Qos Routing Scheme and Route Repair in WSN

[1,2]M. Belghachi
[1]Faculty of Science and Technology
University of Bechar,
Bechar, Algeria

[2]M. Feham
[2]STIC Laboratory
Faculty of Technology, University of Tlemcen,
Tlemcen, Algeria

***Abstract*— During the last decade, a new type of wireless network has evoked great interest among the scientific community; it is the wireless sensor networks (WSN). The WSN are used in various social activities, such as industrial processes, military surveillance applications, observation and monitoring of habitat, etc... This diversity of applications brings these networks to support different types of traffic and to provide services that must be both generic and adaptive for applications, the properties of the quality of service (QoS) are different from one application to another. However, the need to minimize energy consumption has been the most important field of WSNs research. Few studies in the field are concerned with mechanisms for efficiently delivering QoS at the application level from network level metrics and connection such as delay or bandwidth, while minimizing the energy consumption of sensor nodes that are part of network. The idea is to ensure QoS through a routing process, which can detect paths that meet the QoS requirements based on ant colony optimization (ACOs), coupled with detected routes reservation process. However, it is necessary to integrate to this diagram the maintenance of route disrupted during communication. We propose a method that aims to improve the probability of success of a local route repair. This method based on the density of nodes in the vicinity of a route, as well as on the availability of this vicinity. Taking into account these parameters in the route selection phase (end of the routing process) allows selecting among multiple routes, the one which is potentially the most easily repairable. In addition, we propose a method for early detection of the failure of a local route repair. This method can directly trigger a process of global re-routing that better fits to restore communication between the source and destination.**

Keywords- WSNs; Quality of service; ACO; availability; global re-routing; local re-routing.

I. INTRODUCTION

The wireless sensor networks (WSNs) are considered a special type of ad hoc networks. The nodes of such networks consist of a large number of sensor nodes, which can collect and transmit environmental data in an autonomous manner [1, 2]. The position of these nodes is not necessarily predetermined. They are scattered randomly across a geographical area called acquisition field, which defines the field of interest for the phenomenon captured. The data collected are forwarded to a node considered as a collection point, better known as the sink node. WSNs must be dynamically adaptive to continuous changes of the state of the radio channel and must at the same time attempt to satisfy the QoS requirements of the supported applications.

II. QOS ROUTING IN WSNS

Routing is often considered to seek the shortest route in terms of distance between source and destination to transfer data. In the case of routing with quality of service, the goal is not just to find the best path according to some criterion, but to find the best eligible path as well [3]. For this reason, a number of constraints are imposed on the routes in order to determine their eligibility. For example, you may want to search for a route with a certain amount of bandwidth for video traffic or route ensuring that packets are received by the destination within a certain period of time after their issuance by the source. Any route satisfying a quantitative criterion can be described as route ensuring a certain quality of service. WSNs mainly work with the client-server mode. All nodes then attempt to send the captured information to a processing center. This technique is highly energy costly because the information must traverse the entire network to reach the processing center.

III. ACOS AND WSN ROUTING

A new routing approach in WSNs has emerged: this approach is based on algorithms inspired from ant colonies [4, 5]. These algorithms are based on the ability of simple ants to solve complex problems through cooperation. All methods using this paradigm are now called ACOs. Indeed, collective intelligence in social insects results in the emergence of collective behavior due to macroscopic smart simple interactions at the microscopic level. Operation of ant colonies is the best example [6]. The behavior of ants is a collaborative and collective one. Each ant has priority for the welfare of the community. Each individual of the colony is a priori independent and is not supervised (completely distributed system). The colony is self-controlled through relatively simple mechanisms to study. By projecting the behavior of these insects on the characteristics of WSNs, we note that the behavior of ants is well suited to this type of networks, especially when calculating routes.

IV. DESCRIPTION OF THE PROPOSED APPROACH

The idea is to design a decentralized algorithm based on operation of ants [7, 8], which use their natural ability to find the shortest path between a source and destination while moving through the network. Our approach uses the same mechanisms for selecting local vicinity of Ant System [9, 10].

So, the formula to choose next neighbor with joint attraction pheromone trails. The proposed approach consists of two phases, namely route discovery and route maintenance. When a source node needs to send data to the base station with

QoS requirements (bandwidth), it begins with the route discovery phase. Once the route is found, data transfer can begin. During data transmission, it is also necessary to maintain the path to the destination. One of the weaknesses of these networks lies in the fact that the routes used between source and destination is likely to break surreptitiously during communication. This failure is due to the fact that the nodes forming the route may distrust (deplete their energy). In case the link to the next node is broken, the node initiates phase of route repair. This phase is based on global or local re-routing. The objective of the modification is to ensure the selection of the route more easily repairable among those released during the discovery phase of routes. To achieve this goal, we take into account the character of the nodes vicinity in the network, and in particular the density of nodes, as well as their availability. Repair of route in case of a node failure occurs by the implementation of procedure for local re-routing, and thus avoiding global re-routing that consumes bandwidth and execution time, thereby increasing communication delays.

A. *Interpretation of the availability*

We define the availability of a node by the number of neighbors (ie, the number of nodes in the area of its radio range) whose bandwidth is greater than that required by the connection. The availability parameter is specific to a node and a connection (a given bandwidth). Thus, upon receiving a route request, each node assesses its availability by counting the number of neighbors whose bandwidth is greater than that required by the connection [11]. If the availability is equal to one, no neighbors other than the transmitter of the request meet the QoS requirements of the connection. Continue broadcasting the message of connection is useless because no node in the vicinity is able to extend the route to the destination. This leads to early detection of imminent failure of the route discovery through this node, while reducing network congestion. If the availability is equal to two, one neighbor other than the transmitter of the request meets the QoS of the connection. We can continue to broadcast request message. However, no node will take over from the current node fails in case the route would be used for communication. This availability is too low to ensure a local re-routing of the current node. This is called throttling.

B. *Control packets*

to implement our proposed approach, four control packets are used [12, 13].

1) Hello_Ant packet: The Hello_Ant packet is distributed periodically to all neighbors of the current node, containing the delay of its departure. When neighbors receive this packet, they react by responding by an acquired reception (ACK_Ant). Based on the delay of departure, delay of arrival and the Hello_Ant packet size, the current node calculates the available bandwidth on the links. An entry in the neighbor table is created containing the value of the available bandwidth and the residual energy of all its neighbors. For the update of the value of the bandwidth and the residual energy is produced by subsequent Hello_Ant packets to indicate the current status of links.

2) Route request packet: A Route_Request_Ant packet is broadcast on receiving a route request to a destination with a demand for quality of service expressed in terms of bandwidth. At each node, the hop count is incremented and the node ID is added to the stack of visited nodes.In addition to the exploration of the shortest path between the source and the destination, Route_Request_Ant collect the end-to-end delay, the minimal available bandwidth, the average availability and the minimal residual energy of the path through which it is propagated.

3) *Route reply packet*: upon receiving Route_Request_Ant, the destination creates the response message Route_Reply_Ant. Route_Request_Ant packet collects the transmission delay of each link, the processing time at each node, the bandwidth available on each link, residual energy and hop count. This Route_Reply_Ant will be sent (unicast) to the original source along the route established by route_Request_Ant in reverse.

4) Route error packet: This Route_error packet is sent to the source of the communication to indicate that the route to a destination is broken and that the attempt of local re- routing was unsuccessful. Local repair is attempted only on condition that the availability of node is strictly greater than two. When the source receives this packet, it invalidates the status corresponding to the destination, stops sending packets to the destination and places them in a queue. Meanwhile, the source initiates a procedure of global re-routing.

C. *Mathematical Model*

The objective function of the proposed work is to find a path from source to destination through a neighbor with a maximum transition probability. The probability of transition from source i to destination d through neighbor j of i is calculated as follows [14, 15]:

$$P_{ijd} = \frac{[Dispo_{ijd}]^{\alpha}[D_{ijd}]^{\beta}[\eta_{ijd}]^{\chi}[BP_{ijd}]^{\delta}}{\sum_{l \in N_i}[Dispo_{il}]^{\alpha}[D_{ild}]^{\beta}[\eta_{ild}]^{\chi}[BP_{ild}]^{\delta}} \qquad (1)$$

Where α, β, γ and δ (> = 0) are parameters that control the relative importance between availability, delay, residual energy / hop count and available bandwidth. Ni is the set of neighbors of i and l is neighbors of i through which a route is available to the destination. For the calculation of relatives' metrics, the delay and the number of hops are additives metric; bandwidth and the residual energy are considered as non-additive concave metric. Additive metrics must be reduced to a minimum for the shortest paths; the non-additive concave metric is used to maximize bandwidth and residual energy [16, 17].

1) *Availability:*

The problem is to determine the extent to which a node is part of a route between a source and destination and immediately adjacent to another node (i.e. its radio range) and therefore can be replaced by the latter in case of failure. It is particularly important that the replacement node have enough bandwidth communication channels for this new connection.

$$Dispo_{ijd} = Moy\{Availibility(l)\} \quad \forall\, l \in route_j(i,d) \qquad (2)$$

Dispoijd: the average number of neighbors whose bandwidth is greater than that required by the connection along the path from i to d through j.

2) Delay :

The delay between the source and the destination is calculated by:

$$D_{ijd} = \sum_{l \in route_j(i,d)} delay(l) \quad (3)$$

Where the delay (l) is the end to end delay from source i to destination d through the neighbor j by route request message at the time of route exploration.

3) hops count / Minimum Residual Energy :

This relative metric equal the inverse of number of hops multiplied by the minimum Residual battery Energy of all intermediate nodes between source and destination is given by:

$$MBR_{ijd} = \min\{\mathrm{Re}sidual\_Energy(l)\} \quad \forall l \in route_j(i,d) \quad (4)$$

$$\eta_{ijd} = \frac{MBR_{ijd}}{NbSaut(route_j(i,d))} \quad (5)$$

Where

- MBRijd: minimum residual energy along the path i to d through j
- NbSaut (routej (i, d)) is the number of hops along the path i to d through j.

4) The bandwidth:

Available bandwidth on the path from i to d is calculated as the minimum of available bandwidth (Bpijd) of all links along that path.

$$Bp_{ijd} = \min\{Available\_bandwith(l)\} \quad \forall l \in route_j(i,d) \quad (6)$$

Hello_Ant messages are often transmitted to keep the connectivity of the vicinity, and they can better reflect the current available bandwidth on the links rather than the route search messages.

D. Routes discovery phase

The source initiates the routing process. It sends to all its neighbors a connection request (Route_Request_Ant) to the destination with QoS requirements in terms of bandwidth. Nodes that receive the message for the first time and meet the QoS requirements broadcast a demand to their vicinity after collecting:

a) The transmission delay of each link

b) The available bandwidth on each link

c) The number of hop

d) The residual energy

This connection message is thus spread in the network until it reaches the destination. When Route_Request_Ant reaches the destination, it will be converted into Route_Reply_Ant and transmitted to the origin source. Route_Reply_Ant takes the same path marked by Route_Request_Ant in reverse. For each Route_Reply_Ant, when reaching an intermediate node or source node, the node just find the delay, bandwidth, residual energy and the number of hops. The node can calculate the probability to reach the destination. The path with the highest probability is considered as the best, and the data transmission can begin.

E. Route maintenance phase

Because of the failure of nodes responsible for the transmission of data between source and destination, the risk that the route be disrupted before the end of the communication is very high. In case of link-breaking or node failure during data transmission, there are two scenarios for the re-routing [11, 18]:

1) Global re-routing from the source of the communication. This re-routing is implemented in most routing protocols, although it takes a lot of time and consumes a lot of bandwidth.

2) Local re-routing from the node where the failure occurred. This local routing has the advantage of being fast and consume less bandwidth. Local repair is attempted only on condition that the availability is strictly greater than two. Otherwise, a Route error is sent directly towards the source of the communication to undertake a global repair of the route. We bring this change to avoid loss of time expected from a routing attempt to local route repair anyway doomed to failure (if availability is less than three) and thus accelerate the re-routing phase despite hostile conditions.

F. Experimental design

In our approach, we seek to exploit the heterogeneity of nodes distribution in the network by taking into account the availability parameter. It would be pointless to base our tests on a random configuration improvement that is statistically homogeneous. Nevertheless, we conducted several simulations in such a configuration to conclude that our contribution adds little performance compared to AODV protocol. To perform our tests, we chose a concentrate on a heterogeneous configuration of data, and then to change the departure of nodes. In each simulation, we consider the following cases:

a) Case 1: AODV protocol.

b) Case 2: Our approach taking into account the availability parameter and re-routing based on local density.

Different results are presented in the following section and correspond to averages over a series of simulations. In all scenarios the nodes are static, and randomly deployed. Parameter values of relative weight α, β, γ, δ and δ are defined in Table I

Table I. Parameter settings

Parameters	**Values**
Availability weight, α	1.0
Delay weight, β	1.0
Residual Energy/ hop count weight, γ	1.0
Available bandwidth weight, δ	1.0
Interval Hello_Ant	1 sec
Hello_Ant Retry times	3

G. Tests and analysis of results

In the configuration shown in Figure 1, node 2 tries to establish communication with the Sink, thus triggering the route research phase to the destination by broadcasting a Route_Request_Ant packet with QoS requirements in terms of bandwidth. Several types of routes are possible. Among these, the route containing nodes 3, 4, 5 and 6 is the shortest (Case 1). To test our hypotheses, we plan the "disappearance" of a node at a specific time. Then we focus our interest in the evolution of the system and in particular the way to repair the route. We decide to envisage the disappearance of nodes 6, 13 and 18. This choice leans on the fact that the study of the disappearances of the other nodes is impossible (disappearance of nodes 2 and/or Sink which would hinder definitively the communication).

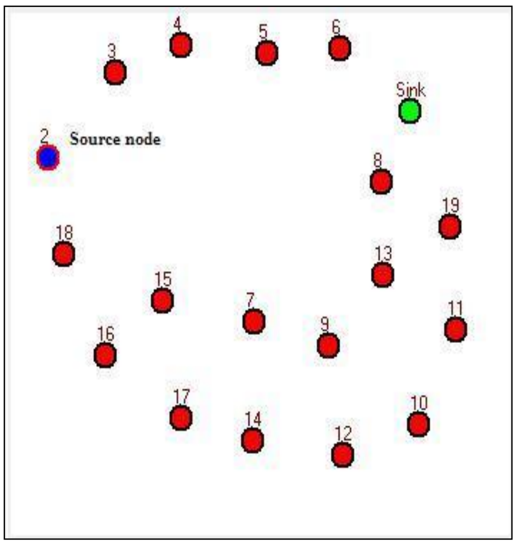

Figure 1. Spatial distribution of nodes

So, the node 7 receives a Route_Request_Ant packet for the destination passed through the nodes 18 and 15. The node 7 adds in its database an entry for the destination by indicating that the next node for the destination is 15. Then, the node 7 updates the values of average availability, End to End delay, Residual energy and hop count. The Route_Request_Ant packet can be then rebroadcasted.

The first packet which arrives at the destination is the packet passed through the nodes 3, 4, 5 and 6. A time-out whose value depends on the delay taken by the packet to propagate until the destination is immediately launched. In this interval, the destination stores each Route_Request_Ant packet that arrives. So, another Route_Request_Ant has passed through the nodes 18, 15, 7, 9, 13 and 8 (Case 2) is treated by the destination. The length of the detected route is equal to 7.

1) Disappearance of node 6:

Table II summarizes the results obtained during the simulations. In case1, we notice that time and the data are strongly influenced by the disappearance of node 6. The regular delay that averages 9.5ms before the disappearance of node 6 rises to 14 ms and the route used extends in terms of number of nodes. These observations can be explained by the fact that this node is part of the route used to transport data to the destination when it disappears. Node 5 notices that the node 6 has disappeared and immediately returns a route errors message to the source node. This triggers an operation of global re-routing. As shown in Table II, the re-routing phase lasts longer than the initial phase (18ms against 9.2ms).

In case 2, the packets follow a path in which node 6 is not part of the route used. So we do not see any change. Delays remain unchanged before and after the departure of node 6. It establishes at 14ms. By contrast, we note that the initial route research delay is much longer than in the previous case. We explain this observation by the fact that we expect more Route_Request_Ant at the destination before returning Route_Reply_Ant to destination on the selected route.

Table II. Summary of results when node 6 disappears

	Case 1	Case 2
Initial routing delay (millisecond)	9.2	19.729
local repair delay (millisecond)	0	0
total re-routing delay (millisecond)	18.1667	0
Average delay before re-routing (millisecond	9.5	14
Average delay after re-routing (millisecond)	16	14
Average length of routes (node)	6.563	7

2) Disappearance of node 13:

The data collected for case 1 indicates that no re-routing is undertaken following the departure of node 13. We could predict it because the node 13 is not part of the route used in case 1. In case 2 however, we notice that the results are strongly affected. Thus, the delay increases from 14ms before the disappearance of node 13 to 16ms. We also observe in Table III that local re-routing was undertaken successfully. We note that the local re-routing lasts 7.82ms. This re-routing delay is compared to the 18ms for global re-routing in case 1 obtained for the disappearance of node 6.

Table III. Summary of results when node 13 disappears

	Case 1	Case 2
Initial routing delay (millisecond)	8.55	22.013
local repair delay (millisecond)	0	7.82
total re-routing delay (millisecond)	0	7.82
Average delay before re-routing (millisecond)	9.5	14
Average delay after re-routing (millisecond)	9.5	16
Average length of routes (node)	5	8.261

3) Disappearance of node 18:

As for the disappearance of node 13, only case 2 is affected by the disappearance of node 18. Moreover, we observe in Table IV. That the delay decreases 14ms before the disappearance of node to 9.5ms after its disappearance. These results for case 2 are to be compared with those obtained for case 1 in the disappearance of node 3. We note that the length of the search path following the disappearance of node 18 is 39ms. This delay is much less than the re-routing duration in case 1 measured for the departure of node 3.

Table IV. Summary of results when node 18 disappears

	Case 1	Case 2
Initial routing delay (millisecond)	8.55	19.756
local repair delay (millisecond)	0	39.012
total re-routing delay (millisecond)	0	30.12
Average delay before re-routing (millisecond)	9.5	14
Average delay after re-routing (millisecond)	9.5	9.5
Average length of routes (node)	5	5.937

4) Assessment of node disappearance:

We seek to extend our results to the disappearance of nodes 3, 6, 13 and 18 to all network nodes. Thus, we consider the disappearance of all nodes belonging to an initial route. We do not envisage the disappearance of node 2 and/or Sink (their disappearance would be unrecoverable for communication) as for nodes 10, 11, 12, 14, 17, 16 or 19. The disappearance of the latter in fact has a limited impact in the system. We assume that the disappearance of each node in the system is equi-probable. We classify nodes into several groups according to the case to which we refer. We obtain the following distributions:

- Case 1: The disappearance of node 13 is representative of the disappearance of nodes as follows: 7, 8, 9, 13, 15 and 18. The disappearance of node 6 is representative of the disappearance of nodes 4, 5 and 6.
- Case 2: The disappearance of node 6 is representative of the disappearance of nodes: 3, 4, 5 and 6. The disappearance of node 14 is representative of the disappearance of nodes 7, 8, 9, 13 and 15.

Using these distributions, we weight the different scenarios for the disappearance synthetic results presented in Table V

Table V. Results of Case1 and Case2 comparison

	Cas 1	**Cas 2**
Initial routing delay (millisecond)	8.78	20.5
total re-routing delay (millisecond)	39	5.8
Average delay before re-routing (millisecond	9.5	14
Average delay after re-routing (millisecond)	11.9	14.6
Average length of routes (node)	5.63	7.52

We note that the initial routing delay is always longer for case 2. This is the main drawback inherent in our improvement. However, waiting longer at the route establishment phase is necessary to choose a better repairable route. We also note that the average re-routing caused by the departure of a node in case 2 is significantly better than case 1. Thus, the waiting time is on average 6.7 times shorter than in case 1. This observation attests to the significant improvement of the QoS management offered by the protocol improved compared to the original version. A weakness of our improvement is observed in the average length of the route. We find that the route is on average equal to 7.52 in Case 2 against about 5.5 for the other case. The route thus involves more nodes and, therefore, the probability that a failure occur increases on the route retained by case 2. Another weakness of our improvement is that the end to end delay of data packets is more important in our version compared to the original version. In case 2, the end to end delay is established before the disappearance 9.5ms against 14ms for both cases. By contrast, we find that, this delay increases shortly after the disappearance and then settled at 14.6ms against 11.9ms for case 1. The end to end delay obtained in case 2 is more stable than that observed in the other case. We conclude that this level, once again our improvement provides a gain in terms of the QoS management in case of node disappearance.

V. Conclusion

The work presented in this paper aims to improve the QoS management by taking into account the availability defined as the number of available nodes in the radio range of a node. Our approach was based on a thorough analysis of the tools taking into account the QoS in WSN. We then devised the concept of availability and describe how a node can use this information to improve QoS. We have identified two main areas:

- Establishment of a mechanism for route choice.
- Prediction of early failure of a local re-routing.

The mechanism of route choice is to select among several competitors whose maintenance is the easiest to achieve. Simulation results confirm our theoretical reasoning: the delay of re-routing is improved. In addition, our mechanism also improves the rate of packet loss. QoS is half-open thus further strengthened. We have highlighted a situation in which a local re-routing attempt fails: if the availability is too low around the node that initiates local re-routing, it is doomed to failure. It is therefore preferable to return directly to the source a Route error packet after detecting a link failure. The main limitation of our route selection process is the loss of time in the initial routing. This loss of time is related to the expectation of receiving other Route_Request_Ant from the source before selecting the best route to facilitate route repair. Another limitation stems from the fact that the routes adopted after our improvement are longer. The end to end delay is, therefore more important. Another line of work would be to conduct a thorough performance study for distributions of nodes inspired from plausible situations in real environments.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless Sensor Networks: A Survey,” Computer Networks, Vol. 38, No. 4, March 2002, pp. 393-422.

[2] H. Karl and A. Willing, “A Short Survey of Wireless Sensor Networks,” Technical Report TKN-03-018, Telecommunication Networks Group, Technical University, Berlin, October 2003.

[3] D. Chen and P. K. Varshney, “QoS Support in Wireless Sensor Network: A Survey,” Proceedings of the 2004 International Conference on Wireless Networks (ICWN2004), Las Vegas, Nevada, USA, June 2004.

[4] M. Dorigo, “Optimization, learning and natural algorithms (initalian),” Ph.D. dissertation, 1992.

[5] R. Montemanni and L. Gambardella, “Swarm approach for a connectivity problem in wireless networks,” in Proceedings 2005 IEEE Swarm Intelligence Symposium, pp. 265—272, 2005.

[6] M. Dorigo and T. Stuzle, Ant Colony Optimization. MIT Press, 2004.

[7] R. Montemanni and L. Gambardella, “Swarm approach for a connectivity problem in wireless networks,” in Proceedings 2005 IEEE Swarm Intelligence Symposium, pp. 265—272, 2005.

[8] W. Agassounon, “Distributed information retrieval and dissemination in swarm-based networks of mobile, autonomous agents,” in Proceedings of IEEE Swarm Intelligence Symposium, pp. 152—159, 2003.

[9] O. Cordon, F. Herrera and T. Stuzle, “A review on the ant colony optimization metaheuristic: Basis, models and new trends,” Mathware and software computing, no. 9, pp. 1—35, 2002.

[10] P. Ji, et al. “DAST: A QoS-Aware Routing Protocol for Wireless Sensor Networks,” Proceeding of International Conferences on Embedded Software and Systems Symposia, Sichuan, 29-31 July 2008, pp. 259-264.

[11] M.Belghachi, M.Feham,“QoS Based on Ant Colony Routing for Wireless Sensor Networks,” International Journal of Computer Science and Telecommunications, January 2012.

[12] M.Belghachi, M.Feham, “Routing diagram for the transport of video traffic in a WSN,” Journal of Theoretical and Applied Information Technology, July 2010.

[13] P Deepalakshmi, S Radhakrishnan, Ant colony based QoS routing algorithm for mobile ad hoc networks. Int J Recent Trends Eng. 1(1), 459–462 (2009).

[14] M Gunes, U Sorges, I Bouazzi, ARA–the ant-colony based routing algorithm for MANETs, in Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'02), Vancouver, BC (2002).

[15] O Hussein, T Saadawi, Ant routing algorithm for mobile ad hoc networks (ARAMA), in Proceedings of the International Performance Computing and Communications Conference (IPCCC 2004), Phoenix, Arizona (2004).

[16] G Di Caro, F Ducatelle, LM Gambardella, AntHocNet: an adaptive nature inspired algorithm for routing in mobile ad hoc networks. Eur Trans Telecommun. 16(2), 443–455 (2005).

[17] Z Liu, MZ Kwiatkowska, C Constantinou, A biologically inspired QoS routing algorithm for mobile ad hoc networks. Int J Wirel Mobile Comput. 4(2), 64–75 (2010).

[18] B. Macabéo " Routage et reparation de routes dans les réseaux mobiles Ad hoc", Maitrise en genie électrique Avril 2003.