# Self-organization and Topology's Control for Mobile Ad-hoc Networks

Maâmar Kef[1], Leila Chergui[1], and Mohamed Benmohammed[2]
[1]Department of Computer Sciences, University Larbi Ben Mhidi, Algeria
[2]Department of Computer Sciences, University Mentouri, Algeria

**Abstract:** *The study and the implementation of routing algorithms to ensure the connexion of Ad-hoc networks as often defined (every summit may reach any other) is a hard task to achieve. The environment is dynamic and evolves through time; the topology of network may change frequently. Two distinct approaches tried to apprehend this problem. The first one is based on a flat vision of network, and the second one on an auto-organisation structure. This paper deals with the study of auto-organisation solution for Ad-hoc networks, and the benefit effects brought by this last to the routing-process.*

**Keywords:** *Auto-organization, topology's control, routing-process, and ad-hoc networks.*

*Received April 12, 2008; accepted August 4, 2008*

## 1. Introduction

A mobile Ad-hoc network consists on much of mobile unities moving in any territory with wireless interfaces as their only mean of communication, without any need for pre-existent infrastructure or centralized administration [1, 2, 4, 9]. Here, we are going to present a virtual structure of auto organisation for this kind of networks, then we will analyse a proposition of routing-algorithm that will profit from the architecture of the obtained network. We will begin by the presentation of distributed algorithm of network's construction, as well as the way to maintain it; and its use to improve routing protocols. Then we will expose the different aspects of the proposed routing-algorithm. The algorithm of auto-organization will be:

- Simple (Converges rapidly) [5, 9].
- Distributed (avoids the concentration of the traffic around some nodes or links. No hierarchy and no centralization) [6].
- Allows the construction of homogeneous groups.
- Reduces the diffusions.
- Adapts itself to the changes of topology (minimizes the traffic of maintenance).
- Scalable (without degrading the performances) [9].
- Reduces the energy consumption (increases the autonomy of the mobile units) [8, 9].
- Protects the limited storage capacity of units.
- Exploitable by routing-protocoles.

## 2. Distributed Algorithm of Network Auto-Organization

The algorithm is based on the principle of dividing the network into groups. Each group is represented by a set of nodes and links [3, 4, 10]. Links might be activated or inhibited. Active links form a cover tree for the set of nodes constituting a given group; those links will be activated if necessary during the diffusion of topology packets within the group. The different groups are either linked or not by what we call group-links. The principal objective of this division is to present an utility for the routing process, this is done through the optimization of flooding by orientating the diffusion to reduce the superfluous retransmission, while considering that all diffusions will be done only on active links of a given group, and will not be propagated on group-links only if necessity would impose it (reducing the diffusions => Reduce the energy consumption => increase the autonomy of the mobile units). It should be mentioned here, that the proposed algorithm relays on the hypothesise that all network's links are bidirectional.

### 2.1. Controls' Structure Managed by Nodes

Each network's node uses a certain number of tables and messages during network's construction. Below we will describe these different structures. The network's table indicates all the identified nodes of the network. The group-table contains all the group members to which the node belongs to; an entry is created to each direct neighbour connected with an active link to the concerned node, then it will be completed by the sequence of all group-nodes having been discovered by or through this neighbour; this structuring will be essentially used during the maintenance procedure that we will see later. The neighbours' table lists all direct neighbours of the node and the type of link that connects it to each neighbour; if the node and its neighbour are in the same group, the link created will be either of activated type or inhibited, otherwise, the link will be considered as a group-link as shown in Figure 1.

In principal, nodes use some messages during their interactions. The most important one is the "hello" message. The "hello" message permits to nodes to signal their presence to their neighbours as well as to nodes recently detected, it also permits to detect broken links. The packet contains only the identifier of the transmitter node. Other messages are used by the nodes during the creation or the maintenance of network's links, they will be subsequently described.

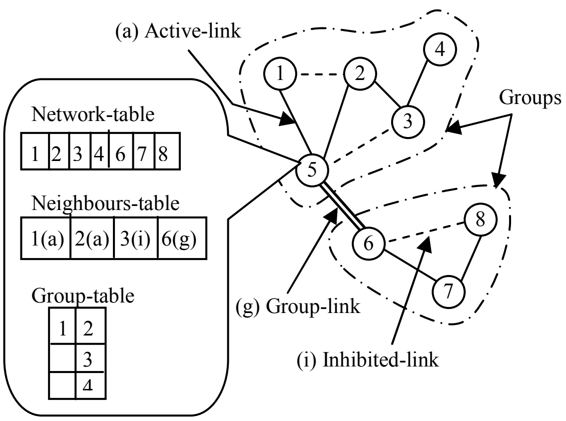


Figure 1. Example of a group, network and neighbours table of a given node.

## 2.2. Dynamic of Network's Construction

The network appears as a graph constituted by a set of trees. A tree is composed by nodes linked by active links; but the existence of inhibited links between tree-nodes is not excluded. Each tree represents a network's group; these groups might be separated, or linked through the group-links. Note here that two given groups of a network might be linked by one or more group-links as shown in Figure 2.

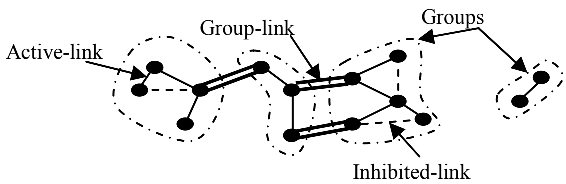


Figure 2. Example of a network.

The construction of groups is made in a completely distributed manner; where each node applies exactly the same procedure in order to be integrated in the network. The principle of building the network is: a node periodically sends a "hello" message to signal its presence. Once it detects the presence of a new neighbour node the two enter in negotiation in order to determine the link's type to establish. Remark here that a node can not effectuate simultaneously several negotiations. Two formats of packet are used during the negotiation: Demand-packet and Confirmation-packet.

General format of a demand-packet

| ID Demand | Hibernation | Link-type |
|---|---|---|

General format of a confirmation-packet

| ID Demand | Link-type |
|---|---|

A node receiving a demand of establishment of link will decide whether to send or not a confirmation according to:

- The state of the field hibernation of the demand (i.e., if the emitter has already executed the procedure of hibernation).
- Its own state towards the procedure of hibernation.
- The type of link of the demand.

The type of link to be established is the one that will be indicated in confirmation message. An active link will be established between two nodes if at least one to them demands the establishment of such a link. A node without any active link will always demand an establishment of an active link during the detection of new nodes; otherwise if the node had at least one active link with another one it will always demand a creation of an inhibited link. If a node has several possibilities to establish an active link it will choose one by random or by using a given metric (e.g., the intensity of signal received from each node). A creation of an active link denotes that a node has been added (or has been re-integrated) to a pre-existent group, or the creation of a new group constituted by two nodes linked by this link.

An isolated node that demands the creation of an active link, without receiving a response from its neighbours' from inhibited links, enter into hibernation during an unpredictable period of time that it calculates. After its awakening, the node repeats its demand, and if still there is no response with inhibited links, it will try to establish an active link with a node that can negotiate. A node that has no active link will prefer to join an existent group rather than to create a new one, this will has as an effect: the reduction of the composition of small groups (construction of homogeneous groups).

When creating an active link, the two concerned nodes update their neighbours', groups' and networks' tables, and proceed to the exchange of these last two. Finally, and if groups' tables are updated, they will be propagated in the concerned group. On the other hand, if modifications are applied on network's tables (detection of a new node in the network); they will be diffused in the whole network.

An inhibited link will be created between two nodes, if only the two concerned nodes demand simultaneously the creation of such a link-type; this same link will be considered as a group-link if the two nodes are not in the same group. Note here that a limitation concerning the number of group-link that one given node can create may be envisaged; this allows to limit the total number of diffusion-links. In all cases, an update of the neighbours' tables of the two concerned nodes is done. If the link is considered as a group-link, an exchange of network's tables will be realized between the two nodes, and if necessary they will propagate modifications in the whole network.

### 2.3. Maintenance of Network's Structure

The frequent changes of topology characterizing Ad-hoc networks may generate some ruptures of links; this brings the necessity to execute a maintenance procedure to preserve the connexion in the network. Here, the procedure of maintaining consists in updating the group-tables of the groups' members concerned by the link failure.

A failure of an inhibited link can not affect the connexion of nodes in a network. But a failure of a group link can generate a rupture between two groups of a network. In both cases (failure of inhibited or group's link), a simple updating of neighbours' tables will be only effectuated on the two nodes concerned by the failure of links, and no control information will be diffused in the network. In contrary, a failure of an active link will cause:

- Either an isolated node. Here, the maintenance procedure described below will be executed in the group concerned with the rupture.
- Or, a node with only inhibited and / or group's links. If the concerned node possesses one or more inhibited links, it will reintegrate its group in transforming one of them into an active link. Otherwise, one of these group links will be chosen, and then transformed into an active link allowing it to integrate in an other group. In the two cases, the maintenance procedure will be only executed within the group concerned by the rupture.
- Or, two groups resulting from division of the group concerned by the link's rupture; inciting for each of the two new generated groups, the execution of the maintenance procedure.

In all cases an update of neighbours' tables will be effectuated on the two nodes concerned by the rupture of link. The maintenance procedure will be executed if necessary in a given group whenever a failure of an active link is detected. This procedure will be initiated by the two nodes directly concerned with the rupture of the link, and will affect only their two groups (hide some change of topology to the other groups => localize the maintenance => minimize the traffic). To have this done, each of the two nodes uses its group-table to determine the set of the members that left the group, namely the nodes previously discovered through the broken link. The two nodes use this information to update their group's table, as well as their neighbours' table by modifying the links type connecting them to the outgoing nodes if they show up in this last; then generate a delete-packet in which they indicate the list of nodes to be deleted from the group, those packets will be diffused to other members by using only active links. Any node that receives the delete-packet will update its group-table and its neighbours-table if necessary (by modifying the type of links connecting it to outgoing nodes if they appear in its neighbours-table), then re-diffuse it through its outgoing active links. If a node receives a suppression packet in which it appears as a node to be deleted, it will propagate just after the diffusion of the suppression packet, an addition packet of a member to the group in which it indicates only its identifier. This last case may happen if the node reintegrates its group after an eventual rupture. A node becoming isolated reinitialize all its control structures. In contrary, if in a meantime a node loses its entire active links but without being isolated, it reinitializes its network and group's tables.

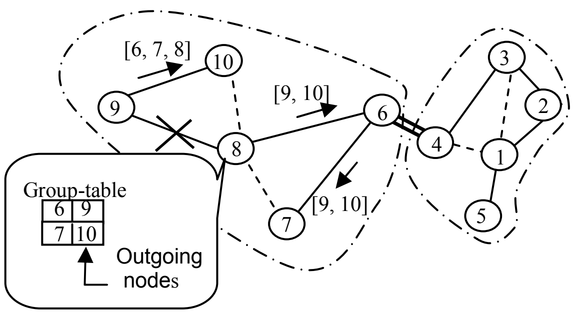


Figure 3. Execution of maintenance procedure.

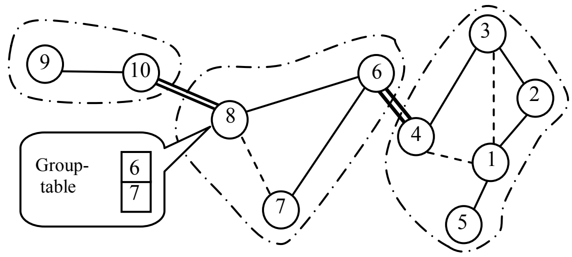


Figure 4. A new topology of the network.

## 3. A Proposed Routing Algorithm

A proposed algorithm is of a reactive type, and it is based on the use of the technique "Source routing", where the source determines the complete sequence of nodes through which the packets of data will be sent to the destination.

The algorithm is inspired from the two protocols DSR and AODV, on which modifications were brought to allow the profit from the architectural construction described above.

### 3.1. The Procedure of Routing Packets

The routing of a packet that disposes of a valid route in its header is done as follows: The emitter inspects the header packet, and looks for one of its direct neighbours, that being the nearest from destination in the sequence of indicated nodes. Once this neighbour is localized; the emitter transmits to it directly the packet after deleting addresses of nodes between it and this neighbour (including even its) from the header of this same packet. This process repeats it self on all intermediate nodes traversed by the packet till it reaches its final destination. Like this, many different paths might be used for the same sequence of nodes indicated in the header packet.

## 3.2. Additional Structures of Controls

Each node in a network manages two other tables dedicated to the routing process: The first is the routes' table, it contains all the routes considered still valid, being discovered during a procedure of route's discovery executed by a given node, or transiting by it. The second table is that of recent demands of route's searching, it is used to detect duplications in the reception of route-requests.; each table's entry contains the identifier of a received request-packet, and the period during which it will be conserved. If the period of a conservation of a given demand is consumed it will be deleted from the table. Here, the demands generated by the node itself will be maintained till the reception of a response-packet concerning them, or after a finite number of retransmissions without success. Retransmissions are separated by periods of waiting fixed by the emitter node.

Two other messages are used by nodes during the process of route's discovery: The RREQ message (route request), and the RREP message (route replay). The message RREQ is diffused by a source-node trying to attain a destination-node; the request contains the identifier of the demand; a field indicating the propagation's mode of the request (i.e., if the diffusion will be done at the group level or the network's one); the two nodes (source and destination); and a recording field of routes in which is accumulated the sequence of nodes visited during the propagation of the request in the network. The message RREP is returned to the source by the destination (or by an intermediate node having at least a valid route to the destination) when this last receives a RREQ concerning it; the packet is constituted of a header indicating the path to take to attain the source-node; the two nodes (source and destination), and the source-route recorded in the RREQ during its diffusion in the network to attain its destination.

General format of an RREQ

| ID RREQ | Propagation-mode | Source | Destination | Recording of route |
|---|---|---|---|---|

General format of an RREP

| Header | Source | Destinatio | Source-route |
|---|---|---|---|

## 3.3. Discovering-Routes Procedure

When a node needs to send packets to an other node in the network and this last is not directly accessible, the node-source looks in its routes' table if a valid route to the suitable destination is available. If one or more routes are found, one from them (the shortest in the number of hops) will be directly used. In the contrary case the emitter must build a route-source and include it in the header of data-packets that will be transmitted. The construction is done by specifying the address of each node through which packets will pass to attain their destination.

The process of discovering routes permits to a source-node to trace a valid route between it and any other destination-node in the network; this is realized through the creation then the propagation of an RREQ by the source-node. If the source-node and the destination-node are in the same group, then the RREQ is propagated first, only on active links; if the source does not receive a response, the request is re-diffused on active and group's links. In the other hand, if the source and the destination are in different groups the request will be directly diffused on the active and group's links at the same time. Passing through the network's nodes, the demand of RREQ is treated as follows:

- If the received demand already exists in the table of requests recently received, the packet will be ignored.
- In the contrary case and if the identifier of the receiving node exists in the field "record of route" of the received request, the packet here too will be ignored.
- Otherwise; if the identifier of the receiving node is the same as the identifier of the destination indicated in the request-packet received, thus the recording of a route (contained in the request) contains the path through which the demand-packet passed before attaining the receptor (destination-node). A copy of this path will be sent within a route's response-packet to the initiator.
- When the receiving-node is not concerned by the demand but it knows about the existence of one or more valid routes to the destination, then it generates a response-packet containing a recording of the route found in the received request to which it will add its identifier and the sequence (or sequences) of nodes relating it to the destination. This packet will be transmitted to the two nodes (source and destination), informing them about the establishment of one or more routes between them.
- Finally; and if the receptor is only a simple intermediate node, it will keep a trace of the received request in its table of recent requests, then adds its identifier in the recording of the route of the received demand, and the packet is re-diffused on

these outgoing links (active and/or of the group) depending on the mode of propagation indicated in this last.

Like this, the request of a route is propagated in the network, till it attains the destination-node. If the discovery operation is a success, the initiator node receives one (or more) response-packet of route that lists the sequence of nodes through which the destination might be attained. Otherwise, and after the consumption of a certain period of time waiting for a response message concerning a given request, a retransmission of that last can be envisaged by the source-node. To return the response-packet of route to the initiator of the route-discovery operation, the destination (or an intermediate node having a valid route to that last), uses the path specified in the recording of the route contained in the request-packet of the route. This path will be eventually inversed (we have supposed that the links between the nodes are bidirectional). If, for the same request many research-packets were having followed different paths are received, many response-packets will be sent on the inverse paths, and the source will thus get many valid paths to attain its target. To reduce the frequency of executing the process of routes' discovery, each node keeps the paths known by the request, response and data packets, passing by it. Those paths will be used till they became invalid. Remark: In the case where the source and the destination are in the same group the source-node might ignore the process of route's discovery, and uses a routing node by node, based on the group's tables of each node. If the transfer fails (reception of a route's error message), the source will execute if necessary the process of route-research.

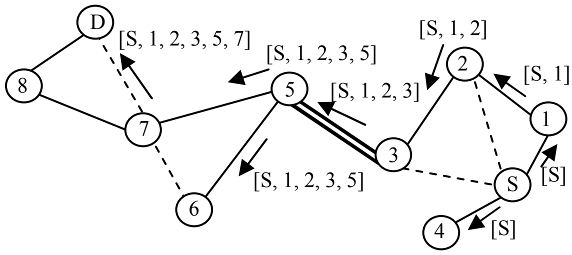


Figure 5. Construction of the source-route.

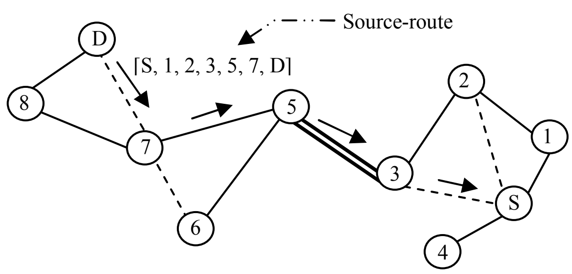


Figure 6. Returning of a response-packet of route.

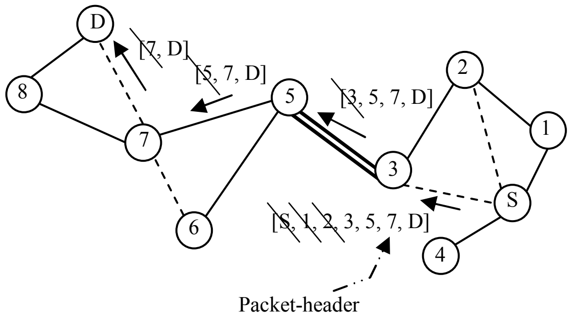


Figure 7. Routing of a data-packet.

### 3.4. The Process of Routes' Maintenance

Links between nodes are maintained through a periodic diffusion of the message "hello". If a given node does not manifest its presence by responding to "hello" messages which were sent to it by its neighbours, those links will be considered broken. Breakdowns in links are generally caused by the moving of the network's nodes. Here, the maintenance procedure is practically the same as the DSR protocol. But it must be noted that a break of a link between two successive nodes in a recording of a given route, will not always need the execution of the maintenance procedure as shown in Figure 8.

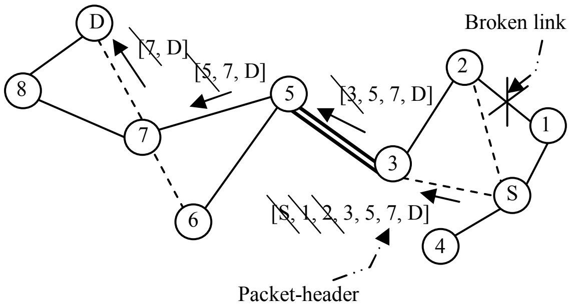


Figure 8. Example of a link's failure that does not need a maintenance procedure.

When a node detects a fatal problem of transmission (a broken link), a message error of route is sent to the original emitter of the packet. The error message contains the identifier of the node that detected the error and that of the node following it in the path (i.e., of the broken node). During the reception of the packet-error of route by the source, the concerned node by the error is deleted from the sequence and the path is truncated at this point. Then if the source has no valid route to the destination, a new operation of discovering routes will be started.

## 4. Evaluation of Performances of Auto-Organization's Structure

We present here the results obtained through the simulations done under a simulator NS2. The results have as an objective to raise the efficiency and the robustness of the proposed auto-organization's structure, and we are particularly interested in the influence of mobility, the scalability, as well as the density of the network on that later.

### 4.1. Parameters of Simulations and Results

The model of propagation used is the free space one. Concerning the model of mobility we choose to use RWM (Random Waypoint Model). The nodes will move in a random limited speed, the pause time is fixed to zero, and thus increasing the mobility of the network. In our simulations the new neighbours detected are (classified trill) in an increasing order according to the intensity of the signal received by the receptor-node before being treated by the later.

A given node may have utmost a single group's link, therefore, a threshold is defined to denote the maximum number of links on which diffusions are permitted.. Remember here that this number is represented by the set of group's links and active one's. We will see that this restriction will not affect the network's connection. For example, for a network of N nodes, the number of active links can not exceed N-1 and the number of group's links should not surpass N/2, therefore the maximum number of diffusions' links is 3/2N-1. The goal of these experimentations is to determine:

- The connectivity of the network, i.e. the number of connected nodes.
- The average number of diffused control messages.
- The number of diffusion-links compared to the total number of links created.

The number of control's messages diffused in the network is directly linked to the frequency of the execution of the maintenance procedure.

Remark here that the retransmissions (much less frequent) and the messages hello (an integral part of the routing process) were not taken into consideration. The radio-range of nodes is 300m; the simulation's duration is set to 300s as well as the time interval between sending two hello messages is set to 3s. The default speed used is 5m/s. This is valid for all the simulations done. The results obtained are as follow:

Table 1. Simulation's environment (group-links/default topology).

| | |
|---|---|
| **Topology** | 800m X 1600m |
| **Nodes' number** | 40 |
| **Movement's maximum speed** | 5m/s |

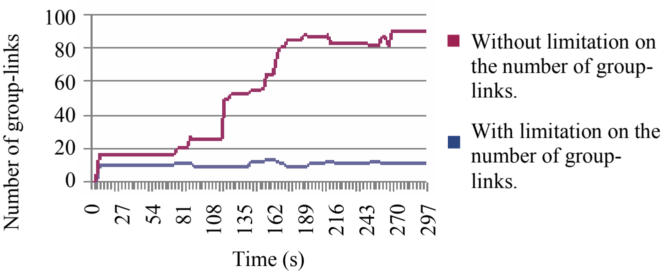


Figure 9. Number of group-links (default topology).

Table 2. Simulation's environment (group-links/scalability).

| | |
|---|---|
| **Topology** | 1500m X 2400m |
| **Nodes' number** | 100 |
| **Movement's maximum speed** | 5m/s |

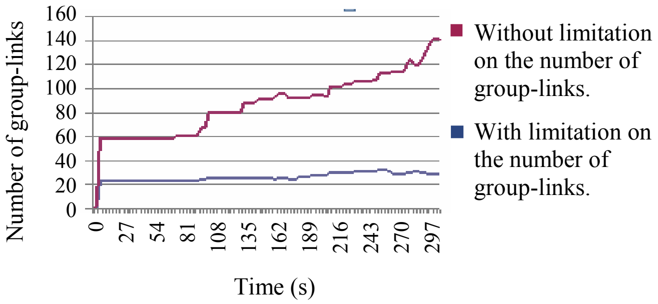


Figure 10. Number of group-links (scalability).

Table 3. Simulation's environment (group-links/density).

| | |
|---|---|
| **Topology** | 800m X 1600m |
| **Nodes' number** | 70 |
| **Movement's maximum speed** | 5m/s |

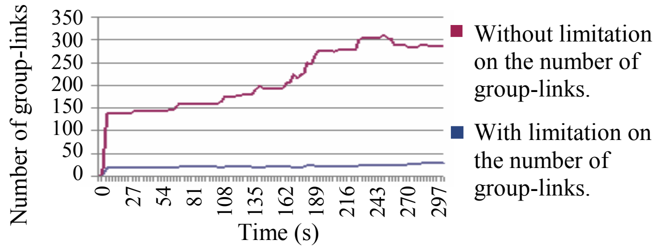


Figure 11. Number of group-links (density).

Table 4. Simulation's environment (impact of mobility).

| | |
|---|---|
| **Topology** | 800m X 1600m |
| **Nodes' number** | 40 |
| **Movement's maximum speed** | 5m/s to 30m/s |

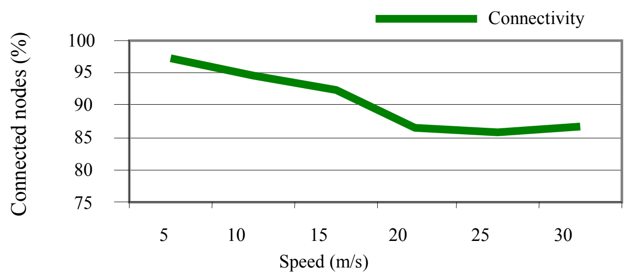


Figure 12. Impact of the mobility on the connectivity.

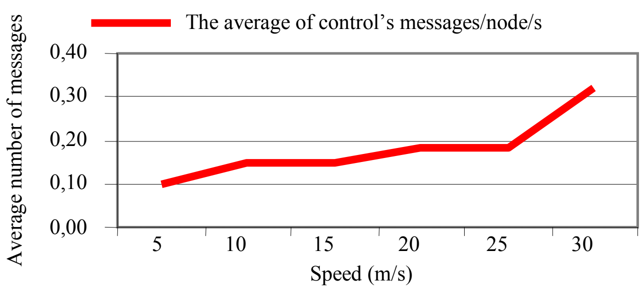


Figure 13. Impact of the mobility on the average of control's messages.

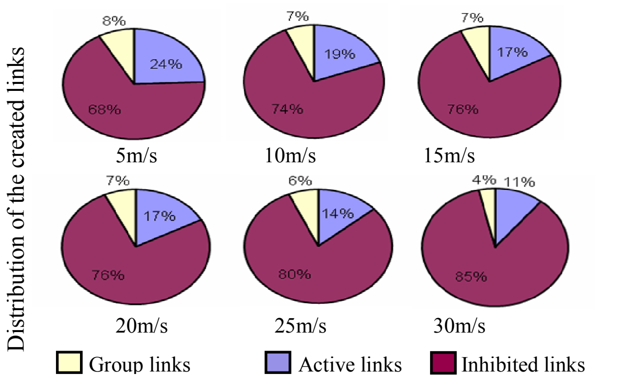


Figure 14. Impact of the mobility on the distribution of the created links.

Table 5. Simulation's environment (Impact of Scalability)

| | |
|---|---|
| **Topology** | 800m X 1600m<br>1200m X 2000m<br>2400m X 1500m |
| **Nodes' number** | 40 / 70 / 100 |
| **Movement's maximum speed** | 5m/s |

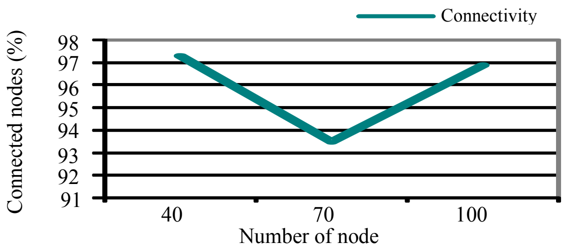


Figure 15. Impact of the scalability on the cconnectivity.

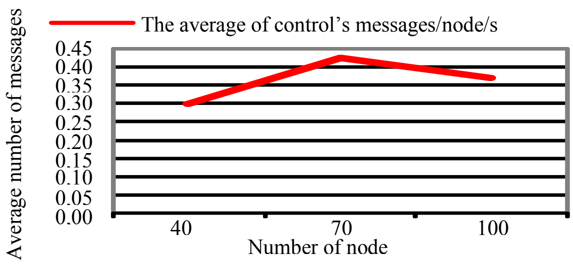


Figure 16. Impact of the scalability on the average of control's messages.

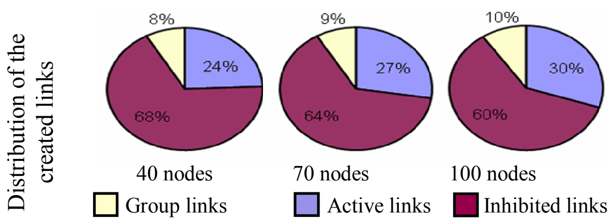


Figure 17. Impact of the scalability on the distribution of the created links.

Table 6. Simulation's environment (impact of density).

| | |
|---|---|
| **Topology** | 800m X 1600m |
| **Nodes' number** | 40 / 70 / 100 |
| **Movement's maximum speed** | 5m/s |

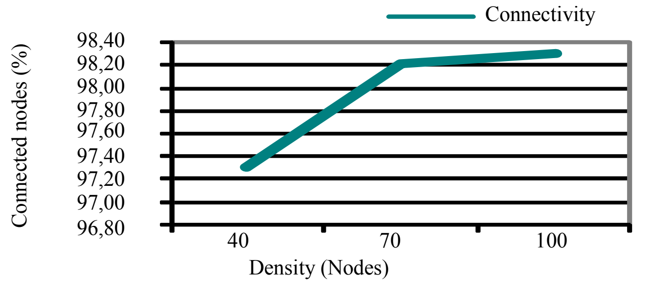


Figure 18. Impact of the density on the connectivity.

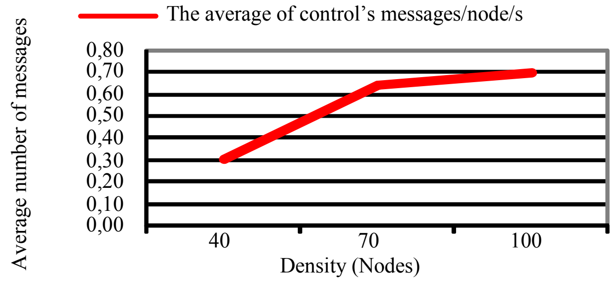


Figure 19. Impact of the density on the average of control's messages.

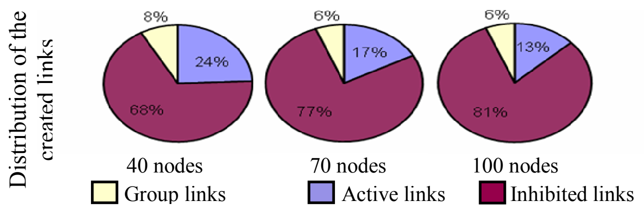


Figure 20. Impact of the density on the distribution of the created links.

## 5. Conclusions

We presented here a scheme of auto-organization permitting the construction and the maintaining of a logical view of an Ad-hoc network. The advantage of this organization lies essentially in:

- The total absence of a centralized control, and its dynamic adaptation to topologic changes.
- Allows reducing the control-traffic.

We evaluated the performances through simulations; the structure reacted well to changes in the network's topology. In a stable situation, it remained connected with more than 92% for slow and medium mobility and more than 85 % of very high speed connectivity (more than 100Km/h). We, also, showed that the proposed algorithm is very well scalable and that it was practically insensitive to density. Finally, the algorithm converges rapidly (between 6s and 8s) and performs well its role by optimizing inundations through the control of diffusion-links' number (60 to 85 % of links created are inhibited) and generates a much reduced traffic. In an other way the rooting protocol has also some advantages that we expose:

- The paths are more exploitable (for a long time)

- Economy of the storage space (some paths are registered in the same record).
- Adaptation to the topology's changes (Not enough maintenance) => Traffic reduced (less energy consumption => more autonomy).

The arguments exposed above permit us to say that the proposed structure can be exploited in an efficient manner by protocols of higher levels especially at the routing level.

## References


[1] AlAgha K., Pujolle G., and Vivia G., *Réseaux de Mobiles et Réseaux sans Fil*, Edition Eyrolles, Paris, 2001.

[2] Andel T. and Yasinsac A., "Adaptive Thread Modeling for Secure Ad-hoc Routing Protocols," *Computer Journal of Electronic Notes in Theoretical Computer Science*, vol. 197, no. 2, pp. 3-14, 2008.

[3] Basagni S., Cont M., Giordano S., and Stojmenovic I., *Mobile Ad-hoc Networking*, New Jersey, 2004.

[4] Camp T., Boleng J., and Davies V., "A Survey of Mobility Models for Ad-hoc Network," *Computer Journal of Wireless Communications Mobile Company*, vol. 5, no. 2, pp. 483-502, 2002.

[5] Dana A., Zadeh A., and Noori S., "Backup Path Set Selection in Ad-hoc Wireless Network Using Link Expiration Time," *Computer Journal of Computer and Electrical Engineering*, vol. 34, no. 6, pp. 503-519, 2008.

[6] Hekmat R., *Ad-hoc Networks: Fundamental, Properties and Network Topologies,* Springer, Netherland, 2006.

[7] Junhai L., Liu X., and Danxi Y., "Research on Multicast Routing Protocols for Mobile Ad-hoc Networks," *Computer Journal of Computer Networks*, vol. 52, no. 5, pp. 988-997, 2008.

[8] Pujolle G., *Cours Réseaux et Télécoms*, Edition Eyrolles, Paris, 2004.

[9] Pujolle G., *Les Réseaux*, Edition Eyrolles, Paris 2004.

[10] Vok L., "A Neighbor-Table-Based Multipath Routing in Ad-hoc Networks," *in Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference*, pp. 1739-1743, 2003.


**Maâmar Kef** received his Master's degree in computer sciences from the University of Larbi Ben Mhidi, Oum El Bouaghi, Algeria, in 2008. From 2004 he was IT manager at a National company. Since 2008, he has been at the University of Constantine, where he is preparing his PhD degree. His current research interests include ad-hoc networks, artificial intelligence techniques with neural networks, pattern recognition and document processing.

**Leila Chergui** received her Master's degree in computer sciences from the University of Larbi Ben Mhidi, Oum El Bouaghi, Algeria, in 2009. Her current research interests include ad-hoc network, neural networks, pattern recognition, and Document Processing.

**Mohamed Benmohammed** received his BSc degree from the High School of Computer Science (CERI) Algiers, Algeria, in 1983, and the PhD degree in computer science from the University of Sidi Belabbes, Algeria, in 1997. He is currently an assistant professor at Constantine University. His current research interests are parallel architectures and high level synthesis.