DSR based routing algorithm with delay guarantee for Ad Hoc networks

Sofiane Ouni
CRISTAL laboratory, Ecole Nationale des Sciences de l'Informatique (ENSI) &
Institut National des Sciences Appliquées et de Technologie (INSAT), Tunisia,
E-mail: Sofiane.ouni@insat.rnu.tn

Jihen Bokri and Farouk Kamoun
CRISTAL laboratory, Ecole Nationale des Sciences de l'Informatique,Tunisia,
E-mail: bokri.jihen@gmail.com, frk.kamoun@planet.tn

Abstract— **Applying real time applications with deadline guarantee for Ad Hoc network seems to be a difficult challenge and to be an impossibility in general. This is due to the features of such networks. However, if we make an assumption on node mobility, traffics and reliability with predictable features, we can get real time guarantee. That is the goal of our present work.**

In this paper, we present a new routing algorithm based on DSR to get the guarantee of Hard real Time traffics. This algorithm is based on a node resource reservation for each traffic path. An admission control is made to preserve the reserved node resources and is adapted to node mobility by preventing path breaking. Our routing protocol is implemented under NS2 and simulation results perform more stable time features than classical routing.

Index Terms—**Hard Real Time, deadline guarantee, expiration delay, communication delay, Ad Hoc routing, Quality of Service.**

I. INTRODUCTION

Ad Hoc networks have been deployed in many military and industrial applications [26], because of their facilities such as wireless communication, easy deployment and mobility. Such networks don't need any infrastructure and each device (or node), joining the wireless network, can work as a sender, receiver or router. Also, these devices can move, making topology changes, without impairing communications. Appropriate routing protocols can maintain this communication between the sender and the receiver.

Due to the Ad hoc networks nature, finding efficient routing protocol is not an easy task. Routing protocols in wired networks are no longer suitable. In fact, dedicated routing protocols are proposed in several works. They are classified in two types: proactive and reactive.

The proactive protocols, like OLSR (Optimized Link State Routing Protocol) [11], always maintain routes through which traffics can be transmitted. This is done by periodically updating the routing tables to all possible destinations. However, reactive protocols discover routes only when demanded by broadcasting a route request packet through the network. The discovered path is maintained until it's no longer desired. DSR (Dynamic Source Routing) [23] is a reactive protocol. The source determines the needed route by transmitting a request packet to destinations and waits for a route reply to have a selected path. Once the route is chosen, traffic will follow the discovered path.

Nevertheless, the above protocols can only provide paths between sources and destinations regardless of the Quality of Service (QoS) requirements such as bandwidth and delay guarantees. These demands are very frequent especially for real time applications. For that reasons, so many studies [8][6] focused on routing with QoS for Ad hoc Networks. Some routing protocols were designed for QoS; others were extended to support QoS.

The QoS routing protocols can be based on IntServ (Integrated Services) or DiffServ (Differentiated Services) architectures. The IntServ (Integrated Services) approach adopts a path's node reservation principle to maintain the QoS like BRuIT (Bandwidth Reservation Under Interferences influence) protocol [8], whereas, the DiffServ method consists on classifying traffic transferred into Ad hoc networks. Thus, the higher priority is given to real time traffics in order to allow them to reach their destinations in the shortest possible time, like MQDR (Multipath QoS Routing protocol of supporting DiffServ) protocol [6].

These works are not efficient for hard real time traffic. Indeed, in this context, the QoS must not be simply handled in most cases, but it must also be guaranteed. The missing of real time constraints, such us deadline on treatment, will lead to an undesired situation and in some cases to a system crash. That's why deadline constraints on traffic transmissions have to be met. This is the goal of our research. So, in this paper, we provide a real time solution which allows to route real time packets with respect to their deadlines.

The guarantee of deadline constraints is a big challenge because of the Ad hoc network's characteristics such as the lack of central coordination and the mobility. In fact, because of the lack of centralized network management, it's difficult to guarantee that packets of real time traffics will be transmitted to their destination before the deadline. Furthermore, new coming traffics can disturb previous real time ones, and prevent packets to reach their destinations before the deadline. As a remedy to this issue, we introduce an admission control that can maintain the guarantee of the accepted traffics. Hence, a new traffic is accepted only if it can be transmitted with respect to its deadline without disturbing the previous guarantees. Moreover, the admission control is made in collaboration with all nodes of the selected traffic path instead of independent admission.

Another big issue is the mobility. The challenge is to find and maintain a path to the destination even in the case of some links breakage. So, we must predict these breakages in advance and find another path with respect to the real time constraints. That's why, we propose an analysis to predict the instants of topology changes and verify the real time constraints.

From the above specified admission control and mobility prediction, we make a new protocol called RT-DSR (Real Time DSR) based on the reactive protocol DSR. With this new extension to DSR, we can react properly to make paths that guarantee real time requirements in spite of Ad hoc networks characteristics. Our proposed protocol is implemented in NS network simulator to prove its efficiency for real time traffics.

The remainder of this paper is organized as follows. In section 2, we study different solutions for QoS in Ad hoc networks. The routing protocol DSR is presented in section 3. In section 4, we describe our proposal based on Expiration delay. Section 5 discusses the mobility issue. We evaluate our proposal with simulations in section 6. Finally, we conclude the paper in section 7.

II. QOS IN AD HOC NETWORKS

Several types of network applications have some demands on the network, such as the delay (end-to-end time), the jitter (deviation of latency), the bandwidth and the reliability. The Quality of Service (QoS) consists in providing the required demands to such applications.

To provide QoS to some flows, two architectures were defined which are IntServ and DiffServ. IntServ (Integrated Services) must be implemented by all the routers in the network, and every application that requires some guarantees has to make an individual reservation. IntServ can be based on the RSVP (Resource ReSerVation Protocol) mechanism which is used by the router to decide if it can support the requested reservation. While the DiffServ (Differentiated services) [12] operates on the principle of traffic classification. In fact, each data packet belongs to one or more classes and is treated differently by routers according to the priority of their classes.

Using IntServ and DiffServ, different approaches of QoS are proposed in literature. Some researchers extended existing Ad hoc routing protocols to support QoS. Others tried to design an independent protocol for QoS. Also, in some researches, authors provide QoS Ad hoc routing protocols specifically for real time applications.

A. Routing protocols extended to support QoS

To provide QoS solutions for ad hoc networks, some researches [7] have enhanced the existing ad hoc routing protocols to support QoS demands. Among these protocols, QOLSR (QoS in OLSR protocol) [9] which is an enhancement of the OLSR routing protocol. OLSR is a proactive routing protocol. Its principle is to declare a subset of links with its neighbors that are its multipoint relay selectors (MPR) .Thus, only the MPRs retransmit the packets in the cases of flooding or broadcast procedures which reduces the number of retransmissions. QOLSR supports multiple-metric routing criteria by adding fields representing QoS conditions to the signaling messages of OLSR.

Other reactive routing protocols, which react to find a route only on demand, are also improved to support QoS requirements, such as AODV and DSR. In fact, AODV with QoS [7] consists in including a QoS Object extension specifying bandwidth and delay parameters on the signaling messages. Thus, a node treats the signaling packet only if it can satisfy the specified requirements.

Whereas, the MP-DSR [11], which is an improvement of DSR, consists on providing multiple available paths, after the Route Discovery phase is done, to improve the end-to-end reliability. Accordingly, the data packets are transmitted over the discovery paths satisfying the reliability requirements fixed by the sender.

B. Routing protocols designed for QoS

Other QoS solutions are independent from the existing ad hoc protocols and they can be divided into three types. The protocols based on IntServ, those based on DiffServ and the protocols which combine both IntServ and DiffServ.

The first type is based on IntServ architecture. It includes CEDAR (Core Extraction Distributed Ad Hoc Routing) [7] which consists in electing a set of nodes to form a core as a backbone for communication. Then, the bandwidth availability information of stable links is propagated to all core nodes allowing them to compute paths.

AQOR (Ad hoc QoS On-demand Routing) [7] also belongs to this type of protocols and it presents some mechanisms to allow QoS routing. It makes an admission control of new flows based on the available resources and it applies fast recovery on QoS violation situations. The detection of these QoS violations is the role of the destination which broadcasts a 'route update message' back to the source, if it identifies a QoS violation on the active route. In this case, the source re-routes traffic to the path of the first update message.

BRuIT (Bandwidth Reservation under Interferences influence) [8] is also a QoS routing protocol which adopts IntServ architecture. Its principle is based on bandwidth reservation. When a node wishes to reserve bandwidth for a flow, it broadcasts a route request message with the address of the receiver and the amount of desired bandwidth. Each node receiving this message checks if it has enough free bandwidth to handle the reservation. If the result is positive, the node forwards the request to its neighbors. Otherwise, the request is discarded. This is repeated until reaching the destination which sends a route reply message the way back to the source. Each intermediate node, when receiving the route reply, reserves bandwidth for this flow by decreasing its free bandwidth counter [16].

The second type focuses on DiffServ principle such as Courtesy Piggybacking [7] which suggests to piggyback low priority traffic into the high priority traffic whenever there is a free space. MQRD (Multipath QoS Routing protocol of supporting DiffServ) [6] is also a routing protocol supporting QoS and based on DiffServ. It provides multi-path routing to avoid traffic congestion and link breakage, and it uses DiffServ to divide traffics into different priority levels. Thus, it performs different mechanisms of scheduling and queuing to support QoS requirements.

The third type of protocols combines both IntServ and DiffServ principles like FQMM (Flexible QoS Model for MANETs) which provides QoS differently according to the traffic priority. IntServ is given to high-priority traffic, while DiffServ is used for other traffics. In fact, FQMM adopts a service differentiation and it provides to some classes of traffic, which have QoS constraints, the possibility to reserve sufficient resources [21].

Providing QoS in generally can't be often suitable for real time transfer. For this purpose, real time constraints can be treated with a particular strategy, as presented in the next section.

C. QoS Routing protocols designed for Real-time applications

Because of the challenge of transferring real time data through ad hoc networks, providing a routing solution for such flows is not an easy task. For this reason, most of the researches [10] have adopted a DiffServ principle based on the classification of flows. Among the proposed solutions, SWAN (Service Differentiation in Stateless Wireless Ad hoc Networks) [13] which is a stateless network model based on the feedback information got from the network. Getting this feedback, the distributed control algorithms used by SWAN deliver a service differentiation depending on the class of the traffic. In fact, SWAN assumes two types of traffics which are the best effort traffic and the real time traffic. The idea brought by SWAN is delaying the best effort traffic according to the demands of the real-time traffic in order to support the network conditions.

QPART (QoS Protocol for Ad hoc Real-time Traffic) [10] is also a proposition to provide probabilistic QoS guarantees to real-time applications for wireless ad hoc networks. It's based on scheduling the packets of flows according to their QoS requirements. Thus, we can find delay-sensitive flows which have the end-to-end delay as requirement, the bandwidth-sensitive flows which have the throughput as requirement and the best effort flows. Each real time flow in QPART has its own packet queue and contention window, and accesses the channel as if it is an independent node. The principle of QPART is to adapt the contention window sizes at the MAC layer. In fact, it assigns a smaller size of contention window to the traffic having a higher priority to increase the probability that it accesses to the media before the other traffics. These sizes are adjusted according to the network status until the network can no longer support the real-time traffic requirements. In this case, the traffics having a lower priority must be rejected.

All the above presented QoS routing protocols don't provide solutions for Hard real time transmissions. They only try to better respect QoS constraints and they didn't provide any guarantee. Some of them tried to improve the quality of transmission by adding QoS conditions like QOLSR and AODV with QoS, or by offering better availability such as MP-DSR. Some others tried to reserve resources for flows demanding QoS like BRuIT.

Concerning SWAN and QPART, both of them adopt DiffServ architecture which is not suitable for hard real time transmissions. First, DiffServ [3] doesn't offer an End-to-End QoS because it doesn't keep a per flow state information. Second, there is a possibility of congestions with DiffServ because of the absence of reservations. So, we decided to take advantage of IntServ's properties such as the possibility of quantitatively specifying the resources requirements.

In the context of this paper, we try to get a guarantee of real time constraints. So we propose a new Hard real time routing protocol.

Thus, we extend an existing Ad hoc routing protocol to incorporate real time constraints with adopting IntServ principle. We choose DSR routing protocol because it provides an excellent performance for routing in multi-hop wireless ad hoc networks which is proved by the authors in [17].

III. DSR: DYNAMIC ROUTING PROTOCOL

DSR (Dynamic Source Routing) [12] [2] [23] is a reactive Ad hoc routing protocol. It consists of two major phases which are Route Discovery and Route Maintenance. The route discovery major phase is subdivided in route request and route reply phases.

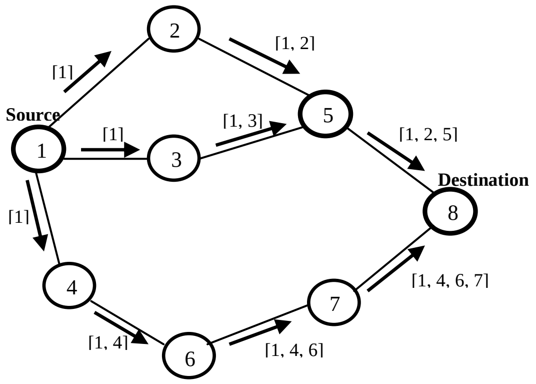


Figure 1: Broadcasting route request packet

When a node desires to send data to an unknown destination, it initiates a Route Request phase to locate the destination node. The principle of this phase is to broadcast a route request packet. Each intermediate node, when receiving a route request, adds it own address to the route record of the packet and forwards it to its neighbors. For example, in Figure 1, node 5 sends a request packet to node 8 with route record containing visited nodes 1, 2 and 5 to inform it about the discovered path.

This process is repeated until the request packet reaches the destination or one of the intermediate nodes finds a valid path to the destination node in its route cache. In these cases, the needed route is found and we must inform the source about it. For this purpose, we initiate a route reply phase in which a route reply packet is generated and sent back to the source node along the reverse recorded route. In the example of Figure 1, the destination receives the request packets having [1, 2, 5] and [1, 4, 6, 7] as recorded routes. It replies the first received route. In this case, it's [1, 2, 5] as shown in Figure 2. When receiving a route reply, the source node starts to send data packets through the discovered path.

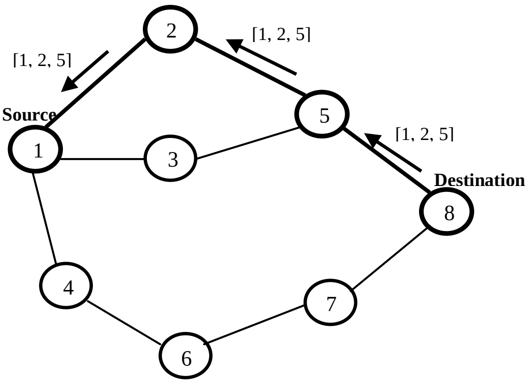


Figure 2: Sending a Route Reply back to the source

If one of the links constructing the route between the source and the destination breaks, a route error packet is generated and sent back to the sender to inform it and the intermediate nodes about the failure. This is provided by the route maintenance phase.

DSR can't be used for real time transfers because of many reasons. First, the delay requirement is not considered. So, we can't ensure that packets will reach their destinations before the deadlines. Second, the mobility issue is not well treated. In fact, when a link breaks, we should wait that intermediate node detects the link failure and send an error packet back to the source. This takes too much time which is not allowed regarding to the real time transfer.

In spite of these limits, DSR has some advantages which lead us to choose it as based protocol in our work. First, DSR is a reactive protocol which is more suitable for the real time transfer, since we can be sure that the constructed route is up-to-date. Second, DSR is simple and flexible [20] which facilitates the implementation of our real time extension. Also, a route request packet broadcasted in the route discovery phase can be used to incorporate real time constraints. That's why we chose to improve DSR protocol by adopting the IntServ architecture in order to make a new routing protocol supporting hard real time transfer.

IV. OUR PROPOSAL BASED ON EXPIRATION DELAY

The objective of our proposal is to provide a solution allowing hard real time flows to be transmitted correctly, in time, through Ad hoc networks. In order to ensure that the delay requirements of hard real-time traffic are met, we propose a solution based on the expiration delay to deadline (real-time constraint). Thus, each packet of the admitted real time flow is transmitted from the source to the destination before the expiration delay to deadline. For that reason, we adopt the IntServ principle based on the admission control and the reservation of resources.

To implement these mechanisms, we conceive two phases which should be performed before starting the transmission of real time data. One phase is designed for seeking a suitable path and the other phase is for the reservation of resources. So, the first phase allows finding, if possible, a path between the source and the destination which can satisfy delay requirements. It's called "Real Time Route Discovery". The second phase, which is called "Real Time route Reply", allows reserving resources of the discovered path, by the first phase, for the real-time flow.

A. Real-time route discovery phase

The route discovery phase allows the sender to find, if possible, a path to the destination which satisfies the delay requirements (deadline transmission constraints) of the real-time flow. In fact, when a node has to transmit real-time data to any other node, it broadcasts a route request with the expiration delay to deadline needed. Each node in the ad hoc network, receiving this route request (Activity -1- in "Fig. 3"), makes the admission control test in order to decide if it can accept a new real-time flow or not, as shown in "Fig. 3".

The goal of the admission control test is to check if the network can admit the new flow and satisfy its constraints without disrupting the real-time flows already admitted in the network. So, two conditions are needed to check if the delay constraints are satisfied. One condition checks the satisfaction for the new flow and the other condition checks the satisfaction of the previous real time flows transmitted over the network in the case of accepting this new flow.

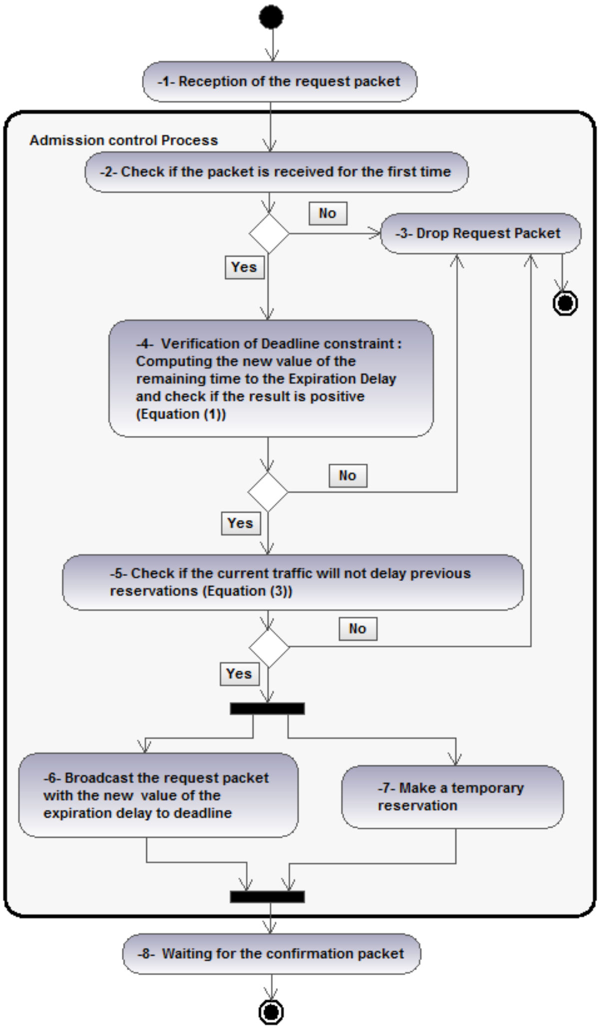


Figure 3: Activity diagram of intermediate node behavior in real time route request phase

Thus, the first condition (Activity -4- in "Fig. 3") is that the expiration delay to the deadline of the new flow can be respected which means that the transmission time between the source and the destination is still lower than the expiration delay to deadline. For that reason, we should take into account the time spent by the packet in each node and the communication time between nodes. That's why each node receives the remaining time of the expiration delay from the previous node. Then, from this received value, the node subtracts the time taken to transfer the request packet from the previous node and the packet processing time. The obtained value should be positive to maintain the respect of deadline on transmission (expiration delay). This can be represented by the following formula:

$$E_k^{i-1} - T_{TL} - T_{TS} > 0 \qquad (1)$$

Where E_k^{i-1} is the remaining time of the expiration delay to deadline, for the traffic k, received from the node (i-1). T_{TL} is the local processing time of any message. T_{TS} is the transmission time between two neighboring nodes in the worst case.

The local processing time is considered as the sum of the processing time machine and the time spent by the packet in the queue:

$$T_{TL} = T_a + T_{TM} \qquad (2)$$

Where T_{TM} is the processing time machine and T_a is the time spent by the packet in the queue.

To better understand this principle, we illustrate it with an example. We assume that a node i receives a request packet from the node (i-1) with the remaining time to the expiration delay having the value 10*ms*: so E_k^{i-1}= 10*ms*. We suppose also that the local processing time of any message in node $\boldsymbol{i}$ is 3*ms*: so T_{TL}= 3*ms*, and the transmission time between two neighboring nodes in the worst case within the ad hoc network is *2ms*: so T_{TS} =2*ms*. To check that the expiration delay is not reached, we use "(1)" as follows:

$$E_k^{i-1} - T_{TL} - T_{TS} = 10 - 3 - 2 = 5 > 0$$

We obtain a positive value (*5ms*), so this condition is validated since a packet still has *5ms* to reach the destination before the expiration delay to deadline.

While the first condition "(1)" checks that packet can reach the destination before the expiration delay to deadline, the second condition imposes that a new flow shouldn't delay the other real-time flows already admitted and stop them to respect their expiration delays to deadline (Activity -5- in "Fig. 3"). To verify the respect of deadline, we must save all reserved traffics with their remaining times. An additional traffic shouldn't delay previous ones to respect their expiration delay to deadlines. That's why we recalculate the first condition "(1)" to all the accepted flows taking into account the changes if we admit a new flow. This can be resumed by the following formula:

$$\forall j,\ 1 \le j \le res;\ \ E_j^{i-1} - T_{TL} - T_{TS} > 0 \qquad (3)$$

Where *res* is the number of real-time flows already admitted in the node.

Based on these formulas "(1)" and "(3)", the node makes a decision if it can admit a new real-time flow or not. If a flow is admitted, the node broadcasts the route request packet with the new remaining time to the deadline (Activity -6- in "Fig. 3") and makes a temporary reservation (Activity -7- in "Fig. 3"). Otherwise, the node discards the route request packet (in route discovery phase, Activity -3- in "Fig. 3") and doesn't send it anywhere because the node can't satisfy the delay constraints; therefore, it shouldn't belong to the path between the source and the destination of the actual real time traffic.

B. Real-time route reply phase

The goal of the route reply phase is to effectively reserve the path between the source and the destination. In fact, once we find a route satisfying the delay constraints between the transmitter and the receiver, we should pass it on to the source and inform the nodes belonging to this route about the new real-time traffic and the final remaining time to deadline. The nodes activities in this phase are shown in “Fig. 4”.

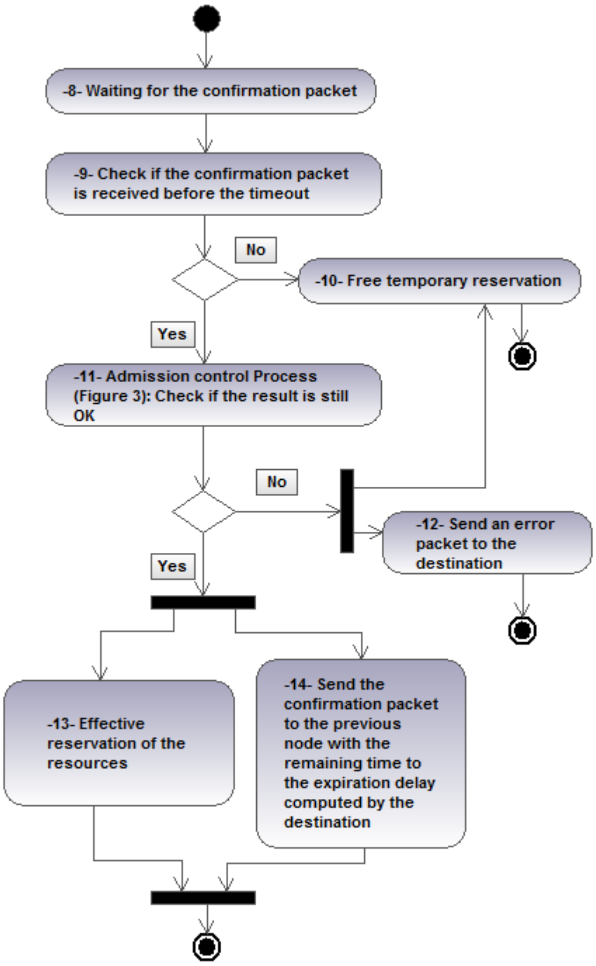


Figure 4: Activity diagram of intermediate node behavior in real time route reply phase

Thus, when the destination receives the route request packet at the end of route discovery phase, it performs the admission control (as explained in the previous subsection, Activity -11- in “Fig. 4”). If the admission control result is negative, the destination rejects the packet. Otherwise, it sends a confirmation packet to the source, in a reverse order of the nodes making up the path, with the final remaining time to deadline (Activity -14- in “Fig. 4”).

Each intermediate node, when receiving the confirmation packet, performs the admission control again. If the conditions are not satisfied, the confirmation packet is rejected, an error message is sent to the destination to inform it that it should select another path (Activity -12- in “Fig. 4”) and the temporary reservation is freed (Activity -10- in “Fig. 4”). Otherwise, the intermediate node reserves the resources (Activity -13- in “Fig. 4”), saves the remaining time value to deadline sent by the destination and sends the confirmation packet to the next node until reaching the source.

When the source receives the confirmation packet, it starts transmitting the data flow through the reserved path until it hasn’t got other packets to send or the path is no longer available (because of nodes mobility). In this case, we reinitiate a route discovery phase to make a new path respecting the real time constraints. At the same time, nodes of the old path should free their reservations. This can be done by defining a time-out for each reserved flow. Thus, if this traffic is not arriving in the specified time-out, the node will be freed from reservation.

The acquiring of an appropriate path for such traffics is not enough, because the availability of paths and the respect of real time constraints depend also on nodes mobility. That’s why we should study the mobility issue in order to provide solutions even in a mobile environment.

V. MOBILITY ISSUE

Since Ad hoc networks have no fixed infrastructure, each node can move in an arbitrary manner. This engenders a challenging issue which is how to be adapted to frequently changing network topologies in a way that a real-time session can’t be interrupted and the transmission delay can usually meet the deadline. For that, we study different mobility models and we try to predict the mobility of the network nodes in a way that we can analyze the respect of real time constraints (deadline).

A. Mobility models for Ad hoc networks

To study the mobility prediction issue, several models for Ad hoc networks have been proposed in literature [3] [15], [24], [19]. These models can be classified into two categories: single user mobility models and group mobility models.

Among the user mobility models, we find the random walk model [25] which defines the user movement between two positions with memory-less randomly selected speed and direction. We find also the Markovian model.

Other models are defined for a group mobility like the pursue model and the column model [19] which study the relationship between mobile nodes, in-disaster recovery and military situations. The RPGM (Reference Point Group Mobility) [24] is also a group mobility model which uses the mobility centre of the group to define the behavior of the entire group. Finally the mobility vector model offers a flexible mobility framework for hybrid motion patterns [19].

B. Mobility prediction

The main idea for resolving the mobility issue is to previously have enough information about the nodes movement. Therefore, we can determine the instants of topology changes, so that we can plan a new route discovery at these instants. For the period of time between those instants, we can consider that the topology stays unchanged. So, a new route discovery is initiated after this period and the resources, already reserved, are freed.

In the literature, some works [25], [4] , [14] tried to estimate the link availability. In fact, in [14], a link breakage prediction algorithm was added to the Dynamic Source Routing Protocol (DSR). It consists on using the signal power strength from the received packets to predict the link breakage time and send a warning to the source node if the link is soon-to-be-broken. In this case, the source performs a pro-active route rebuilt to avoid disconnection. This algorithm is based on the following formulate [14]:

$$P = k\frac{P_t}{{d_0}^4} \qquad (4)$$

Where P is the signal power at the receiver, Pt is the signal power at the transmitter, k is a constant and d_0 is the distance between the two nodes. When the signal power P becomes lower, we can suppose that this node will leave and the direct link to this node will be broken.

In the paper [5], authors present an iterative algorithm that predicts continuously the link availability between two mobile nodes. For that, they estimate the probability ($L(d_0,t)$) that the link between two mobile nodes will be continuously available in a defined period of time (between t_0 and t_0+t which called epoch time) having the initial distance (d_0) between these nodes. The calculations were done in the case that the initial velocity of the nodes in unknown and in the case that the initial relative velocity is known. However, in this article, it's mentioned that the work is applicable to many areas in ad hoc networks and it can be used to improve the DSR routing protocol.

While in [4], the authors resort to predict the stability by characterizing the mobility of mobile nodes. So, they propose a new scheme to estimate the mobility parameters such as relative speed, orientation and epoch time, in real-time. These parameters are estimated only from the time-varying inter-node distance information without need to localization systems or features of wireless channel. Its principle is that every node knows both the distances between its one hop neighbors and the distance separating it to them. Those distances are measured periodically by means of either received signal strength (RSSI), time-of-arrival (ToA), or time-difference-of-arrival (TDoA) measurements [4].

Other works studied the connectivity of mobile ad hoc networks such as in [1] where the authors have modeled the connectivity using Markov jump theory. They established so many definitions and theorems to formulate the connectivity model properties.

These approaches, in order to predict mobility, can be summarized on two categories. The probabilistic approaches such as markovian ones which are not adequate for hard real time traffic, and the deterministic approaches consisting in predicting the link breakage or the mobility of nodes such as the method based on the signal power strength which can be used in our case.

C. Our approach of mobility prediction

Since the nodes are mobile, the selected path to transmit real-time data can be broken at any moment. To avoid the violation of real-time constraints, we should predict the instants in which the topology can change, and the path can be no longer valid, in the worst case. Therefore, we have to perform a new route discovery. So, we were inspired by the approaches presented in the precedent section to conceive our check model composed of two modules.

The first module allows determining the possible paths. While the second module checks the path availability and the deadline delay satisfaction. This check model is presented by the following diagram ("Fig. 5"):

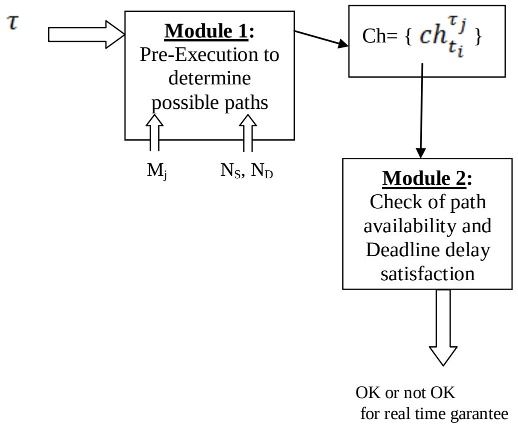


Figure 5: pre-execution check model to predict real time guarantee

Where $\tau = \{(d_j, P_j)\}$ is a set of CBR traffics defined by the Expiration delay d_j (deadline) and the inter-arrival period P_j, NS is the Source node, ND is the Destination node and M_j is the Movement function of the mobile nodes in ad hoc network.

Ch is the set of paths determined from the pre-execution module (Module 1) which consists in performing different simulations to have an idea about the nodes behavior.

The second module is the checking of path availability and deadline delay satisfaction. For that reason, we assume that each node in the ad hoc network has a predictable movement like the movements of robots in a production manufacturing with a particular mobility pattern. We define this movement by the function $M_i(t)$ which indicates the position of the node i at the instant t. So, we have the equality $M_i(t)=(x(t), y(t))$ when (x,y) is the node position at the instant t. We assume also that the movement of the nodes is cyclical. So, if the cycle is T_i, we have the following property: $M_i(t+T_i)=M_i(t)$.

As shown in the diagram ("Fig. 5"), we must start by doing pre-execution to establish paths. That's why we adopt the property 1 which is based on the periodical characteristic of the nodes movement pattern.

Property 1: $M_i(t)/t \in [0,\infty] \Leftrightarrow M_i(t)/t \in [0,T]$

Where $T=PPCM\ (T_i)$, $i \in [1, N]$, N: number of nodes.

This property allows checking the availability of the selected path only in the period T since the movement pattern of the nodes will be similar in the other periods ("Fig. 6").

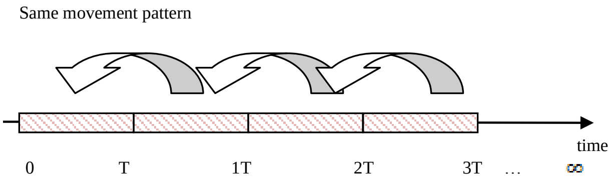


Figure 6: Periodical characteristic of the nodes movement pattern

After the pre-execution to determine possible paths, the check model will be programmed to validate different paths with regard to deadline (second module). Thus, we should check two conditions, in the *T* period, which are the availability of the path and the satisfaction of delay constraints.

- **First Condition: Availability of the path**

This condition allows to check how long is the path available and if the data can be transmitted from the source to the destination over the same route, already selected.

For that reason, we assume that the path, noted *chX(t)* for the instant *t* and the traffic *X*, is available when the neighboring nodes don't leave the coverage of each other's (no path breakage) . We can use the signal power strength (Eq. 4) to predict the link breakage time between two neighboring nodes. In fact, when the signal is weak, a warning message is sent to the source node informing it that the link is soon-to-be-broken. In this case, the source performs a route rebuilding.

From the previous strategy, we can get the set of paths used by the source node relying on the instants of topology changes. All these paths must be verified regarding real time constraints. This is the aim of the second condition.

- **Second condition: satisfaction of delay constraints**

To verify the satisfaction of delay constraints, we should check that the response time *R* is lower than the expiration delay *d* *(R<d)*. This condition can be represented by the following expression:

$$\forall t \in [0,T], R(ch^j(t)) < d_j \qquad (5)$$

Where j is the traffic, *R* is the Response time, d_j is the Expiration delay (deadline on transmission) of the traffic *j* and *ch* is the path taken by traffic *j* at the instance *t*.

Verifying the delay constraint for the specific traffic *j* means that, for each instant of topology's changes, the response time for the selected path should be lower than the defined expiration delay to deadline. This is represented by the following property:

<u>Property 2</u>:

$$\forall t \in [0,T], R(ch^j(t)) < d_j \Leftrightarrow \qquad (6)$$
$$\forall t \in \{t_1, t_2, \ldots, t_n\}, R(ch^j(t)) < d$$

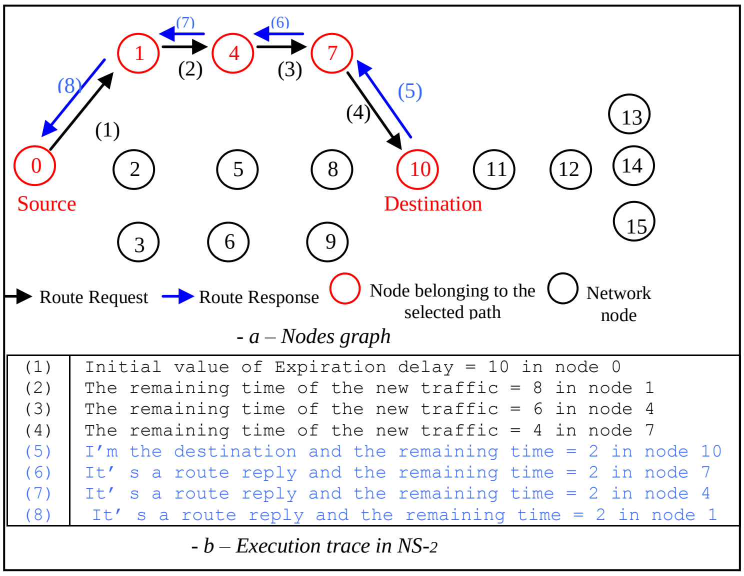
Where $\{t_1, t_2, \ldots, t_n\}$ is the set of change path instants. This property (Eq. 6) minimizes the checking instants of real time constraints (expiration delay).

VI. EVALUATION

To evaluate the performance of our proposal, we implemented RT-DSR (Real-Time DSR) using the ns-2 simulator. RT-DSR is the improvement of DSR protocol by introducing the principle of the proposed admission control based on the Expiration delay (or deadline). In fact, we modified some files, related to the functioning of DSR protocol, in the source code of NS simulator. We added the principle codes, representing the admission control and the reservation, to the file *dsragent.cc* which is DSR agent class handling routings. However, we introduced also some modifications to other files such as *request_table.cc* and *hdr_sr.cc* to get RT-DSR extensions for routing packet headers.

A. Simulation environment

As mentioned before, we have used the NS-2 simulator in our simulations. The basic simulation scenario consists of 16 nodes in an area of 1300x1300m2. Each mobile node has a transmission range of 250 meters. Hence, we can represent multi-hop communications without shortcuts.



- *a – Nodes graph*

```
(1)  Initial value of Expiration delay = 10 in node 0
(2)  The remaining time of the new traffic = 8 in node 1
(3)  The remaining time of the new traffic = 6 in node 4
(4)  The remaining time of the new traffic = 4 in node 7
(5)  I'm the destination and the remaining time = 2 in node 10
(6)  It' s a route reply and the remaining time = 2 in node 7
(7)  It' s a route reply and the remaining time = 2 in node 4
(8)   It' s a route reply and the remaining time = 2 in node 1
```

- *b – Execution trace in NS-2*

Figure 7: Ad hoc network scenario executed in NS2

The "Fig. 7" stands for the basic simulation network which can make multi-paths from source node *0* to destination node *10*. Among them, we chose a matrix sub-network of dimension 3X3. The simulation variances will be the workload, the number of traffics, the mobility and the expiration delay or deadline on transmission.

Because we treat hard real time traffics, we focus on the response time as result of simulations with regard to DSR and our routing protocol RT-DSR. Our simulation is based on the comparison with DSR because we want to perform extensions to this protocol for real time.

B. Admission control tests

To validate the admission control functionality, we assume that the node 0 wants to send real-time traffic to node 10 with an expiration delay equal to "10" (as shown in "Fig. 7"). Then, we observed the behavior of the nodes when receiving the request route of this flow. The "Fig.7" shows that the initial value of the Expiration delay (remain time to deadline) decrements, within the value 2, in each node receiving the route request, until reaching the destination. In this example, the path selected is 0=>1=>4=>7=>10. That's why the destination node, 10, sent a confirmation packet to the source 0 through the reverse path informing the intermediate nodes and the source about the final value of the remaining time which is "2". Indeed, if a new traffic wants to allocate a path containing a previous reserved node, the admission control will be verified with regard to the remaining time “2”. Thus, we can preserve previous reservation without distorting the real time constraint (guaranteed remaining time to deadline).

C. Comparison between RT-DSR and DSR in a fixed architecture

To compare the transmission delays of the same traffic between our proposal RT-DSR and DSR, we have performed simulations of various cases observing the delays. The “Fig. 8” shows the delay to transmit the packets of the first flow under DSR and RT-DSR when increasing the number of traffics in the network. It is clear that RT-DSR offers a better delay, compared to DSR, which is notably lower and more stable. This stability is ensured by the path reservation that has been done for the first traffic. We can also see, as shown in “Fig. 8”, that for a small number of traffic, we have almost the same result. This is due to the use of the cache by DSR. However, when the number of traffics increases, we see the importance of RT-DSR which is presented by the decrease and stability of the transmission delay.

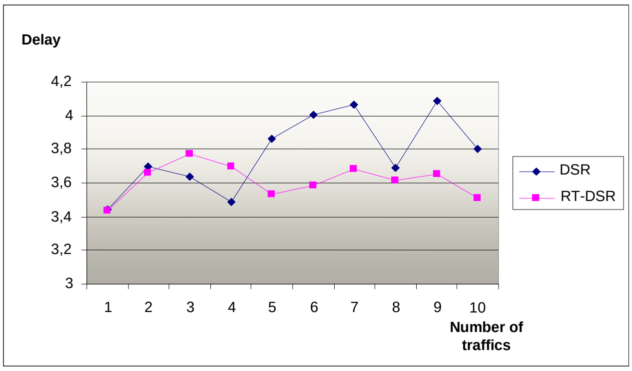


Figure 8: Delays depending on the number of traffics

D. Comparison between RT-DSR and DSR in the case of moving nodes

“Fig. 9” shows the delay values, with DSR and RT-DSR, when the nodes between the source and the destination move. In fact, we consider that some nodes have moved in a rectangular way allowing them to come out from the ranges of each other. We observe that the RT-DSR delay is still lower than DSR delay. Also, it's notably stable. So, we can conclude that our solution is more interesting especially in a dynamic topology.

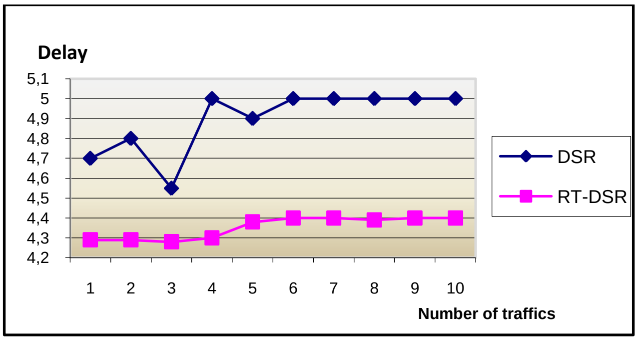


Figure 9: Delays depending on the number of traffic in the case of moving nodes

E. Comparison of different Expiration Delay values

“Fig. 10” shows the delay values of the first flow according to the Expiration delay values. First, we note that RT-DSR delay values are more stable than DSR delay. Then, we study the effect of expiration delay on RT-DSR transmission delay. For a small number of traffics (between 1 and 4), when the expiration delay value is reduced (noted E), the transmission delay decreases consequently. However, when the number of traffics is great, the delay values are close. In fact, from 10 traffics, the different delay values are almost the same. With a small number of traffics, there are several paths with RT-DSR. Whereas, if a number of traffics is important, there is no other alternative to get a new path. This leads to stabilize the time delay regardless of the expiration delay value. We note also that between 4 and 9 traffics, the transmission delay is better for lower expiration delay (E=11). Indeed, our routing protocol (RT-DSR) finds the path verifying that transmission delay is lower than expiration delay (respect of deadline). The more the expiration delay is lower, the more the transmission delay will be.

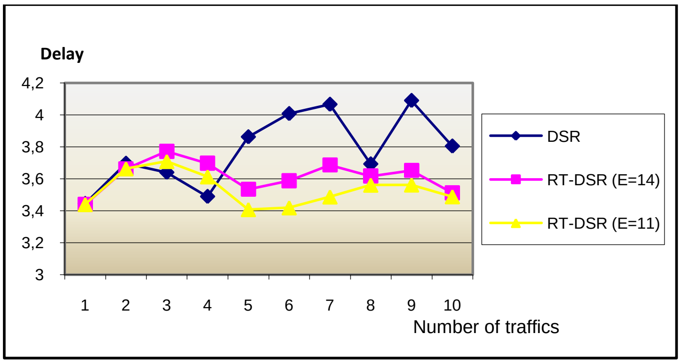


Figure 10: Comparison of delays depending on Expiration delay values

VII. CONCLUSION

In this paper, we present a new real time proposal RT-DSR. It's a routing protocol which allows the packets of the real time flows to be routed from the sender to the receiver before the expiration delay to deadline. This is done by reserving the selected path between the source and the destination nodes. This reservation is maintained by the admission control mechanism. We predict also the validity periods of the selected path and the satisfaction of delay constraints using a check model. This later can make a decision about the real time constraint respect. With assumption a predictable and periodical movement pattern, we raise the constraint guarantee problem of hard real time applications.

Simulations are made to prove the feasibility and the performances of our proposal routing protocol. The RT-DSR offers better delay which is particularly lower and more stable than DSR even in mobile context. In future works, we will investigate the case of unreliable networks in which the nodes can be broken down. In general, we'll try to conceive a routing solution with more realistic hypothesis.

REFERENCES

[1] Demin Li1, Jie Zhou2 and Jiacun Wang3,"A Markov Jump Theory Based Connectivity Model of Mobile Ad Hoc Networks", 2007 International Symposium on Nonlinear Dynamics (2007 ISND), Journal of Physics: Conference Series 96 (2008) 012012, 2008

[2] B. Hu and H.Gharavi, "DSR-Based Directional Routing Protocol for Ad Hoc Networks", National Institute of Standards and Technology, IEEE Communications Society, IEEE GLOBECOM, 2007.

[3] G. Bos, "QoS support using DiffServ", 6th TSConIT, 2007

[4] Z. Li, L. Sun and Emmanuel C. Ifeachor, "Range-based Mobility Estimations in MANETs with Application to Link vailability Prediction", School of Computing, Communications and Electronics, University of Plymouth, 2007

[4] Z. Li, L. Sun, and E. C. Ifeachor, "Range-based relative velocity estimations for networked mobile devices," Submitted to IEEE Transactions on Vehicular Technology, August 2006.

[5] Min Qin, Roger Zimmermann and Leslie S. Liu, "Supporting Multimedia Streaming Between Mobile Peers with Link Availability Prediction", Department of Computer Science, University of Southern California, Los Angeles, CA 90089, MM'05, November 6–12, 2005, Singapore.

[6] Li Xuefei and L. Cuthbert, "Multipath QoS routing of supporting DiffServ in mobile ad hoc networks", Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005.

[7] A. Proutiere, "QoS in multi-service wireless networks A state of the art", Workpackage: http://www.eurongi.org, Project Acronym: Euro-NGI Project Full Title: Design and Engineering of the Next Generation Internet, towards convergent multi-service networks , August 2004

[8] C. Chaudet, " Autour de la réservation de bande passante dans les réseaux ad hoc", INRIA – Institut National des Sciences Appliquées de Lyon France, septembre 2004

[9] H. Badis, K. Al Agha, "Optimal Path Selection Analysis in Ad Hoc Networks", in: Technical Report; LRI, université de Paris-Sud XI, August 2004.

[10] Y. Yang, "Distributed QoS Guarantees for Realtime Traffic in Ad Hoc Networks", Department of Computer Science, University of Illinois at Urbana-Champaign, October 2004.

[11] A. Laouiti and C. Adjih, "Mesures des performances du protocole OLSR, Hipercom", in international conference in Sciences of Electronic, Technology of Information and Telecommunications (SETIT), 2003.

[12] X. Hou and D. Tipper, "Impact of Failures on Routing in Mobile Ad Hoc Networks Using DSR", Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, Jan., Orlando, 2003.

[13] Gahng-Seop Ahn, Andrew T. Campbell, Andras Veres and Li-Hsiang Sun , "SWAN: Service Differentiation in Stateless", Wireless Ad Hoc Networks, IEEE, 2002

[14] L. Qin and T. Kunz, "Increasing Packet Delivery Ratio in DSR by Link Prediction", Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada K1S 5B6, IEEE2002

[15] T. Camp, J. Boleng, V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", Wireless Communications and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, Volume 2, No.5, 2002.

[16] C. Chaudet and I. Guérin-Lassous, " BRuIT : Bandwidth Reservation under InTerferences influence", July 2001

[17] D. B. Johnson, D. A. Maltz and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", 2001.

[18] R. Leung, J. Liu, E. Poon, A. Charles Chan and B. Li, "MP-DSR: A QoS-aware Multi-path Dynamic Source Routing Protocol for Wireless Ad-Hoc Networks", Department of Electrical and Computer Engineering, University of Toronto, 2001

[19] X. Hong, T. Kwon, M. Gerla, D. Gu and G. Pei, "A Mobility Framework for Ad Hoc Wireless Networks", Proceedings of ACM 2nd International Conference on Mobile Data Management, MDM 2001,Hong Kong, January 2001.

[20] Y. –C Hu and D. B. Johnson, "Implicit Source Routes for On-Demand Ad Hoc Network Routing", *ACM MobiHoc*, 2001.

[21] Hannan XIAO1, Winston K.G. SEAH2y, Anthony LO2 and Kee Chaing CHUA2, "Flexible Quality of Service Model for Mobile Ad-Hoc Networks", Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st.

[22] T. LEMLOUMA, Dr. N. BADACHE, "Le Routage dans les Réseaux Mobiles Ad Hoc", Université des Sciences et de la Technologie Houari Boumèdiene, Institut d'Informatique, Septembre 2000.

[23] S. Basagni, I. Chlamtac, and V. R. Syrotiuk, "Dynamic source routing for ad hoc" networks using the global posit ioning system. In Proceedings of the IEEE Wireless Communications and Networking Conference 1999 (WCNC'99), New Orleans, Louisiana, September 1999.

[24] X. Hong, M. Gerla, G. Pei, C. Chiang, "A Group Mobility Model for Ad Hoc Wireless Networks", Proceedings of ACM/IEEE International Workshop on Modeling and Simulation or Wireless and Mobile Systems, MSWiM '99, Seattle, August 1999.

[25] M. M. Zonoozi and P. Dassanayake, "User mobility modelling and characterization of mobility patterns", IEEE Journal on Selected Areas in Communications, Volume 15, No.7, September 1997.

[26] D. Ballard and D. Nielson, "A real-time knowledge processing executive for Army rotorcraft applications", Digital Avionics Systems Conference, 1992.

Sofiane Ouni received his Engineer, Master degree and PhD in Computer Science from Ecole Nationale des Sciences de l'Informatique (ENSI) of Tunisia in 2004. His past researches and publications focus on access methods of industrial local area networks and Hard Real Time.

Since 1998, he worked as Assistant Professor at Institut National des Sciences Appliquées et de Technologie (INSAT). He is currently Associate Professor at the Department of Computer Science and Mathematics, at INSAT.

The current research interest of Dr. Sofiane Ouni includes Wireless, Ad Hoc and sensor Networks with Hard Real time guarantee. He is reviewer and organizer of International conferences (CFIP’06, WMSCI, CCCT, ICESCA'08) on computer networks and embedded systems.

Jihen Bokri is an Assistant Professor of Computer Science and a PhD student working on real time communication over Ad hoc Networks. She received her Master in Computer Science from Institut National des Sciences Appliquées et de Technologie (INSAT) and Ecole Polytechnique de Tunis (EPT) in 2007. Before being an Assistant professor, she worked as HP support engineer at Hawelett-Packard and as a Software engineer at SAGEM Softwares & Technologies. Now, Her current research focuses on Ad Hoc networks, mobility and Real Time QoS .

Farouk Kamoun received his PhD degree from the University of California at Los Angeles Computer Science Department in 1976. He is Professor in computer science since 1982. He is currently Professor of Computer Sc. at Ecole Nationale des Sciences de l'Informatique (ENSI) of Tunisia. He founded in 1999, "CRISTAL" laboratry at ENSI. His Research work is focused on the QoS and multimedia communications for the Internet, Security, Mobility, Wi-Fi and Ad Hoc networks as well as e-commerce.

From 1982 till 1993, he chaired the CNI, Centre National de l'Informatique, a Government Office in charge of IT policies and strategic development. From 1993 till 1999, He served as Dean of ENSI. He is also Advisor to the Minister of Higher Education of Tunisia, Scientific Research and Technology, in charge of Information Technology.

Pr. Farouk Kamoun is author and co-author of over 100 scientific publications, technical and strategic reports. He is member of the International Council for Computer Communications (ICCC), the IFIP Technical Committee 6 and The Internet Society and the editorial board of 2 scientific journals. He also organized several workshops and conferences on Internet and recently on Ad-Hoc networks.