Dynamic Source Routing with Delay and Bandwidth Guarantees for Mobile Ad Hoc Networks

Bentaleb Abdelhak(1), Hicham Lakhlef(2), *Jean Frédéric Myoupo(3)
1 Department of Mathematics and Computer Science, University of Bachir el Ibrahimi Bordj Bou Arreridj, Algeria, Bentaleb_abdelhak@yahoo.com
2 Université de Franche-Comté, UFR Sciences, Cours le Prince Ringuet 25200 Monbéliard France, hicham.lakhlef@univ-fcomte.fr
3 Université de Picardie-Jules Verne, UFR Sciences, 33 rue Saint Leu, 80039 Amiens France, jean-frederic.myoupo@u-picardie.fr

Abstract

Mobile ad-hoc networks (MANETs) are a specific kind of wireless networks that can be quickly deployed without pre-existing infrastructures. They are used in different contexts such as collaborative, medical, military or embedded applications. However, MANETs raise new challenges when they are used to support multimedia and real time applications (e.g., videoconference, VoIP, Video on Demand, etc.) that require constraints on Quality of Service like the delay or the bandwidth. Indeed, these networks undergo drawbacks due to both the characteristics of the transmission medium (transmission medium sharing, low bandwidth, etc.) and the routing protocols (information diffusion, path calculation, etc.). In this paper, we have added and enhanced the aspect QoS in dynamic source routing (DSR) to get the new protocol called DB-DSR (Delay & Bandwidth Dynamic Source Routing). In this proposal we have considered the constraints of bandwidth, delay and coupling of these two constraints together in research and establishment of paths. The simulation results obtained after implementation of our protocol in OPNET showed the efficiency and level of performance of our protocol compared to the DSR protocol.

Keywords: Mobile Ad Hoc Network, Routing, Routing Qos.

1. Introduction

A Mobile Ad hoc NETwork (MANET) consists of a group of mobile nodes that spontaneously form temporary networks without the aid of a fixed infrastructure or centralized management. Such networks are characterized by: Dynamic topologies, existence of bandwidth constrained and variable capacity links, energy constrained operations and are highly prone to security threats. Due to all these features routing is a major issue in ad hoc networks [1].

Several routing protocols have been proposed in mobile ad hoc networks. These protocols can be classified according to the manner of creation and maintenance of routes when routing according to several criteria. The working group MANET (Mobile Ad hoc NETwork) of the IETF (Internet Engineering Task Force) is concerned with the standardization of ad hoc protocols running IP. Among these protocols we can cite AODV [2], Optimized Link State Routing (OLSR) [3], and DSR [4]. All these protocols work in best effort mode. For some applications such as multimedia or real-time the best effort service is not at all sufficient. Such applications require guarantees under certain requirements of quality of service (minimum bandwidth, maximum delay not to exceed, etc ...). Indeed, it seems important to adapt MANETs to support a certain level of QoS in order to deploy demanding applications.

1.1. Related work

Routing is a fundamental problem in MANETs. Currently, as we say above, in the IETF MANET group, four routing protocols have been finally standardized, including AODV [2], Optimized Link State Routing (OLSR) [3], DSR [4] and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [5]. Standard ad-hoc routing protocols can be divided into two categories: reactive (on-demand) and proactive. On demand routing protocols will flood route discovery messages upon arrival of a connection request. The on-demand routing protocols include AODV, DSR. Proactive routing...
protocols require the nodes to respond to changes in network topology by broadcasting updates throughout the network. OLSR and TBRPF fall into this category.

Supporting end-to-end QoS in MANETs is very challenging. The AODV has been extended to support QoS in MANETs [6]. A resource reservation based routing and signaling protocol, Ad-hoc QoS on-demand routing (AQOR) has been introduced in [10, 17]. In [11], Yang and Kravets presented a new algorithm to estimate the available bandwidth in the node for MANETs. In [12], R.Kumar, M.Misra and K.Sarje presented new algorithm to estimate the delay in the node for MANETs. All the routing protocols [2, 3, 4, 18, 19] work in best effort mode. For some applications such as multimedia or real-time the best effort service is not at all sufficient. Such applications require guarantees under certain requirements of quality of service (minimum bandwidth, maximum delay not to exceed. etc ...). Indeed, it seems important to adapt MANETs to support a certain level of QoS in order to deploy demanding applications.

1.2. Our contribution: DB-DSR

This work is part of implementing QoS in ad hoc routing protocols. Our study offers primarily a detailed study of protocol DSR, and then makes an extension called DB-DSR in order to provide quality of service in these protocols, while respecting the constraints Bandwidth, Delay of QoS and coupling of these two constraints together in research and establishment of paths. The simulation results obtained under the OPNET simulator shows that our protocol (DB-DSR) improves a good performance in terms of end to end delay, throughput, jitter, and data dropped.

The rest of the paper is organized as follows: Section 2 revisits related work. In section 3, we propose DB-DSR (DSR with Delay and Bandwidth constraints), a new routing protocol with two constraints of QoS. It is based on Kravets and Yang [11] work to estimate the available bandwidth at node and on Kumar, Misra and Sarje [12] work to estimate the delay and of the draft (RFC) of Johnson, Hu and Maltz, authors of the DSR protocol for working mechanism. In section 4, we evaluate the performance of our protocol. The conclusion ends the paper.

3. Delay and Bandwidth Dynamic Source Routing

In this section, we present our protocol, namely, DB-DSR (Delay and Bandwidth Dynamic Source Routing). It is an extension for the DSR protocol. We will discuss the specifications for a solution that integrates QoS in the DSR protocol based on two metrics, bandwidth and delay. The choice of these two metrics is justified by the need to deployment of applications sensitive to these metrics in MANETs, such as real-time and multimedia applications (videoconferencing, Voice over IP .. etc).

To reach this goal, we first start with the description of bandwidth and delay estimation methods. The estimation model used to estimate the available bandwidth in this work is based on the work of R. Kravets, Y. Yang [11], and the estimation model used to estimate the delay in this work is based on the work of R.Kumar, M.Misra and K.Sarje [12]. Next we make the description of the algorithm of periodic packets exchange used in the estimation models. Subsequently, we explain the necessary changes added to the structure of messages exchanged and the new mechanism of research of routes and source route selection used in route discovery ensuring the two metrics (delay end-to-end and bandwidth).

3.1 Algorithm of periodic packets exchange

In the algorithm of periodic packets exchange (Algorithm1) [13], each node in the network shares your flow state information\(^1\) with their neighbors in each period \(T\) seconds (Figure 1).

\(^{1}:\) Flow State Information: contains the values of the bandwidth consumed by the flows reserved at the node and other local information, these values used by neighbors in the estimation models.
Algorithm 1. Periodic_flow_state_exchange(packet* flow_state_packet)
N: number of nodes existing in interference range of the node i
For all ni ∈ N
Broadcast the flow stats information in each period T second
For All nj ∈ N and nj ≠ ni
Store the data extracted from the flow state packet received
Rebroadcast the received flow state packet until neighbors of 3 hops.
End for
End for

The problem of the algorithm 1 is that the number of flow states packets (Figure 2) transmitted in the interference range of the node i depends on the number of nodes existing in their interference range. Here it produces an increase in overheads on traffic\(^2\). We propose a solution presented in algorithm 2 to overcome this problem.

Algorithm 2. Periodic_flow_state_exchange_sol(packet* flow_state_packet)
N: number of nodes existing in interference range of the node i
M: Maximum number of rebroadcast of the same flow state packet, fixed to 3 hops.
For all ni ∈ N
Broadcast the flow stats information in each period T second
For All nj ∈ N and nj ≠ ni
Store the data extracted from the flow state packet received
Compute: Value = M/N
If (Value > 1) Then
Rebroadcast the received flow state information until neighbors of 3 hops.
End if
End for
End for

Figure 1. Broadcast the flow state packet information

Figure 2. Overheads in the two Algorithms

\(^2\): The increase in overhead on the traffic influenced in the estimation models used.
3.2 Estimation of the available bandwidth in the node

The method used to estimate the available bandwidth in the node [11] is used to perform admission control for all new routes between the source node and destination node. The objective of admission control is to determine whether the available resources can meet the requirements of a new flow while maintaining bandwidth levels for existing flows. Each node views a different channel state. The available bandwidth in the network is not a local concept. To tackle this condition, two terms are introduced: local available bandwidth ($BW_{local}$), neighborhood available bandwidth ($BW_{neigh}$).

Local available bandwidth is the amount of unconsumed bandwidth as observed by a given node. Neighborhood available bandwidth is the maximum amount of bandwidth that a node can use for transmission without affecting the reserved bandwidth of any existing flows in its carrier sensing range. Here the admission control is performed in each node by estimation of the available bandwidth at each node [11] for the new RREQ received. And the value of the available bandwidth estimated by the node is compared to the value of bandwidth requested ($BW_{req}$) that exists in the RREQ packet. If the RREQ does not satisfy the QoS constraints, the RREQ is simply dropped at the intermediate node with no further action, otherwise the RREQ is handled according to the DSR protocol rules. Upon reception of the new RREQ by a node, the algorithm used for estimating the available bandwidth starts. It computes two values $BW_{local}$ and $BW_{neigh}$. The minimum value between the two calculated values is used as the upper limit decision to accept the RREQ by the admission control of node.

3.2.1 Estimation of the local available bandwidth

It is the unconsumed bandwidth at a given node. Each node in the MANET can determine its $BW_{local}$ by passively listening to network activities. In our approach, we use the fraction of channel idle time based on the past history as an indication of local available bandwidth at a node. A node can perceive the channel as either idle or busy. The channel is idle if the node is not in any of the following three states: First the node is transmitting or receiving a packet. Second, the node receives a RTS or CTS message from another node, which monitors the channel for a period of time specified in the message. Third, the node senses a busy carrier with signal strength larger than a certain threshold, called the carrier-sensing threshold. But the node cannot interpret the contents of the message. By monitoring the amount of channel idle time, $T_{idle}$, during every period of time, $T_p$, the local available bandwidth $BW_{local}$ of a node $n_i$ can be computed using a weighted average [11] as follows:

$$BW_{local} = \omega \cdot BW_{local} + (1-\omega) \frac{T_{idle}}{T_p} \cdot BW_{channel}$$ (1)

Where $BW_{channel}$ is the capacity of the channel and weight $\omega \in [0,1]$.

Figure 3. Different sensing ranges of a mobile node

3.2.2 Estimation of the neighborhood available bandwidth

Each node perceives the network in a different state. Hence a node's local available bandwidth cannot provide information about its contention neighbors, since it does not know the amount of
BW\textsubscript{local} at other nodes\textsuperscript{3}. In our approach, during the normal medium access using IEEE 802.11, a node listens to the medium using a threshold value known as contention carrier sensing threshold. In Figure 3 the inner circle shows the transmission range of node A. Outer circles indicate the carrier sensing range of nodes B, A and C respectively. Normally carrier sensing range is twice the transmission range of a node. Contention carrier sensing threshold refers the range that covers the carrier sensing ranges of all of the sensing nodes contention neighbors. Hence it is set to a value much lower than the carrier sensing threshold. When the signal strength of the carrier sensed by a node is smaller than the contention carrier sensing threshold there is no communication in its contention neighborhood and contention neighbors of the node experience idle channels. The amount of time that the channel is in this idle state denoted as $T_{idle}^{contention}$, for every period of time denoted $T_p$, the neighborhood available bandwidth $BW_{neigh}$, is calculated using the following formula:

$$BW_{neigh} = \omega BW_{neigh} + (1-\omega)(\frac{T_{idle}^{contention}}{T_p})BW_{channel}$$ (2)

Where $BW_{channel}$ is the capacity of the channel and weight $\omega \in [0,1]$.

### 3.2.3 Estimation the Application’s Flow Bandwidth Consumption ($BW_{a\_flow}$)

We must quantify the bandwidth that a new flow requires so that it can be decided whether the bandwidth available will satisfy the requirements of the flow. Foremost, the applications data rate has to be converted into the corresponding channel bandwidth requirement. As per IEEE 802.11, for every application data packet, the MAC layer performs handshaking. During this RTS, CTS and ACK control packets are involved. Hence each data packet's transmission time is calculated as follows:\textsuperscript{4}:

$$T_{data} = T_{rts} + T_{cts} + T_{ack} + T_{dfs} + 3T_{sifs} + \frac{P + Q}{BW_{channel}}$$ (3)

Where:
- $T_{data}$: transmission time of each data packet
- $T_{rts}$: time for transmitting RTS
- $T_{cts}$: time for transmitting CTS
- $T_{ack}$: time for transmitting ACK
- $T_{dfs}$: DCF inter frame space defined in the IEEE 802.11 protocol standard
- $T_{sifs}$: short inter frame space defined in theIEEE 802.11 protocol standard
- $P$: size of the data packet
- $Q$: IP and MAC packet header length
- $BW_{channel}$: channel capacity

If at every second, the application generates $R$ packets with average packet size $P$, the corresponding channel bandwidth requirement is computed as follows:\textsuperscript{4}:

$$BW_{flow} = R \times T_{data} \times BW_{channel}$$ (4)

Next factor to be considered is multiple nodes on the route of a new flow may contend for bandwidth at a single location. Every such node needs bandwidth to be equal to $BW_{flow}$. The number of such kind of nodes is known as contention count ($C_{ct}$). Hence the bandwidth consumption of the flow [11] at this location is expressed as:

$$BW_{a\_flow} = C_{ct} \times BW_{flow}$$ (5)

### 3.3 Estimation of the Delay

#### 3.3.1 Estimation of the one hop delay

The method used to estimate the delay end-to-end present in [12]. Each node in the network receive a new RREQ is used the equation (D(i)) [12][15] to estimate the delay of one hop. The one hop delay encountered by a packet from node $m$ to node $n$ can be divided into the following components: the queuing delay, the contention delay, the transmission delay and the propagation delay.

\textsuperscript{3} The node knows only the bandwidth consumed by the neighbors with flow state packet shared periodically.
\[ D(i) = \text{propagation delay} + \text{transmission delay} + \text{queueing delay} + \text{contention delay} \quad (6) \]

The propagation delay is influenced by the distance between node \( m \) and node \( n \). Since the value of the propagation delay is quite small compared to the other delays, we neglect the effect of the propagation delay to the one hop delay \( D(i) \). The transmission delay is the duration of a successful packet transmission at the physical medium. We assume that packet sizes are fixed for all nodes, the transmission delay is a constant. We have:

\[ \text{Transmission delay} = \frac{\text{packet size}}{\text{data rate}} \quad (7) \]

The queuing delay is the interval between the time that a packet arrives at the node and the time that the packet becomes the head of line packet in the node’s queue. We can estimate the queuing delay by counting the number of existing packets in the outgoing buffer \( P_{\text{num}} \). Assume the channel is idle, we have

\[ \text{queueing delay} = P_{\text{num}} \times \text{transmission delay} \quad (8) \]

The contention delay is the interval between the moment when the packet becomes the head of line packet and the moment when the packet is actually in transmission on the physical medium. It captures the fact that when a packet becomes the head, the node may need to backoff before transmitting the packet on the physical medium. The contention delay can be influenced by many factors. The backoff probability and backoff time can be estimated by the number of neighbor nodes \( N_{\text{num}} \) and the average channel busy time \( T_{\text{busy}} \) respectively. The larger the value \( N_{\text{num}} \) is (i.e., the nodes compete to get the channel more intensively), the larger the value \( T_{\text{busy}} \) is (i.e., the nodes wait more time to send packets). Both of them lead to increased contention delay.

Because of the queuing delay, the contention delay and the transmission delay occur at the node. We propose node delay \( N_{\text{delay}} \) as a metric to describe the delay a packet spends to pass the node. This value can reflect the current status of the node.

From the above analysis, we can see that the node delay is affected by the following three factors: \( P_{\text{num}}, N_{\text{num}} \) and \( T_{\text{busy}} \). The estimation of these three factors is updated every \( T \) seconds. We use these three factors to calculate \( N_{\text{delay}} \) as follows

\[ N_{\text{delay}} = \alpha P_{\text{num}} + \beta N_{\text{num}} + \mu T_{\text{busy}} \quad (9) \]

Where \( \alpha \) and \( \beta \) and \( \mu \) are weighted factors. Each node updates \( N_{\text{delay}} \) every \( T \) seconds.

### 3.3.2 Estimation of the multi hop delay

End-to-End Delay (equation end-to-end Delay (route)) estimated by adding the values estimates calculated for each hop in the route between the source node and the destination node.

\[ \text{Delay end-to-end (route)} = \sum D(i), i \in \text{route} \quad (10) \]

### 3.4 Cost Function (Weight function)

The MCOP problem (Multi-Constraint Optimal Path problem) is introduced when two QoS metrics [14] are used. How to choose the Best Source Route between the source node and the destination node using the two QoS metrics?

We propose a solution for this MCOP problem by constructing a cost function. This function is defined for route selection\(^4\), which is the weighted sum of estimated delay and estimated available bandwidth. The cost of a node \( i \) is given by algorithm 3.

\(^4\) The best route between a source node and destination node selected by the cost function
Algorithm 3. Cost_per_hop (ni,f,D,B,R,BW_channel)

\[ C(B) = (1 - \frac{\text{available bandwidth estimated} (ni)}{BW_{\text{channel}}}) \]
\[ C(D) = \frac{\text{delay estimated}(f)}{D} \]
\[ C(\text{link}) = \alpha C(B) + \beta C(D) \]

Return (C(link))

ni : Node i  
B : Parameter of the bandwidth requested by the source  
D : Parameter of the delay requested by the source  
BW_channel: Total Bandwidth of the channel

3.5 Integration in DSR protocol

DSR protocol is improved by adding:
- Algorithm for estimating the bandwidth and algorithm for estimating the delay.
- Coupling between the two QoS metrics (delay, bandwidth) to use in the route selection (Cost Function).
- Extension of DSR packet and addition another packet.

For routing on implements an algorithm based on two QoS metrics with three tests:
- Tests concerning the source node.
- Tests concerning intermediate nodes.
- Tests concerning destination node.

3.5.1 Extension of DSR packet

We propose a modification in the DSR packet format (Figure 4) by adding new fields to this packet

Route metrics: it is a structure of field type that is used
To transport application requests (bandwidth required, delay required, packet size, flow rate) and the route metric (cost, estimated delay, path delay).
Packet ID: is an integer field that is used to identify each DSR packet sent in the network.

3.5.2 The packet shared periodically

When the estimation models used, requires the information exchanged periodically\(^5\) between neighboring nodes we create a new packet (Figure 5) that is shared periodically between neighboring nodes. This packet is called flow state packet and contains a single field:

\(^5\) In the DB-DSR protocol the neighboring nodes sharing your flow state information periodically.
Flow state information: it is a field of type structure that is used to transport the information of the bandwidth consumed by flows reserved in the node and other local information.

3.6 DB-DSR Procedure

DB-DSR relies on two procedures say route discovery and route reservation which details are given below.

3.6.1 Route Discovery

DB-DSR protocol routing operates on the same principles as DSR protocol. First new fields were added in the DSR packet and we create a new packet. Moreover, according to the node’s type, three different algorithms may be executed as explained below.

a. Source node algorithm
When a source node wishes to send the data packets, it first checks its resource availability such as bandwidth. If there is no resource available, the route discovery is canceled and the upper layer is informed. If the source node has sufficient resource, it checks the route in the route cache to see whether there is a route to the destination that satisfies the traffic QoS requirement. If so the source node sends the data packets and then makes the update for reserved flow at each hop in the source route. If no valid route is stored in the route cache, the source node keeps the information in DSR packet with RREQ option, initializes the QoS function and broadcasts the RREQ packet to all of its neighbors.

b. Intermediate node algorithm
This algorithm is executed by the intermediate nodes on receiving a RREQ packet. We can explain the process of intermediate nodes in the following steps:
- on receiving RREQ packet it looks into its cache to find out whether it has some route to the destination, if so it replies to the sender by sending route reply RREP, containing route. If intermediate node does not have route, this node extracts the information of route metrics from a received packet then starts the algorithm for estimating the bandwidth and delay, finally performs an admission control to determine acceptance or rejection of received RREQ. Details are given the algorithm depicted below.

---

6 This information place in route metrics field
Algorithm 4. Control Admission RREQ \((f, B_L, B_N, B_D)\)

If \(\text{destination_node}(f)\) Then
\(\alpha_i = \text{hop_number}(f)\).
Else
\(\alpha_i = \text{hop_number}(f) + 1\).
EndIf

\(\begin{align*}
B_L & : \text{local available bandwidth } (ni, \alpha) \\
B_N & : \text{neighborhood available bandwidth } (ni, \alpha) \\
B_D & : \text{Delay estimate}(f)
\end{align*}\)

If \((B_L \geq |f_B| \text{ And } B_N \geq |f_B|) \text{ And } B_D < |f_D|\) Then
\(\begin{itemize}
  \item Admission Control of node ni accept \(f \rightarrow \text{(RREQ)}\).
  \item Node ni made the necessary operations (calculate cost, updated the information exists in DSR packet with RREQ option).
  \item ni rebroadcasts \(f\).
\end{itemize}\)
Else
Admission Control of node ni reject \(f\), finally drop \(f\).
EndIf

\(|f|\): flow required by the source \{delay, bandwidth\} required
\(|f_B| = \text{value of the bandwidth requested by the source}
\(|f_D| = \text{delay estimated by ni (delay of partial path between and node ni)}

|f_D|\): value of local available bandwidth in the node ni
|f_B|: value of neighborhood available bandwidth in the node ni

3.6.2 Route Reservation

It is performed by the source node when receiving a RREP packet. The reservation [10] is processed by the first data packet sent. Nodes check that their resources are available, if not or if a link failure occurs, an error packet (RERR) is sent.

4. Simulation Results

To demonstrate the effectiveness of our proposal, we simulate our protocol DB-DSR and compare with the DSR standard protocol using the tool OPNET 14.0[16].

4.1 Simulation model and parameters

The network model that we design to simulate and evaluate our proposal consists of 50 mobile nodes of type MANET (Figure 6) placed randomly in an area of simulation 2000M \(\times\) 2000M. The mobility model we used is the model RWP (Random Way Point), interference range 250 meters, pause time is 5 seconds, packet size 128 bytes, simulation time is 900 seconds.

\[\text{Same RREQ packet: sent by different intermediate nodes with the same source.}\]
4.2 Results and discussions

The performance metrics we have chosen to evaluate the performance of our protocol DB-DSR are:
- Average End to End Delay
- Average Throughput
- Average Jitter
- Average Data Dropped

4.2.1 Average End to End Delay

The results of simulation shown in Figure 7 illustrate the average end to end delay. We can observe that the average end to end delay in DSR protocol is much higher compared to DB-DSR protocol, because nodes in DB-DSR only accept routes that satisfies the constraints of QoS (delay and bandwidth) requested by the sources when sending data packets. It shows that the average end to end delay in our protocol is approximately 0.001 s at every time of simulation. But in DSR, the average end to end delay increases with the time of simulation to reach a peak 0.018 s in minute 8. This is because in DSR the nodes choose routes according to the conventional criterion; the shortest path as well as the routes are overloaded, which engenders increasing propagation and transmission delays. So it yields important average end to end delay.

4.2.2 Average Throughput

Figure 8 shows the simulation results and illustrates the average throughputs. We observe that the average throughput in DSR protocol is lower compared to DB-DSR protocol, because the nodes in DB-DSR perform admission control at every node to verify QoS satisfaction. So it accepts only the routes that satisfy the QoS constraints requested by sources (guarantee of a good delivery of data packets between the communicating entities with routes providing necessary bandwidth without a loss). It also shows that the average throughputs in our protocol increases with time of simulation to reach peak 50000 bits in minute 3 and stays constant until the end of simulation. But in DSR, nodes choose routes...
according to the conventional criterion, the shortest path as well as the routes are overloaded, which engenders congestion in the network, then the average throughput decreases.

4.2.3 Average Jitter

The simulation results shown in Figure 9 illustrate the average jitters. We observe that the average jitter in DSR protocol is high compared to our protocol, because nodes in DB-DSR choose the most stable routes in terms of delay and bandwidth, the choice of route is subject to the delay and bandwidth constraints which must not exceed 150 ms. The route chosen respects delay and bandwidth constraints. Therefore, obtained jitter does not exceed 0.01 s in most of the time. But in DSR, the average jitter increases with the time of simulation to reach a peak 0.03 s in minute 5 and stays constant until the end of simulation. And nodes choose routes without any constraint of QoS. The shortest paths as the routes are perturbed by other connections which cause an increase in average jitter.

4.2.4 Average Data Dropped

Figure 10 shows the average dropped data packets. The average dropped data packets in DSR is much higher compared to the proposed protocol. It is approximately 200 bits at every time of simulation, but in DSR protocol it increases with the time, because the DB-DSR protocol is used and the route selection criterion with the delay and bandwidth constraints. But in DSR, nodes choose routes based on the number of hops as the routes are overloaded (is approximately 2300 bits at every time of simulation), which creates a saturation of the outgoing buffer in each node and the lack of resources and congestion which causes an increase in average data dropped.

After analyzing the simulation results for performance metrics we considered most important to evaluate our proposal and after comparison of simulation results with DSR protocol, we can conclude that our proposal appears more efficient than DSR protocol in simulation conditions described in terms of most performance with the previous metrics.

5. Conclusion and future work

We revisited quality of service issues and solutions in ad hoc networks in general, and QoS routing in particular. We tackled the QoS extensions for the DSR protocol. For this purpose, three building blocks are added to the latter, namely: application requirements, resource estimation, and admission control. We propose a new approach for QoS aware routing to support multimedia and real time applications in mobile ad hoc networks called DB-DSR. DB-DSR considers with multiple QoS constraints such as delay and bandwidth to find the most feasible route from the source node to the destination node. It also selects the most stable links and routes in term of delay and bandwidth.

The simulation results obtained under the OPNET simulator shows that our protocol (DB-DSR) improves a good performance in terms of end to end delay, throughput, jitter, and data dropped.

For further research, we consider:
Extend the same work to another reactive protocol like AODV or proactive like OLSR and compare with DB-DSR.
- Implement a Cross Layer for the estimation models used in DB-DSR and add to the OPNET node model.
- Implement DB-DSR under Linux.
- Comparing delay and bandwidth constraints routing with a DSR based multipath routing protocol like MP-DSR.
- Testing the scalability of the protocol according to the number of nodes and to the number of flows.

6. References