for very long. In addition, as the number of ordinary nodes in a cluster is increased, the throughput drops and system performance degrades. The Lowest-Identifier (LID) [4, 5, 6] chooses the node with the lowest ID as a clusterhead, the system performance is better than Highest-Degree in terms of throughput. However, those CHs with smaller IDs suffer from the battery drainage, resulting short lifetime of the system. Based on an analytical performance model, Lian et al. [7] have concluded that Lowest-id performs better than MCC in terms of bandwidth consumption and induced control overhead. Least Clusterhead Change Algorithm (LCC) [8] allows minimizing clusterhead changes that occur when two CHs come into direct contact. In such a case, one of them will give up its role and some of the nodes in one cluster may not be members of the other CH's cluster. Therefore, some nodes must become CH while causing a lot of re-elections because of the propagation of such changes across the entire network.

3hBAC (3-hop Between Adjacent Clusterheads) [9] have introduced the concept of clusterguests which are some nodes that may not be connected to any CH, but can access a few clusters with the help of member nodes. The elected CHs are 3-hops away from each other. First, the node with the highest degree is elected as CH and forms the first cluster. Then, the formation of other clusters is done in parallel in the whole network. However, the stationary assumption is required for cluster formation in order to decide the first cluster. DDCA (Distributed Dynamic Clustering Algorithm) [10] uses (α, t) criteria that indicate that every node in a cluster has a path to every other node during some time period with a probability greater than α regardless of the hop distance between them. A node can join a cluster if it has a mutual path to satisfy the above criteria between itself and the CH of that cluster. Otherwise, the node creates a new cluster just to cover itself. DDCA can adaptively adjust its cluster size, considering the same stability level α. In low mobility, it forms large-sized clusters, but in highly mobile network, it diminishes the cluster size.

MOBIC [11] uses a new mobility metric; Aggregate Local Mobility (ALM) for CH election. ALM is computed as the ratio of received power levels of successive transmissions (periodic hello messages) between a pair of nodes, which means the relative mobility between neighboring nodes. It is easy to see that MOBIC is effective for group mobility behavior, in which a group of nodes moves with similar speed and direction. Thus, an elected CH can normally promise the low mobility with respect to its member nodes. However, if nodes move randomly and change their speeds from time to time, the performance of MOBIC will be greatly degraded. DLBC (Degree Load Balanced Clustering) [12] periodically runs the clustering to keep the number of nodes in each cluster around a system parameter named Elected Degree (ED), which indicates the optimum number of nodes that a CH can handle. This mechanism tries to balance the load between the CHs. However, DLBC will cause frequent clustering invocations because the movement of mobile nodes and consequent link setup/break results in dynamic variation of node degree. In addition, how to decide the used system parameters is not discussed in DLBC.

The Distributed Clustering Algorithm (DCA) [13] and Distributed Mobility Adaptive clustering algorithm (DMAC)

[14] are enhanced versions of LID; each node has a unique weight instead of just the node's ID, these weights are used for the selection of CHs. A node is chosen to be a clusterhead if its weight is higher than any of its neighbor's weight; otherwise, it joins a neighboring clusterhead. The DCA makes an assumption that the network topology does not change during the execution of the algorithm. Thus, it is proven to be useful for static networks when the nodes either do not move or move very slowly. The DMAC algorithm, on the other hand, adapts itself to the network topology changes and therefore can be used for any mobile networks. However, the assignment of weights has not been discussed in the both algorithms and there are no optimizations on the system parameters such as throughput and power control.

The Weighted Clustering Algorithm (WCA) [15] is based on the use of a combined weight metric that takes into account several parameters like the node-degree, distances with all its neighbors, node speed and the time spent as a clusterhead. 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. Based on assigned weights similar to those used in WCA, the authors in [16] proposed the formation of k-clusters (member is k-hops away from its CH) by limiting the size of each cluster to S nodes. The maintenance is distributed among all the nodes for load balancing purposes. The simulations show better performance when comparing to LID and LCC. However, the assignment of k and S values has not been discussed, the maintenance seems very complicated and the induced overhead is very high.

III. Cluster Formation Procedure

A. Preliminaries and Network Design

An ad hoc network is formed by a set of nodes and links that can be represented by a graph G(V, E). V represents the set of mobile nodes v_i. E represents the set of links between the connected nodes. The network cardinality is represented by: $n = |V|$ and $E = \{(v_i, v_j) \in V^2 \mid P_{j,i} \quad P_{i,j} \geq threshold\}$, where $P_{i,j}$ represents the received power from v_i at v_j. Assume that $N_k(v_i)$ is the set of k-hops neighboring nodes of v_i and $\Delta_k(v_i)$ is the cardinality of $N_k(v_i)$.The clustering is a graph partitioning problem with some added constraints and metrics. More formally, we look for the set of vertices $C \subseteq V(G)$ of cardinality $N = |C|$ such that: $\bigcup_{v_i \in C} N_k(v_i) = V$ and

$$\forall\, v_i \in C, \exists\, v_j \in V \mid \{(v_j = v_i) \quad (v_j \in N_k(v_i))\} \qquad (1)$$

Vuong et al. [18] concluded that the construction of k-clusters in an ad hoc network is a NP-Complet problem. For that reason, we are focusing on the case of 1-clusters, where $k = 1$. More formally: $N_1(v_i) = \bigcup_{v_j \in C, v_j \neq v_i} \{v_j \mid P_{i,j} \geq threshold\}$.

B. Code Assignment and Node States

Enabling CDMA-based solutions for ad hoc is fraught with challenges, due to the absence of centralized control. An ad hoc network is time-asynchronous system because it is generally not feasible to have a common time reference for all the transmissions that arrive at the receiver side. In addition, it is not possible to design orthogonal spreading codes due to multipath effect which make these transmissions suffer different time delays. In this case, the cross-correlation between codes cannot be neglected. We also take into consideration these assumptions:

a) There is a common signaling CDMA code named "sig_code" used by all nodes (new arrival nodes, existing nodes and CHs) for signaling messages (i.e. node wish to join/leave a cluster, CH admits a node, CH rejects a node, etc.). This might help in separating the payload traffic from the signaling traffic.
b) There is a common CDMA code named "intercluster_code" for interclusters' communications. CHs will communicate between them using this unique code.
c) CH assigns a CDMA code exclusive to the cluster named "intracluster_code". All nodes that are accepted as belonging to the cluster must use this code to communicate with the CH. No other nodes can interfere with the throughput of these nodes. Other nodes that will be rejected by the CH must align themselves with other clusters or create a new cluster.

We assume that the total number of "intracluster_codes" is large, so that neighboring clusters are not assigned the same "intracluster_code". However, a spatial reuse of these codes may be used. With a large number of codes, it is unlikely for a node to have an interfering node which is on the same channel but associated to another CH and using a different quasi-orthogonal "intracluster_code" (*Gold, Kasami*). CHs also maintain a list of the intracluster_codes used in the neighboring clusters in order to avoid interferences and to reduce the collisions (bit error rate).

C. State of a Node Inside the Network

For cluster formation, we allocate IDs for the nodes and for the CHs. The 2-tuplet (id_{v_i}, id_{CH}) forms a unique identifier for every node in the ad hoc network. Every node in the cluster will have information about its CH so that it can communicate across the cluster. More formally, every node $v_i \in V$ will be identified by the following state:

v_i: $(id_{v_i}, id_{CH}, w_i,$ intracluster_code, intercluster_code, sig_code, counter$)$

The weight parameter w_i is periodically calculated by each node in order to indicate the suitability of a node for playing CH's role. The 'counter' is maintained by each node in order to calculate the QoS level inside the cluster.

D. Weight Calculation and Clusterhead Election

Each node should be able to compute its weight based on several clustering metrics. The CH' election is based on the weight values of the nodes. In comparison with other approaches, we do not use neither the sum of distance for the largest range coverage, nor the cumulative time during which the node acts as a CH. The metrics used are the following:

a) The node degree $\delta_i = \Delta_1(v_i)$: defined as the actual number of neighbors for v_i. It allows estimating the connectivity degree and the position of that node regarding the other nodes.
b) The transmission power P_i: which allows electing the node

the can cover the largest range.

c) The average speed S_i: which is calculed as:

$$S_i = \frac{1}{T}\sum_{t=1}^{T} \sqrt[2]{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}$$

Where (X_t, Y_t) represents the node's coordinate of v_i at the moment t.

d) The remaining battery power E_i: which allows extending the lifetime of nodes by relinquish the role as a CH in case of insufficient battery power.

Once these metrics are calculed, every node v_i must calculate its combined weight w_i which is a good indication for its suitability to be a CH. The node which has the best weight value within the 1-hop neighboring is elected as CH. The weight calculation is done periodically and locally as following:

$$w_i = a \times \delta_i + b \times E_i + c \times P_i + d \times S_i;\ a + b + c + d = 1 \quad (2)$$

We suppose that the nodes inside each cluster have the same needs. The idea is to combine each of the above system parameters with certain weighing factors depending on the system needs. The flexibility of changing the weighting factors helps us apply our algorithm to various scenarios. For example, in low mobility environment (conference room), we can privilege the remaining battery parameter, thus the factor 'b' can be adjusted on the nodes. In a military environment where the battery energy is always available, we can adjust the factor 'd' in order to elect the less mobile node as CH. On the other hand, we believe that the node average speed does not reflect the relative mobility of that node. However, we still use that parameter for initial CH election, because it is impossible to estimate the relative mobility when the node is alone in the zone without any other reference point (neighbors). During maintenance procedures, the re-election is more sophisticated; we use the historical of the received power signals from the one-hop neighboring to determine the relative mobility of a node regarding the others. This mechanism is detailed in the section V.

IV. Analytical Model For Admission Control Based On Cluster QoS Level

Under overloading conditions in the network, the load balancing and the admission control are required in order to provide a certain level of QoS for the participant nodes. The estimated knowledge of the number of members sharing an 802.11 cluster might effectively drive congestion avoidance on the CHs and inter-clusters load-balancing to achieve better network resource utilization. First, we start by modeling the intraclusters and interclusters performance parameters based on the code assignment procedure mentioned previously. More specifically, we investigate the impact of active nodes and wireless channel collisions on the performance of the DCF. Based on this model, the cluster will be able to admit or reject the nodes by restricting the input traffic being admitted to the system in order to maintain the QoS of the existing members.

A. Modeling the DCF Channel Parameters

Note that the collisions and transmission errors can only occur between the members of the same cluster due to the use of the same CDMA intracluster_code inside that cluster. The communications in neighboring clusters that use a distinct quasi-orthogonal intracluster_code do not affect at all those of the current cluster. Therefore, to model the exponential backoff schema implemented in DCF 802.11 MAC layer, we know that for each packet transmission, a node initializes a backoff time which is a random integer uniformly distributed over the interval $(0, \mathrm{W}-1)$. The value of W is called contention window, and depends on the number of failed transmissions of the packet. At the first transmission attempt, the value of W is equal to $\mathrm{CW_{min}}$ called the minimum contention window.

Let p be the probability that the transmitted packet faces a collision in the channel due to two or more nodes transmitting simultaneously in the same slot. In this case, after each unsuccessful transmission, the value of W is doubled, up to a maximum value $\mathrm{CW_{max}} = 2^{\mathrm{m}}\mathrm{CW_{min}}$ where m represents the number of unsuccessful attempts for this packet, i.e., the maximum backoff stage. Once W reaches $\mathrm{CW_{max}}$, it keeps this value until it returns to its initial value $\mathrm{CW_{min}}$. Now, we can derive the general probability that the contention window W that a node chooses and is given by:

$$\mathrm{P\{Window = W\}} = \begin{cases} \mathrm{p}^{\mathrm{m}-1}(1-\mathrm{p}) \text{ for } \mathrm{W} = 2^{\mathrm{m}-1}\mathrm{CW_{min}} \\ \mathrm{p}^{\mathrm{m}} \text{ for } \mathrm{W} = \mathrm{CW_{max}} \end{cases} \quad (3)$$

The authors in [19] derived the collision probability p for the case of saturated network where a transmitting node has persistently a queue of packets to send, so each incoming packet is immediately backlogged, i.e. it is preceded by a backoff. At saturation, each packet is backlogged immediately. The average backoff window in the saturated case is given by:

$$(1-\mathrm{p})\frac{\mathrm{W}}{2} + \mathrm{p}(1-\mathrm{p})\frac{2\mathrm{W}}{2} + \cdots + \mathrm{p}^{\mathrm{m}}(1-\mathrm{p})\frac{2^{\mathrm{m}}\mathrm{W}}{2} + \mathrm{p}^{\mathrm{m}+1}\frac{2^{\mathrm{m}}\mathrm{W}}{2} = \frac{1-\mathrm{p}-\mathrm{p}(2\mathrm{p})^{\mathrm{m}}}{1-2\mathrm{p}}\frac{\mathrm{W}}{2} \quad (4)$$

B. Modeling the Intraclusters Performance Parameters

In this paper, we extend the model proposed in [19] to obtain an approximate expression for collision probabilities in case of non-saturated arrival rates. Therefore, we use Poisson data source instead of saturated data source. We consider a cluster with "N" members operating in discrete time where each member could be represented as an M/M/1 queuing system with an infinite storage, the packet arrival process is a Poisson memoryless processes with a rate given by λ packets while the packet service process of the network has an exponential distribution with first moment μ which will be explained in subsequent sections. The probability that the packets' interface queue is empty could be approximated by: $\pi_0(\mathrm{node}) = 1 - \frac{\lambda}{\mu}$.

A packet is backlogged if the system is non-empty at the instant of arrival. When an arbitrary arrival occurs, we

approximate the probability that the cluster (N members in steady state) is empty by: $\pi_0(\text{cluster}) = \left(1-\frac{\lambda}{\mu}\right)^{N}$.

Then, the backoff window is 0 for any arbitrary packet with probability $\pi_0(\text{cluster})$, and it is backlogged with probability $1-\pi_0(\text{cluster})$. Therefore, the average backoff window size for general (non-saturated) arrival rates is given by:

$$\overline{W} = \left[1-\left(1-\frac{\lambda}{\mu}\right)^{N}\right]\left[\frac{1-p-p(2p)^{m}}{1-2p}\frac{W}{2}\right] \quad (5)$$

Let T be the saturation throughput inside the cluster, defined as the expected time needed to transmit the data payload with respect to idle, collision and header transmission time, during a cycle of frame exchange. A cycle of frame exchange consists of several collision cycles and one successful data frame transmission plus header transmission and idle times.

$$T = \frac{P(successful\ transmission\ of\ a\ packet\ in\ a\ slot)\times Size\ (packet)}{duration\ cycle_{succes}+duration\ cycle_{errors}+duration\ cycle_{idle}} \quad (6)$$

Where $\begin{cases} duration\ cycle_{succes} = \alpha \times \text{P(successful transmission)} \\ duration\ cycle_{errors} = \beta \times \text{P(unsuccessful transmission)} \\ duration\ cycle_{idle} = \gamma \times \text{P(idle slot)} \end{cases}$

α represents the time that the channel is captured with a successful node transmission, β represents the duration of a unsuccessful transmission, i.e., the time that the channel is captured by the node with errors (collisions on the same intracluster_code) and γ represents the duration of a time slot. The value of the parameters α and β differs depending on the access model. Assuming that the packets are data fragment only, that means that there is no fragmentation. Thus, for basic access mechanism:

$$\alpha = DIFS + \frac{PHY_{header}+MAC_{header}+DATA}{\vartheta} + SIFS + \frac{ACK}{\vartheta} + 2\varepsilon$$

$$\beta = DIFS + \frac{PHY_{header}+MAC_{header}+DATA}{\vartheta} + \varepsilon \quad (7)$$

 represents the channel bit rate and ε represents the average propagation delay of any frame on the channel. All these parameters are defined in the IEEE 802.11 standard [20]. Assuming that at an instant t, an arrival to an idle node will not be backlogged even if there are some other backlogged nodes having non empty queues. Now, considering the fact that the cluster contains N members and only those with a nonempty queue can actually collide with packets from other nodes.

On the other hand, assuming q the probability that a node transmits in a randomly chosen slot in $\overline{W}$, the probability p that a transmitted packet faces an unsuccessful transmission on the channel in a given slot will be equivalent to the probability that at least one of the $(N-1)$ remaining nodes transmits in the same time slot. In other words, if we assume that each remaining node transmits a packet with probability q in the same time slot, thus the probability p that a collision occurs is given by:

$$p = 1-\left(\frac{\lambda}{\mu}\right)^{N-1}\left[1-\frac{2(1-2p)}{W\left[1-\left(1-\frac{\lambda}{\mu}\right)^{N}\right]\left[1-p-p(2p)^{m}\right]}\right]^{N-1} \quad (8)$$

Knowing that $p = 1-(1-q)^{N-1}$, therefore:

$\text{P(successful transmission)} = \alpha \times Nq(1-q)^{N-1}$, we also can have $\text{P(unsuccessful transmission)} = \beta \times [1-(1-q)^{N}-Nq(1-q)^{N-1}]$, and $\text{P(idle slot)} = \gamma \times (1-q)^{N}$. By resolving (6), we can obtain the relationship between N and T as:

$$T = \frac{Nq(1-q)^{N-1}\times L(DATA)}{\alpha Nq(1-q)^{N-1}+\beta[1-(1-q)^{N}-Nq(1-q)^{N-1}]+\gamma(1-q)^{N}} \quad (9)$$

The intracluster delay $\overline{D}$ is defined by the average time from the point when a packet becomes head of the node's queue to the point when it is successfully received by the destination, i.e., when a positive acknowledgment is received. We model this delay without considering the waiting time in the packets' interface queue before transmitting: $\overline{D} = \left(\gamma\overline{W}+\alpha\right)+\frac{1}{\mu}$. The service rate μ is expressed via the average time required for successful packet transmission, thus:

$$\mu = \frac{1}{(\gamma\overline{W}+\beta+\Delta)\overline{N}_{cp}} \quad (10)$$

Δ represents the time that a node has to wait after an unsuccessful transmission, before starting to re-sense the channel. Therefore $\Delta = SIFS + ACK\_Timeout$. Owing to the assumption of independence at each retransmission, we can calculate $\overline{N}_{cp}$ (the average number of unsuccessful transmission of a packet) and approximate the average rate of successful transmission per packet $\overline{N}_{sp}$ which follows a geometric distribution having an average of $\frac{1}{1-p}$. Knowing that $\overline{N}_{sp}$ defines the average number of node attempts to successfully transmit its packet, i.e., the node has been exposed to $(\overline{N}_{sp}-1)$ unsuccessful transmission before to successfully transmit its packet; therefore: $\overline{N}_{cp} = \frac{1}{1-p}-1$. Finally, we have:

$$\overline{D} = \left(\gamma\overline{W}+\alpha\right)+\frac{p(\gamma\overline{W}+\beta+\Delta)}{1-p} \quad (11)$$

C. Modeling the Interclusters Performance Parameters

As for the interclusters communications, the CH must route all intraclusters traffic; we assume that the CH has a single antenna that serves either for transmission, or for reception (half duplex channel). Consequently, the CH cannot communicate simultaneously with its members and with other neighboring CH. When listening for intraclusters and interclusters communications, the CH throughput will be affected not only by its members but also by its neighboring CHs. Note that the average number of clusters in the network (if the nodes are uniformly distributed) can be calculated as $\frac{n}{N}$.

On the other hand, when the CH is in transmission mode and all the neighboring CHs are not transmitting; the CH does not have to wait to immediately transmit a packet on the intercluster_code channel. That means that the packet will be directly transmitted if the CH is not in reception mode of incoming traffic on the intracluster_code channel and if the neighboring CHs do not interfere with its own transmissions on the intercluster_code channel. We consider our CHs as traffic aggregation points, thus they will be saturated by transmitting packets and serving the other nodes. Therefore, the average backoff window size of a CH in saturated case is given by:

$$\overline{\acute{W}} = \frac{1-\acute{p}-p(2\acute{p})^m}{1-2\acute{p}}\frac{W}{2} \quad (12)$$

Under the same conditions, the packet collision probability for interclusters communications can be obtained by the following equation:

$$\acute{p} = 1 - \left[1 - \frac{2(1-2\acute{p})}{W[1-\acute{p}-\acute{p}(2\acute{p})^m]}\right]^{\frac{n}{N}-1} \quad (13)$$

On the other hand, assuming $\acute{q}$ the probability that a CH transmits in a randomly slot in $\overline{W}$, thus $\acute{p} = 1-(1-\acute{q})^{\frac{n}{N}-1}$. Following the same definitions given in (6) while taking into account the average number of clusters in the network, the achievable interclusters throughput can be obtained by the following equation:

$$\acute{T} = \frac{\acute{q}\frac{n}{N}(1-\acute{q})^{\frac{n}{N}-1}\times L(DATA)}{\alpha\acute{q}\frac{n}{N}(1-\acute{q})^{\frac{n}{N}-1}+\beta\left[1-(1-\acute{q})^{\frac{n}{N}}-\frac{n}{N}\acute{q}(1-\acute{q})^{\frac{n}{N}-1}\right]+\gamma(1-\acute{q})^{\frac{n}{N}}} \quad (14)$$

By following the same definition given in (11), the interclusters delay is defined as the average waiting time for a successful transmission on the interclusters_code channel:

$$\overline{\acute{D}} = \left(\gamma\overline{\acute{W}}+\alpha\right) + \frac{\acute{p}\left(\gamma\overline{\acute{W}}+\beta+\Delta\right)}{1-\acute{p}} \quad (15)$$

The average end to end delay "d" depends on the average number of hops separating the source node from the destination node:

$$d = 2\overline{D} + \left(\frac{n}{N}-1\right)\overline{\acute{D}} \quad (16)$$

V. Distributed Maintenance Procedure

A. Neighboring Discovery Mechanism

Every node (member or CH) has to maintain an 'intracluster_table' wherein the information on the local members including the weights of each node is stored. However, the CHs maintain another clusterhead information table 'intercluster_table' wherein the information about the other CHs is stored. In complex networks, the nodes must coordinate between each other to update their tables. The hello messages (TTL set to 1 hop) are used to complete this role. A hello contains the full state of the node including its weight; it is periodically exchanged either between CHs or between each CH and its members in order to update the 'intercluster_tables' and the 'intracluster_tables' respectively.

Once a wireless node is activated, it has (id_{CH} = NULL), since it does not belong to any cluster. The node continuously monitors the channel until it figures out that there are some activities in its 1-hop neighborhood. This is due to the ability to receive the hello messages from other nodes in the network. The node still has no stable state until it is fully identified by the reception of the CH identifier, the intracluster_code and the counter values. Table I illustrates the messages used for the maintenance.

B. New Arrival Nodes Maintenance Mechanism

Every node v_i (id_{CH} = NULL) broadcasts a Join message (TTL set to 1 hop) on the sig_code channel in order to join the most powerful 1-hop clusterhead. Thus, it waits either for an Accept or for a Reject from a CH. More formally, this mechanism can be written as the following:

```
for all (v_i ∈ V, id_CH(v_i) = NULL) do /* V represents the set of nodes in the network */
  Start monitoring the channel;
  if (no activity)
    The node declares itself as an isolated node and retries the algorithm later;
  else
    Broadcast Join (id_vi, other fields = NULL);
    Wait during a time interval for a message accept or reject;
    if (there is neither accept nor reject)
      The node declares itself as an isolated node and retries the algorithm later;
    else
      Store during a time interval all the received accept and reject in a vector [ ];
    end
  end
  Search in vector [ ] the accept which has the best weight;
  if (accept is found)
    Send Join_accept to the CH which is the accept's source and wait for CH_ACK from the chosen CH;
    if (CH_ACK is received)
      The node declares itself as a member of the chosen cluster and fulfills its state;
    else
      The node tries with next best weight CH selected from the vector [ ];
    end
  else
    The node chooses the reject which has the best weight and decides to declare itself as CH;
    Send CH_request to the CH which is the reject's source and wait for CH_response;
    if (CH_response is received)
      compteur = 1;
      id_CH = id_vi; /* The node declares itself as CH */
    else
      The node tries with another CH selected from the reject vector [ ];
    end
  end
end
```

TABLE I
Messages Involved in the Clustering Algorithm

Type of the message	Message description
hello (id_{v_i}, id_{CH}, w_i, counter)	To update the tables of the nodes
Join (id_{v_i}, id_{CH})	To affiliate a cluster
Accept (id_{v_i}, id_{CH}, w_i)	The CH accepts a Join
Reject (id_{v_i}, id_{CH}, w_i)	The CH rejects a Join
CH_request (id_{v_i})	The node declares itself as CH
CH_response(id_{CH}, intracluster_code)	The CH accepts a CH_Request
Join_accept (id_{v_i}, id_{CH}, w_i, counter)	The node accepts the Accept
CH_ACK (id_{v_i}, id_{CH}, w_i , intracluster_code, counter)	The CH adds the node as a member
Database_info(id_{v_i}, id_{CH}, w_i, counter)	The current CH sends the database to a new elected CH
Database_ACK (id_{v_i}, id_{CH}, w_i, counter)	The new elected CH accepts the received database
CH_change (id_{v_i})	The CH notifies a CH change
CH_info (id_{v_i}, id_{CH}, w_i, counter)	The CH accepts the presence of a new CH in the network
leave (id_{v_i}, id_{CH})	The node leaves the cluster

C. QoS Driven Procedure for Nodes Admission Control

The reason that a CH accepts or rejects a Join depends on the ability to provide and guarantee the QoS level. We proposed an admission control mechanism based on the analytical model presented previously in order to estimate the achievable throughput and delay. This mechanism will be only implemented on the CHs and applied during the maintenance so that they can take decisions to accept and/or reject the requests while guarantying an acceptable QoS level. More

formally, this procedure can be written as the following:

```
for all v_i ∈ C do /* C represents the set of clusterheads' nodes */
    estimate the intraclusters throughput T;
    estimate the end to end delay d;
    if (new Join is received from a node)
        if (T > threshold or d > threshold)
            reject the Join request;
            send a reject message to the node;
        else
            accept the Join request;
            send a accept message to the node and wait for a confirmation;
            if (Join_accept is received from the node)
                update the counter value;
            end
        end
    end
end
```

D. Clusterheads Maintenance Mechanism

A CH ($id_{CH} = id_{v_i}$) has to calculate periodically its weight, and sends periodically hello messages to its members and to the neighboring CHs in order to update the intraclusters_tables and interclusters_tables respectively. When the CH receives a Leave message, it updates the intraclsuter_table and broadcasts a hello message to its members and to its neighboring CHs to inform them that its previous 'counter' was decremented. When the CH receives a hello from a neighboring CH, it updates the interlcuster_table. If the hello's source is a node member, the CH updates the intracluster_table and verifies the weights. In case of better weight is found, the CH must invoke the re-election procedure. We restrict this procedure to the CHs in order to simplify the maintenance and the complexity of the cluster management. The re-election does not necessarily mean that a new CH must be elected even if there is a member node having a better weight, we will explain in details this procedure in subsequent section.

When the CH receives a Join ($id_{CH} = NULL$) from a new arrival node or a Join ($id_{CH} = x$) from a node which belongs to another cluster x, the CH must invoke the handoff procedure explained in subsequent section in order to admit or to reject the request based on the admission control mechanism presented previously. On the other hand, in order to maximize the scalability of the architecture, if the CH receives a CH_Request from a node desiring to be CH, it must accept the request without performing any admission control; the CH sends a CH_Response to the node, updates the intercluster_table and broadcasts a hello message to the neighboring CHs. Thus, an interclusters link is established on the intercluster_code channel.

E. Handoff Procedure

We have tried to balance the load between all the clusters in the network in order to give more flexibility to the members by allowing them to leave a weak CH and join another one which seems stronger than the current one. However, in order to maintain a certain QoS level, the solicited CH must perform an admission control so that the handoff is not accepted in the detriment of the QoS level of the existing members. More formally, this procedure can be written as the following:

```
for all (v_i ∈ C, id_{v_i} ≠ x) do /* C represents the set of clusterheads' nodes */
    A = Perform admission control as explained previously;
    if (Join is received from a new arrival node) /* (id_CH = NULL) */
        if (A is True)
            Send accept to the source of the message Join and wait during a time interval for Join_accept;
            if (no Join accept)
                Ignore the request;
                Break;
            end
        else /* A is False */
            Send reject to the source of the message Join;
            Break;
        end
    end
    if (Join is received from a node already member of another cluster x /* (id_CH = x) */
        B = Check if the historic indicates that the power of the received signals does not decrement rapidly;
        if (A is False or B is False)
            Ignore the request and send a reject to the source of the message Join;
            Break;
        end
    end /*Join is received from a member node (id_CH = x) or a new arrival node */
    counter = counter + 1;
    Add the node to the intraclusters_table and send a CH_ACK to the new added member;
    Update intraclusters_table and interclusters_table;
end
```

F. Member Nodes Maintenance Mechanism

After joining a cluster, the node declares itself as a member of this cluster; it calculates and sends periodically its weight in hello messages towards its CH. More formally, this mechanism can be written as the following:

```
for all (v_i ∈ V, id_CH(v_i) ≠ NULL) do /* V represents the set of nodes in the network */
    Calculate periodically its weights w_i;
    Send periodically hello messages to its CH;
    if (no hello from the CH during a time interval)
        The CH is down;
        Perform Neighboring Discovery Mechanism;
    end
    if (CH_change is received from the CH)
        Copy the id of the new CH in the field id_CH;
    end
    if (CH_info is received from the CH)
        Copy the id_{v_i} in the field id_CH; /* The node became a CH */
    end
    if (database_info is received as a result of the re-election procedure)
        Prepare itself to become a new CH in the cluster;
        Store received database_info received from the old CH;
        Send database_ACK to the old CH and wait for CH_info;
    else
        if (id_CH(hello) = id_CH(v_i)) /* the hello is received from its CH */
            Update intraclusters_table;
        else /* the hello is received from another CH */
            if weight (hello from another CH) > weight (last hello from current CH)
                Ignore this hello;
            else /* Possibility to migrate to a stronger CH */
                Send Join to that stronger CH while staying as a member of the current CH;
                Wait for CH_ACK from that stronger CH during a time interval;
                if (reject is received from the stronger CH)
                    Ignore the hello;
                end
                if (no CH_ACK is received)
                    Ignore the hello;
                else
                    Send a leave message to the actual CH;
                    Copy the id of that CH in the field id_CH; /* The node became member of a new CH */
                end
            end
        end
    end
end
```

G. Re-election Procedure

The re-election is not periodically invoked; it is performed by the CH just in case of a best received weight, it allows minimizing the generated overhead and the utilization of nodes' resources. As we explained above, the re-election may not result a new CH, it depends on the stability of the new

node for playing the CH's role. In the case where a new CH must be elected, the procedure should be soft and flexible while copying the databases from the old to the new CH. We limit the execution of the algorithm where there is a CH's change to minimize the effect on the whole topology. Thus the furthest nodes are not affected by any problem which occurs in other clusters. More formally, this procedure can be written as the following:

```
for all v_i ∈ C do /* C represents the set of clusterheads' nodes */
    Verify periodically all weights w_j ∈ intracluster_table;
    if (there is a w_j better than w_i & w_i ≤ threshold) /*be ready for reelection*/;
        The old CH stays CH until the reception of a confirmation;
        Send database_info (old CH) to the new elected node;
        Wait during a time interval for a database_ACK from the new elected CH;
        if (no database_ACK)
          Don't perform the reelection and the CH continues playing its role;
          Break;
        else
          The old CH copies the id_{v_j} in the field id_CH;
          Update intraclusters_table and interclusters_table;
          Send CH_info to the elected node;
          Broadcast CH_change () to the members and the neighboring CHs;
        end
    else
      Don't perform the re-election and the CH continues playing its role;
    end
end
```

VI. Implementation And Performance Analysis

A. Simulation Environment and Parameters

We have used the parameters specified in IEEE 802.11b specifications [20] which define the frame sizes of the MAC layer and the Direct Sequence Spread Spectrum (DSSS) physical layer parameters used the previous analytical model. In the rest of the paper, the channel bit rate has been assumed equal to 1 Mbps (for a range of 90 meters) without RTS/CTS. On the other hand, the generation of appropriate scenarios with realistic mobility pattern is very challenging when designing ad hoc networks. Agba and al. [21] have developed a simulation tool including a scenarios generator and a propagation modeler. It allows a complete description of environmental parameters, mobility model parameters and other simulation settings.

The simulations scenarios used in this paper were randomly generated based on the Random Waypoint mobility model (RWM). For the physical layer, a semi-deterministic channel model that takes into account the path-loss calculation with respect of 3D environment parameters is used. The model also allows defining the maximum radio range as the radius of a mobile when operating at full transmission power and having an effective communication range. The network size is 500m x 500m. The number of nodes used in the whole network varies between 20 and 100. All the nodes follow the RWM used in the scenario generator with speed ranging from 3 to 10 Km/h. The hello interval was set to 5 seconds and the simulations were run for 300 seconds.

B. Cluster Performance Analysis and Discussion

Figure 1 shows that the throughput strongly depends on the number of members inside the cluster. We realize that, in most cases, the greater is the number of the active members, the lower is the throughput. This condition remains valid until the value of the window "W" is approximately equal to 500.

We see that an higher value of 'W" tends to give better throughput performance in the case of large members' number in the cluster, while it drastically penalizes the throughput in the case of small members' number. On the other hand, we neglect the probability of having great values of W in case of small number of members, since the probability of collisions in this case is very small in comparison to the case of large number of members.

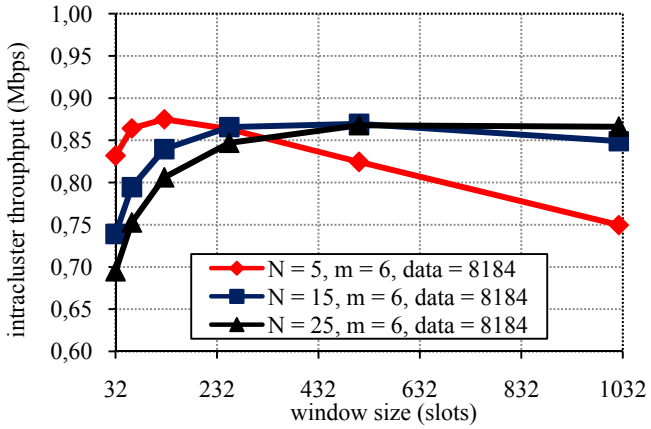

Fig. 1. Cluster throughput versus backoff window size

This behavior is also seen in figure 2, the amount of channel time wasted in collisions is extremely large for a small value W and a large number of members. Large value of W may, in fact, increase the delay; because for small W, the amount of idle slot times per packet transmission is very low. This value becomes significant only when W gets greater and the number of members is small. When the number of members is large, the throughputs of large windows are very close, but the cluster delay deteriorates much more severely for smaller contention windows.

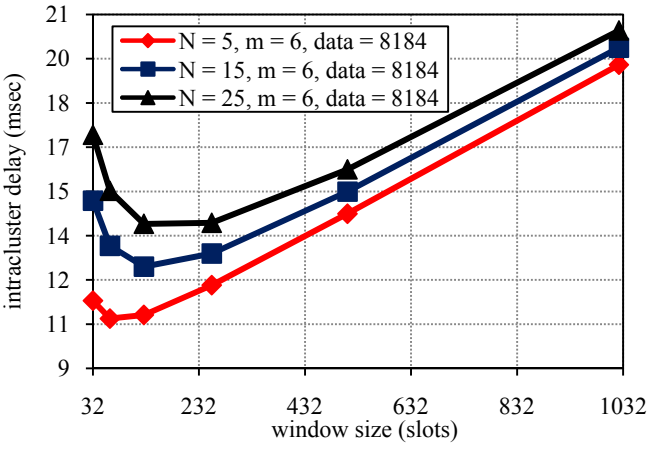

Fig. 2. Cluster delay versus backoff window size

On the other hand, figure 3 shows that the number of transmissions per packet significantly increases as the window size decreases. This effect is much more significant for large number of members. Therefore, we can settle the fact that the clustering of an ad hoc network cannot be concluded by the randomly execution of an algorithm regardless the

specifications of the applications used inside the cluster. By using a sophisticated algorithm like ours (ECA), we can adjust the parameters of this algorithm in order to generate a certain number of clusters which fulfill the requirement of each member in terms of the required throughput, the tolerated delay and the allowed retransmission number per packet.

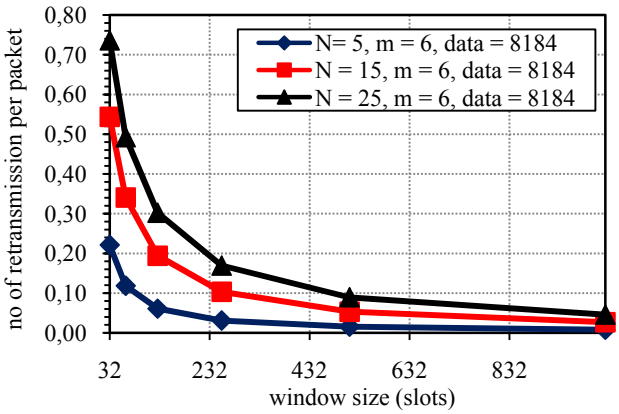

Fig. 3. Average no. of retransmission per packet vs. window size

Figure 4 plots the achievable throughput versus the packet size for three different cluster sizes. We see that when the maximum channel bit rate □ is equal to 1Mbps, the throughput efficiency increases (approaches to □) as the packet size increases. The situation is explained by considering that the time spent for frame transmission is decreased as the data rate increases but the time overhead spent on DIFS, SIFS and the backoff delay remains the same. We can conclude that the choice of a smaller number of members performs better in the whole cluster. We fixed that number to 15 nodes to define the cluster size "N" with respect to the QoS parameters requested by the current applications.

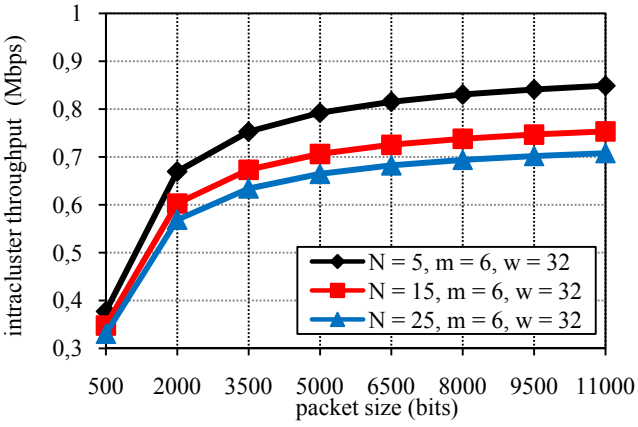

Fig. 4. Cluster throughput versus packet size

Figure 5 shows a strong correlation between the theoretical and simulation throughput results for different transmission range and network density. The average deviation of the simulation from theory is very small. This comes to validate the analytical model proposed in this paper.

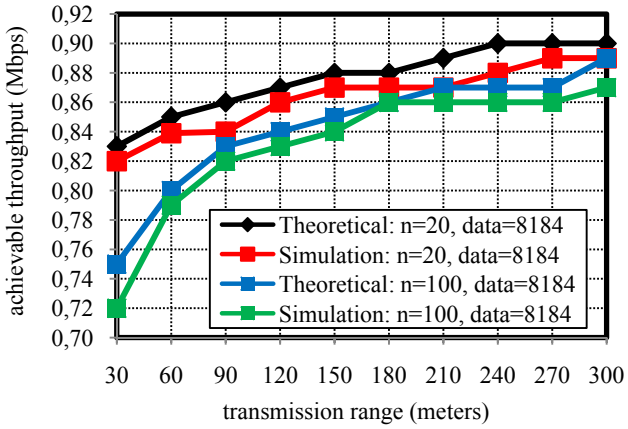

Fig. 5. Achievable throughput (theoretical vs. simulation)

C. Performance Analysis of the Maintenance Procedures

Figure 6 shows that for small ranges, most of nodes remain out of each other's transmission range, thus the number of clusters is relatively high and the network may become disconnected because there are no other choices. When transmission range increases, more nodes can hear each other. The average number of clusters formed decreases and the clusters become larger in size. When the transmission range is very small, most of nodes form one node cluster which only consists of itself.

On the other hand, when the transmission range begins to be larger, mobile nodes tend to remain in the range of their neighbors, the clusters are less dynamic and the number of CHs' transitions decreases as depicted in figure 7. In disconnected networks, we can conclude a high rate of transitions on CHs. However, we argue that this will not affect network performance as this will only occur when the network is disconnected (A disconnected network is unable to function too).

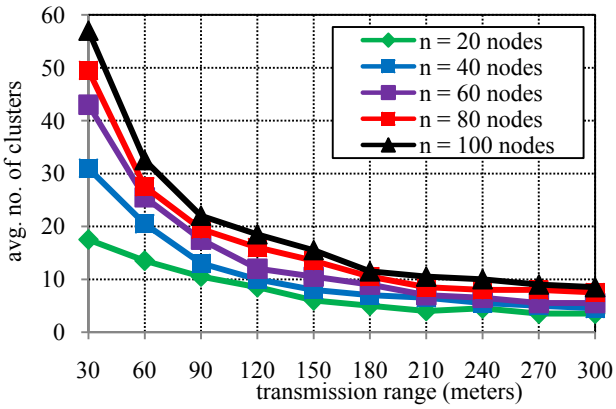

Fig. 6. Avg. number of clusters vs. transmission range

We have also tried to balance the load between all the clusters; we have based on the numerical results obtained from the proposed analytical model to choose the optimal cluster size in respect to the QoS level. The results shown in figure 8 show that none of the clusters has reached a 100 % of load rate. The load is balanced between the different clusters even when we increase the number of nodes in the network. This allows avoiding congestion on CHs and guarantying the scalability issue.

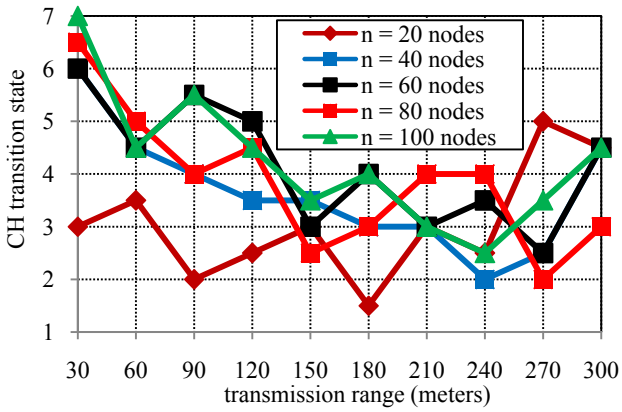

Fig. 7. Avg. no. of CH transition state vs. transmission range

Figure 9 shows that the probability that every node can reach the other nodes is very high. The situation is improved when the topology is denser. By increasing the transmission range, the CH will be able to establish more intercluster links; which improves the global connectivity in the network. We can also remark fewer fluctuations when the number of nodes increases in the network; this is also a good indication on the stability level and the scalability of the approach.

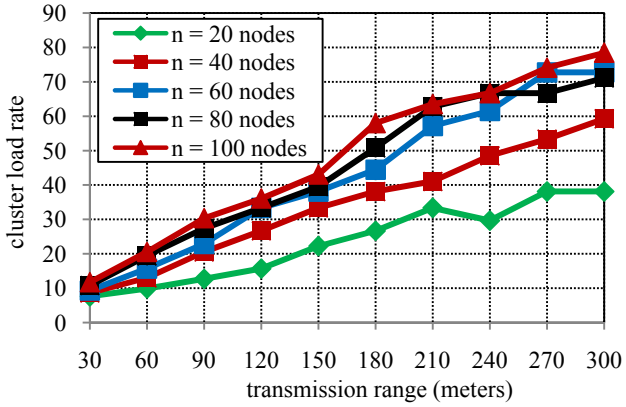

Fig. 8. Cluster load rate vs. transmission range

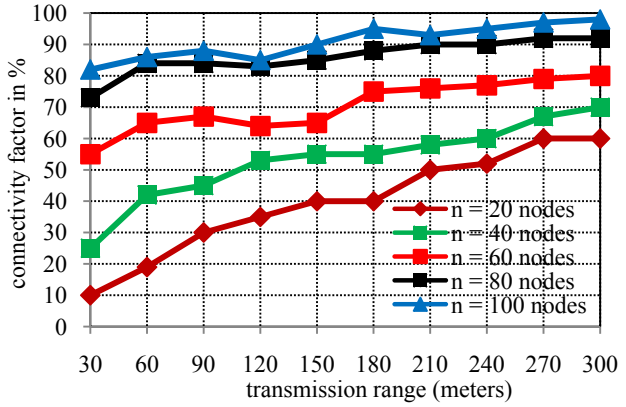

Fig. 9. Connectivity factor vs. transmission range

We also compare the performance of our approach with the corresponding performance of the WCA algorithm while the nodes are moving under the same conditions. In figure 10, we note that the performance difference is small between WCA and ECA with respect to the average number of clusters. This is because both algorithms are variations of a local weight based clustering technique that forms one-hop clusters. For high transmission range (more than 250 m), WCA generates less CH than ECA but to the detriment of a large number of transition on each CH (figure 11), where the stability is one of the important criteria in clustering because the frequent changes of CH adversely affect the performance of the clustering algorithm.

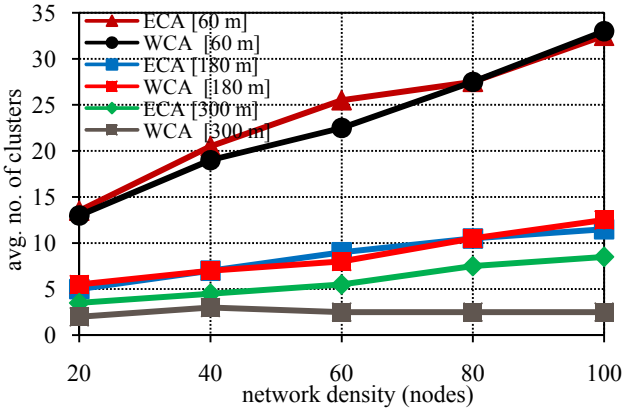

Fig. 10. Avg. no.of clusters vs. network density

As shown in figure 11, with 100 nodes in the ad hoc network and for a transmission range equal to 180m, the proposed algorithm produced about 50.0% to 83.3% less transitions on each CH than WCA. In the WCA algorithm, WCA will keep changing with changes in topology. The CH of WCA algorithm relinquishes its position when another node having better weight joins the cluster. In our algorithm, the CH has to verify the suitability of a new election even if a new node having better weight has joined the cluster. As a result, our algorithm gives better performance in terms of stability when the node density in the network is high.

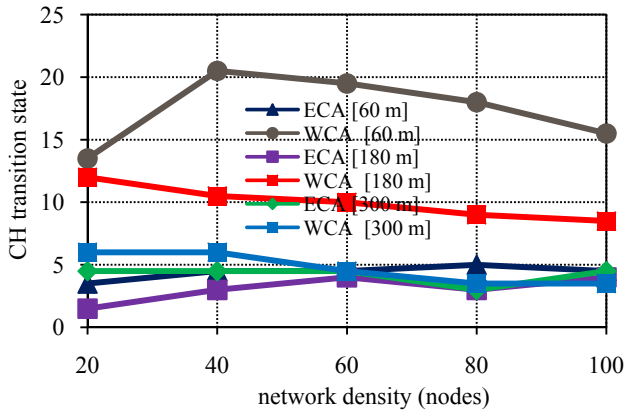

Fig. 11. Avg. no. of CH transition state vs. network density

As shown in figure 12, for a transmission range of 120 meters, the number of handoffs increased when varying the number of nodes in the network for both our algorithm and WCA. As the number of nodes increased, the increasing rate slowed down in ECA, which was not the case in WCA. For a node speed varying between 3 and 10 km/h, when there were 20 nodes in the network and for the same transmission range (120 m), the proposed algorithm produced 61.5% less handoffs than WCA. When the number of nodes was increased to 100, our algorithm gave 66.5% less handoffs than WCA for the same node speed.

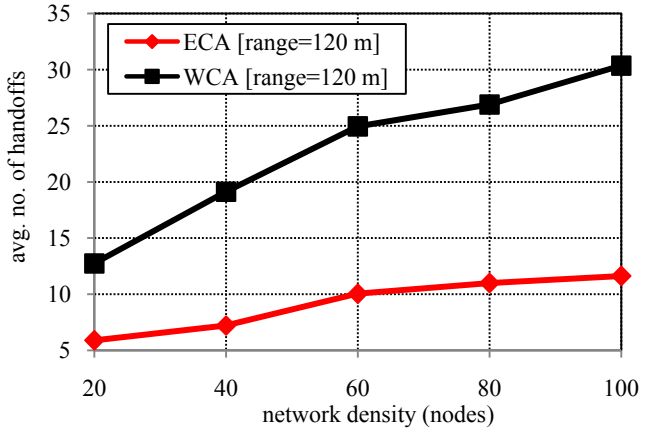

Fig. 12. Avg. no. of handoffs vs. network density

Figure 13 shows the connectivity improvement of ECA in comparison with WCA. We also remark more fluctuations in WCA; ECA tends to stabilize more quickly. This is explained by the higher number of transitions on the WCA's CHs, causing more frequent breaks of links and reducing the overall stability level.

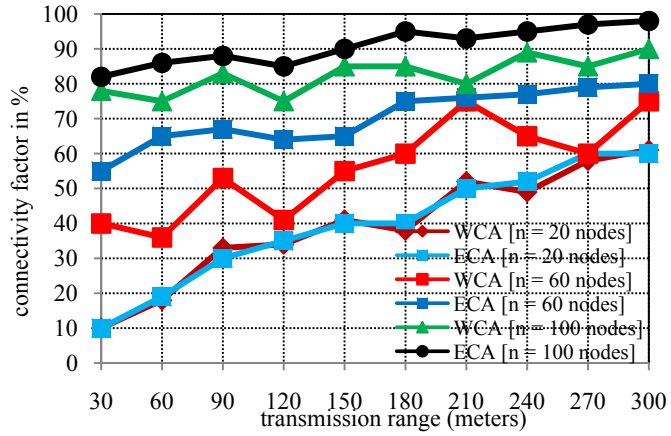

Fig. 13. Connectivity factor vs. transmission range

VII. CONCLUSION AND FUTURE WORK

This paper has presented a Quality of Service Driven Approach for Clustering in Mobile Ad hoc Networks Based on Metrics Adaptation. It has the flexibility of assigning different weights and takes into account a combined metrics to form clusters automatically. An analytical model has been proposed in order to estimate the QoS parameters when we follow a specific CDMA code assignment mechanism in the whole network. We have concluded that the cluster performance strongly depends on the number of members. Limiting the number of nodes inside a cluster allows restricting the number of nodes catered by a clusterhead so that it does not degrade the MAC functioning.

A clusterhead with constrained energy may drain its battery quickly due to heavy utilization; in order to spread the energy usage over the network and achieve a better load balancing among clusterheads, re-election of the clusterheads may be a useful strategy; the algorithm is executed only when there is a demand. Also, if a node is moving away from the clusterhead, then the algorithm is flexible and cheap enough to be applied iteratively as the network configuration changes. Simulation results indicated that the model agrees well with the behavior of the algorithm. For future work, we are planning to investigate the model in presence of clusterheads that are MIMO capable; more research must be done in case of heterogeneous nodes inside the network, especially where the nodes follow distinct mobility models.

REFERENCES

[1] Das B., Bharghavan V., "Routing in ad-hoc networks using minimum connected dominating sets," IEEE International Conference on Communications (ICC Montreal 97), vol. 1, Jun. 1997, pp. 376-380.

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

[3] Gerla M., Tsai J. T. C., "Multicluster, Mobile, Multimedia Radio Network," ACM/Baltzer Wireless Networks Journal 95, vol. 1, Oct. 1995, pp. 255-265.

[4] Baker D.J., Ephremides A., "A distributed algorithm for organizing mobile radio telecommunication networks," Proceedings of the 2nd International Conference on Distributed Computer Systems, Apr. 1981, pp. 476-483.

[5] Baker D.J., Ephremides A., "The architectural organization of a mobile radio network via a distributed algorithm," IEEE Transactions on Communications, Nov. 1981, pp. 1694-1701.

[6] Ephremides A., Wieselthier J. E., Baker D. J., "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling" Proceedings of the IEEE, vol. 75, no. 1, Jan. 1987, pp. 56-73.

[7] Lian, Jie, Kshirasagar Naik et Gordon B. Agnew. 2007. « A Framework for Evaluating the Performance of Cluster Algorithms for Hierarchical Networks ». In IEEE/ACM Transactions on networking. vol. 15, no 6.

[8] Chiang C. C., Wu H. K., Liu W., Gerla M., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," IEEE SICON, Singapore, Apr. 1997.

[9] Yu, J. Y. et P. H. J. Chong. 2003. « 3hBAC (3-hop between Adjacent Clusterheads): a Novel Non-overlapping Clustering Algorithm for Mobile Ad hoc Networks ». In Proceedings IEEE Pacrim'03. (Victoria, 2003), vol. 1, p. 318-321.

[10] McDonald, Bruce et Taib F. Znati. 2001. « Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks ». In Proceedings 34th Annual Simulation Symposium. p. 27-35.

[11] Basu P., Kham N., Little T. D. C., "A mobility based metric for clustering in mobile ad hoc networks," International Conference on Distributed Computing Systems Workshop, Apr. 2001.

[12] Amis, Alan D. et Ravi Prakash. 2000. « Load-Balancing Clusters in Wireless Ad hoc Networks ». In Proceedings 3rd IEEE Application-Specific Systems and Software Engineering Technology. p. 25–32.

[13] Basagni S., "Distributed clustering for ad hoc networks," Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, Jun. 1999, pp. 310-315.

[14] Basagni S., "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," Proceedings of Vehicular Technology Conference, VTC, vol. 2, fall 1999, pp. 889-893.

[15] Chatterjee M., Das S.K., Turgut D., "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," Cluster Computing Journal, vol. 5, no. 2, Apr. 2002, pp. 193-204.

[16] Venkataraman, Gayathri, Sabu Emmanuel et Srikanthan Thambipillai. 2007. « Size-restricted cluster formation and cluster maintenance technique for mobile ad hoc networks ». In International Journal of Network Management. p. 171-194.

[17] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function", IEEE Journal on Selected Areas in Communications, Vol. 18, No. 3, 2000, pp. 535-547.
[18] Vuong, Thai H. P. et Dung T. Huynh. 2000. « Adapting d-hop dominating sets to topology changes in Ad hoc networks ». In Ninth International Conference on Computer Communications and Networks. p. 348-353.
[19] Y. Tay and K. Chua, "A capacity analysis for the IEEE 802.11 MAC protocol", Wireless Networks, Vol. 7, No. 2, 2001, pp. 159-171.
[20] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
[21] Agba L., Gagnon F., Kouki A., "Scenarios generator for ad hoc networks," International Symposium on Industrial Electronics, Montreal, Canada, Jul. 2006.

Zouhair El Bazzal received the B.S. degree in Computer Sciences from the Lebanese University, Beirut, Lebanon in 1999, the M.S degree in telecommunication networks in 2001 from the Ecole Superieure des Ingenieurs à Beirut, Saint-Joseph University, Lebanon, and the Ph.D. degree in engineering in 2008 from the University of Quebec, Canada. Since 2006, he is working as a lecturer in the Department of Software and Electrical Engineering at the École de Technologie Supérieure, Montreal. Also, Dr. El-Bazzal is currently working as a Researcher at Ericsson Canada Inc. His main research interests are: wireless networking and communications, Long Term Evolution in 3GPP, IP mobility, load balancing, admission control, and development of Policy and quality of service tools for LTE.

Michel Kadoch (S'67, M'77, SM'04) received the B. Eng from Sir George Williams University (Canada) in 1971, the M. Eng from Carleton (Canada) in 1974, MBA from McGill (Canada) in 1983 and the Ph.D from Concordia (Canada) in 1991. He is a full professor at Ecole de technologie supérieure ETS (Canada) and the director of the Master Program in engineering. He is also an adjunct professor at Concordia University (Canada). He is presently working on Cross layer, and on Reliable multicast in wireless Ad hoc and WiMax networks. Professor Kadoch has published many articles and is the author of a book « Protocoles et réseaux locaux ». He has been involved for many years at ITU-T as a special rapporteur and with the industry namely Teleglobe Canada, CAE, and Communication Canada. He has been a consultant with Harris, Bell South, BC Tell, Concert and British Telecom UK, as well as the Commonwealth Telecommunication Organization on issues related to installation and optimization of SONET, telephone and high speed data networks and services.

Basile L. Agba (M' 2005) was born in Kara, Togo. He received his B.S in physics in 1998 (university of Lomé, Togo). He received the M.S degree in high frequency telecommunications in 2001 and he completed the Ph.D. degree in high frequencies electronics and optoelectronics in 2004 (University of Limoges, France). From 2005 to 2007, he is Postdoctoral Fellow at Electrical Engineering Department (École de Technologie Supérieure, Montreal - Canada) where he mainly worked on tactical ad hoc networks project in partnership with Ultra Electronics (TCS). Since 2007, Dr. Agba worked as Senior Researcher in telecommunications at the Research Institute of Hydro-Quebec, IREQ. His main research interests are: channel modeling particularly in high voltage environments, wireless networks design, deployment software for wireless systems and antenna design.

Mohamad Haidar received his B.S. degree in Electrical Engineering from the Lebanese American University, Byblos, Lebanon in 2000. The M.S. degree was received from Roosevelt University in Telecommunications, Chicago, IL, in 2002 and Ph.D. in Applied Science from the University of Arkansas at Little Rock (UALR), Little Rock, AR, in 2007. His major field of research is in wireless networking and communications. Previously, he did a one year post-doc at Ecole de Technol,ogie Superieure, Montreal, Quebec, in 2008 where he developed a power control algorithm for Cognitive Wireless Mesh Networks. Prior to that, he worked as a Development Engineer at IBM, Durham, NC, teacher assistant at UALR, Little Rock, AR, and WiMesh/WiMax RF Engineer Analyst on a UALR-Alltel-Motorola project. His research interests are load balancing, channel assignment, MAC and PHY layers design in WLAN, Cognitive Wireless Mesh Networks, Ad Hoc Networks, sensor networks, and WiMax environments.

François Gagnon (S'87-M'87-SM'99) was born in Québec City, QC, Canada. He received the B.Eng. and Ph.D. degrees in electrical engineering from École Polytechnique de Montreal, Montreal, QC, Canada. Since 1991, he has been a Professor in the Department of Electrical Engineering at the École de Technologie Supérieure, Montreal. He has chaired the department from 1999 to 2001 and is now the holder of the Ultra Electronics (TCS) Chair in Wireless Communication at the same university. His research interest covers wireless high-speed communications, modulation, coding, high-speed DSP implementations, and military point-to-point communications. He has been very involved in the creation of the new generation of high-capacity line-of-sight military radios offered by the Canadian Marconi Corporation, which is now Ultra Electronics Tactical Communication Systems.