A reactive QoS Routing Protocol for Ad Hoc Networks

Stéphane Lohier, Sidi-Mohammed Senouci, Yacine Ghamri Doudane and Guy Pujolle
LIP6 - Université de Paris VI
8, rue du Capitaine Scott 75015 Paris – France
{Stephane.Lohier, Sidi-Mohammed.Senouci, Yacine.Ghamri, Guy.Pujolle}@lip6.fr

ABSTRACT
Due to bandwidth constraint and dynamic topology of mobile ad hoc networks, supporting Quality of Service (QoS) is a challenging task. In this paper we present a complete solution for QoS routing based on an extension of the AODV (Ad hoc On Demand Vector Distance) routing protocol that deals with delay and bandwidth measurements. The solution is based on lower layer specifics. Simulation results shows that, with the proposed QoS routing protocol, end to end delay and bandwidth on a route can bee improved in most of cases.

KEYWORDS
MANET, QoS routing, AODV, IEEE 802.11.

INTRODUCTION
The increasing progress of wireless local area networks (WLAN) has opened new horizons in the field of telecommunications. Ad Hoc networks are multi-hop wireless networks where all nodes cooperatively maintain network connectivity without fixed infrastructures and centralized administration. Due to their capability of handling node failures and fast topology changes, such networks are usually needed in any situation where temporary network connectivity is needed, such as in battlefields, disaster areas, and meetings. These networks provide mobile users with *ubiquitous* communication capability and information access regardless of location.

Throughputs reached today by Mobile Ad hoc NETworks (MANET) [1] enable the execution of complex applications such as multimedia applications (video conference, visiophony, etc.). However, these applications consume significant amounts of resources and can suffer from an inefficient and an unfair use of the wireless channel when they coexist with bursty data services. A lot of work has been done to support QoS on the Internet. However, none of these works can be directly used in MANET due to their specifics. Therefore, new specific QoS solutions need to be developed taking into account the dynamic nature of ad hoc networks. Since ad hoc networks should deal with the limited radiorange and mobility of their nodes, we believe that the best way to offer QoS is to integrate it in routing protocols. Such protocols will have to take into consideration QoS requirements, such as delay or bandwidth constraints, in order to select the adequate routes.

In this paper, we present a complete solution to the QoS routing problem based on an extension of the AODV (Ad hoc On Demand Vector Distance) routing protocol [2]. This solution consists of tracing routes in a reactive way by taking into account the QoS requirements (in terms of bandwidth, delay or both) associated with each flow. This work is inspired from the proposal of QoS extensions made in [3] in which we add QoS loss notifications, and delay/bandwidth measurements. The delay (resp. available bandwidth) are measured based on MAC and PHY layer specifics. Based on these measurements on each node/link, an end-to-end cumulative delay or available bandwidth can be estimated, which will enable route selection. This selection uses only the QoS extensions proposed for AODV. No additional signaling is required.

The rest of the paper is organized as follows. In section 2, we introduce the ad hoc routing issues. Section 3 abstracts the AODV protocol. Section 4 describes our solution and section 5 summarizes simulation results. Section 6 concludes the paper and presents some perspectives.

QOS IN AD HOC NETWORKS
The quality of service in ad hoc networks can be introduced in several interdependent levels [4]:

At the *medium access protocols (MAC) level*, by adding QoS functionalities to the MAC layer in order to offer guarantees [5];

At the *routing protocols level*, by looking for more performing routes according to various criteria (in this study we are interested more particularly by this approach);

At the *signaling level* with routing protocol-independent resource reservation mechanisms. The QoS at the signaling level is responsible of the coordination between other QoS levels as well as other components, such as scheduling or admission control (cf. Figure 1).

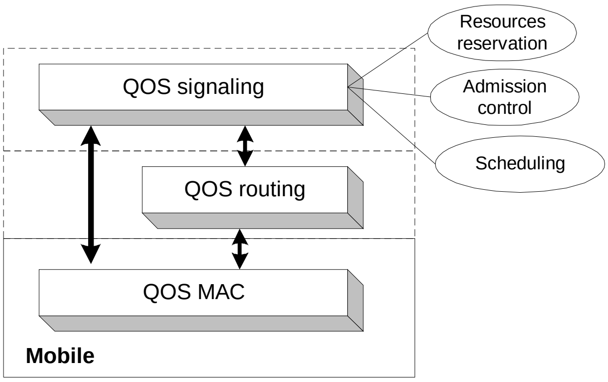

Figure 1. QoS Model.

The QoS routing objective is to find a route with enough available resources to satisfy a QoS request. Resource reservation on the optimum route, evaluated by the routing protocol, is generally carried out by the signaling layer. The QoS routing in ad hoc networks can be introduced from existing ad hoc routing protocols like AODV, by extending it with the help of mechanisms that allow differentiating end-to-end paths according to chosen metrics (*delay*, *throughput* or *cost*[1]). The advantage of such a solution is to avoid a systematic overhead when QoS is not required.

Among the proposed QoS models, we distinguish a class of solutions called "*soft QoS*" [6]. The basic idea is that if the QoS is guaranteed as long as the path remains valid, it is possible to tolerate, depending on application requirements, transition periods that correspond to route reorganizations. During these periods, the service is only "best effort". This class of solutions seems to be the most suitable for ad hoc networks allowing to offer QoS with a reduced complexity and overhead.

AODV PROTOCOL

AODV is a reactive ad hoc routing protocol which uses a broadcast route discovery mechanism. When a route is established, the nodes which are not concerned with the active path do not have to maintain routing tables or to take part into the route-update process.

Route discovery

Each node maintains a temporary routing table with an entry for each active route that contains:

- destination IP address;
- destination sequence number;
- hop count (number of hop to the destination);
- next hop;
- list of precursors;
- lifetime of the route.

The route discovery process is initiated whenever a source node needs to communicate with another node for which it has no routing information in its table.

The source node initiates path discovery by broadcasting a route request (RREQ) packet to its neighbours. The RREQ packet contains the following fields: < source addr; source sequence number; broadcast id; dest addr; dest sequence number; hop count >

The pair < source addr; broadcast id > uniquely identifies a RREQ (the source broadcast id is incremented each time it issues a new RREQ). If a node has already received a RREQ, it drops the redundant RREQ and does not rebroadcast it.

When a node receives a new RREQ, it looks in its route table for the destination.

If it does not know any route or a fresh enough one (the dest sequence number received in the RREQ is greater than the destination sequence number stored in the table), the node rebroadcasts the RREQ to its own neighbours after increasing the hop count.

If it knows a fresh enough route or if the node is the destination, the node stores the new information transported by the RREQ and sends a route reply (RREP) back to the source.

Insofar as the destination node replies to the first received RREQ, only one end-to-end route will be established.

A RREP packet contains the following information:

< source addr; dest addr; dest sequence number; hop cnt; lifetime >.

When an intermediate node receives back a RREP, it updates his table and forward the packet to the source which begins to send data after the first received RREP. The source node will change the route if a new RREP teaches him a better one (greater destination sequence number or lower hop count).

To set up a reverse path and then to be able to forward a RREP, a node records the address of the neighbour from which it received the first copy of the RREQ. These reverse route entries are maintained for at least enough time for the RREQ to traverse the network and produce a reply to the sender.

In the same way, nodes have to store the direct route. As the RREP travels back to the source, each node along the path sets up a forward pointer to the node from which the RREP came.

Figure 2 shows a route discovery with a RREQ broadcast when no intermediate node has a valid route. Figure 3 recall the reply to the first RREQ received by the destination. Note that nodes, B, C…, which are not involved on the initiated route do not have to maintain a routing table.

[1] Number of hops, resources requested for each node, utilization ratio of the links, etc.

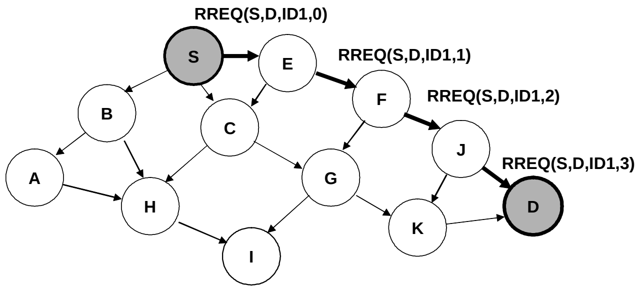

Figure 2. Route discovery broadcast initiated by the source.

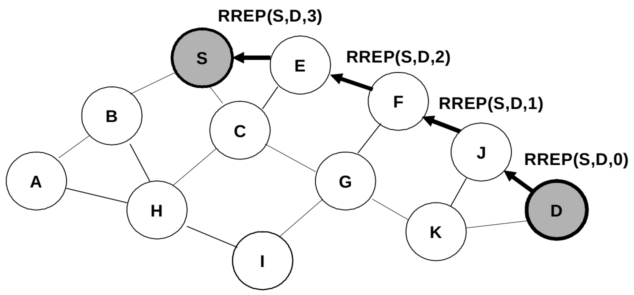

Figure 3. Route reply unicasted from the destination.

Route maintenance

If the initiated route breaks, due to node movement or failure, during an active session, the source has to reinitiate the route discovery procedure to establish a new route towards the destination. Periodic hello messages are used to detect link failures.

When a node detects a link break for the next hop of an active route (or receives a data packet destined to a node for which it does not have an active route), it sends a Route Error packet (RERR) back to all precursors. The RERR packet contains the following fields:

< unreachable dest ; unreachable dest sequence number >

When a node receives a RERR from a neighbour for one or more active route, it must forward the packet to the precursors stored in its table. Routes are erased by the RERR along its way. When a traffic source receives a RERR, it initiates a new route discovery if the route is still needed.

QOS EXTENSION FOR AODV

The QoS routing solution we propose uses two metrics: the delay and the available bandwidth. The QoS route is traced node by node using AODV QoS extensions [3]. For each crossed node, an estimate is made to know whether the maximum delay or minimum bandwidth requirements could be satisfied. If not (i.e. in the case where the delay estimate remains too long at an intermediate node or the available bandwidth too weak on a selected link), the route search will be interrupted. Thus, the QoS routing remains reactive, using only extensions on the AODV request (RREQ) and reply packets (RREP).

Delay estimation

The delay estimation uses one of the existing AODV parameters: the NTT (NODE_TRAVERSAL_TIME), initially considered as a constant [2]. In our proposal, the NTT becomes an estimate of the average one hop traversal time for a packet. It includes the transmission delay over the link and the processing time in the node (delays in queues, processes interruption time, etc).

As shown in Figure 4, the NTT parameter for node B is divided on 2 parts:

$$NTT_B = d_{AB} + t_{T\,B} \qquad (1)$$

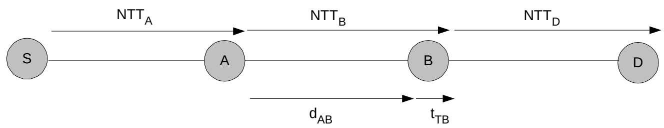

Figure 4. NTT estimation.

d_{AB} corresponds to the transmission delay between two adjacent nodes introduced by MAC and PHY level operations. For example, on an IEEE 802.11 [7] network, the transmission delay (d_{AB}) is due to the durations of frame transmission (RTS, CTS, data, ACK); to the inter-frame spacing (DIFS, SIFS), to propagation delays and to contention resolution (including possible retransmissions due to collisions).

As numerous MAC level protocols for ad hoc networks uses frame acknowledgments to ensure that no collision occurs during a frame transmission, we can define d_{AB} as the time difference between the time the packet is handled by the MAC layer in the source node and the time its acknowledgment is transmitted back by the destination node.

$$d_{AB} = T_{ACK} - T_{transmission} \qquad (2)$$

In order to keep only one time reference for the source node [8], we can take into account the propagation delay, between two nodes, for the acknowledgement. This parameter is a constant and its value depends on PHY layer specifics.

$$d_{AB} = T_{ACK\_reception} - T_{transmission} - T_{propagation} \qquad (3)$$

For the NTT calculation at the destination node, d_{AB} can be sent with another AODV extension.

The choice of doing the delay measurement using only RREQ and RREP packets rather than all data and routing packets is motivated by the processing overhead which is reduced when using passive measurement. Note that, the obtained delay d_{AB} depends closely on the packet size. A correction should therefore be made in order to take into account an average size instead of the RREQ or RREP packet lengths used for such measurements. For example, with a control-packet length of 32 bytes and with an average length of 100 bytes for data packets sent at 11Mbit/s, the correction could bee:

$$d'_{AB} = d_{AB} + \frac{(100-32)\times 8}{11.10} \qquad (4)$$

Insofar as route delays depend on unpredictable events (node movements, arrivals, extinctions, variations of streams and traffic, etc.), the variance of node-to-node delays can be significant. Two methods exist in order to take these delay variations into account [9]. The first one calculates an average according to a fixed size window. The second method consists of calculating an average, weighted by a forgetting factor (*exponential forgetting*). As our aim is to minimize the overhead, the second method is naturally more suitable. The delay between nodes *A* and *B* is then given as follows:

$$d_{AB}(t)=(1-\lambda)\sum_{k=0}^{\infty}\lambda^{k}.d_{AB}(t-k) \quad (5)$$

Where $\lambda \in [0,1]$ is the forgetting factor.

The processing time in the node (t_{TB}) includes a node-specific constant (corresponding to the processing capability of the packet at the different levels) and a variable delay, function of the packet number in the queue. A first estimation is done by computing the average number, over a sliding window, of the queued packets. The length of the window is based on another specific AODV parameter: ACTIVE_ROUTE_TIMEOUT. This first estimation gives satisfactory results and has to be compared to other more complex queuing delay estimators. This comparison is outside the scope of this paper and is a subject of a future work.

Note that to estimate the end to end delay, we take into account the processing time in the source node. For the destination node, as there is no forwarding, the queuing delay is not considered.

Bandwidth estimation

An estimate for the available bandwidth on a link can be formulated as follows [10]:

$$BW_{available} = (1-u) \times Throughput_{on\ the\ link} \quad (6)$$

Where *u* represents the link utilization.

To calculate the available bandwidth for a node, the link throughput must first be evaluated. An initial evaluation can be done simply by emitting packets and measuring the corresponding delays:

$$Throughput_{on\ the\ link} = \frac{S}{d_{AB}} \quad (7)$$

S being the packet size and d_{AB} the transmission delay between two adjacent nodes defined above. As for the delay estimation, it is necessary to limit the random aspect of the measurement. *Exponential forgetting* can also be used to calculate the average available bandwidth.

The link availability (*1-u*) is evaluated by the following formula:

$$1-u = \frac{idle\ times\ in\ window}{window\ duration} \quad (8)$$

Where '*idle times in window'* is the sum of all transmission idle times measured during a time sliding window of width '*window duration'*. The '*window duration'* is set to ACTIVE_ROUTE_TIMEOUT. Note that, this computation is done only if there are active routes stored in the node's routing table. Otherwise, *u=0*.

QoS Routing

For each route entry corresponding to each destination, the following fields are added to the routing tables:

- Maximum delay;
- Minimum available bandwidth;

An extension is foreseen by AODV for its main packets RREP and RREQ (cf. figure 5).

8 bits	8 bits	n bits
Type	Length	Type-specific data…

Figure 5. AODV Extension format.

Depending on the packet type, a "delay" extension can have two meanings:

For an RREQ packet, it means the delay allowed for a transmission between the source (or an intermediate node forwarding the RREQ) and the destination;

For an RREP packet, it means an estimate of the cumulative delay between an intermediate node forwarding the RREP and the destination.

Thus, a source requiring maximum delay constraint transmits a RREQ packet with a QoS delay extension. Before forwarding a RREQ packet, an intermediate node compares its NODE_TRAVERSAL_TIME with the remaining delay bound indicated in the extension. If the delay bound is inferior, the packet is discarded and the process stops. Otherwise, the node subtracts its NTT from the delay bound provided in the extension, updates the QoS delay extension, and propagate the RREQ as specified by AODV (cf. section 3.1). In the example of Figure 6, each node in the route satisfies the comparison and the requested delay at the destination (50ms-10ms) remains greater than zero.

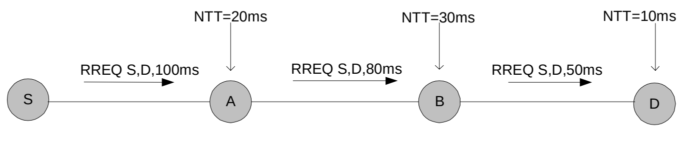

Figure 6. Example of QoS delay request.

In response to a QoS request (RREQ), the destination sends an RREP packet (cf. Figure 7) with an initial delay corresponding to its NTT. Each intermediate node adds its own NTT to the delay field and records this value in the routing table for the concerned destination before forwarding the RREP. This entry update allows an intermediate node to answer the next RREQ simply by comparing the maximum delay fields of the table with the value of the transmitted extension. The answer of the intermediate node is always valid in time because the old routes are deleted from the table according to the ACTIVE_ROUTE_TIMEOUT parameter.

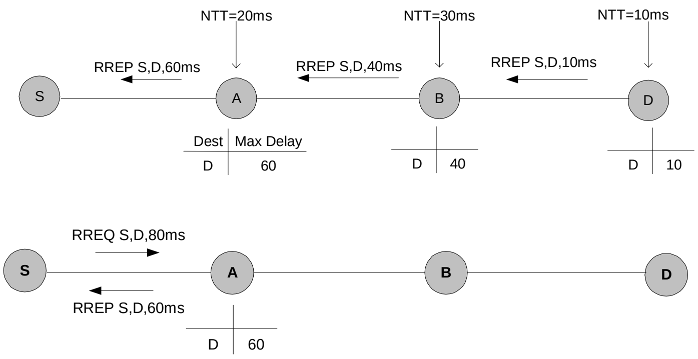

Figure 7. Examples of QoS delay responses.

For a “bandwidth” extension, the principle remains the same. A source requiring a bandwidth constraint transmits a RREQ packet with QoS bandwidth extension. Thise extension indicates the minimum bandwidth having to be available on the whole path between the source and the destination. Before forwarding the RREQ packet, an intermediate node compares its available bandwidth to the bandwidth field indicated in the QoS extension. If the bandwidth required is not available, the packet is discarded and the process stops. In response to a QoS request, the destination sends a RREP packet with its measured available bandwidth. Each intermediate node, forwarding the RREP, compares the bandwidth field of the extension with its own available bandwidth on the selected route and keeps the minimum between these two values to propagate the RREP. This value is also recorded in the routing table for the concerned destination. It indicates the minimum available bandwidth for the destination (see example on figure 8). This information remains valid as long as the route is valid (lifetime < ACTIVE_ROUTE_TIMEOUT).

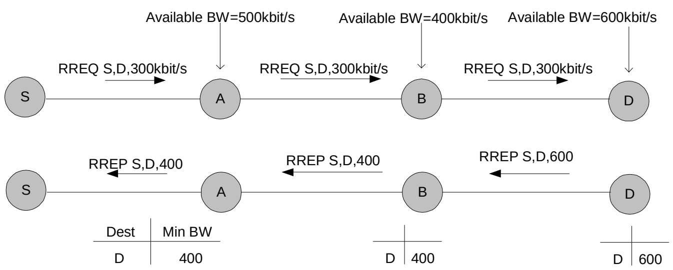

Figure 8. Example of QoS bandwidth request and response.

If the QoS request concerns both delay and bandwidth, the two extensions can be appended to the same request and reply packets. In this case, both maximum delay and available bandwidth verifications of request (RREQ) and reply (RREP) will be applied simultaneously. RREQ packets are discarded if one of the constraints cannot be satisfied.

To prevent an eventual variation of the NTT on a node and a possible lost of QoS, a predefined QoS Delay Margin (says QDM) can be introduced. A route error packet (RERR) is generated when an intermediate node detects an increase in its NTT that is greater than QDM. The RERR packet is also generated if the node detects a decrease in its available bandwidth that is greater than a QoS Bandwidth Margin (says QBM).

Note that, if the margin is chosen too large, the source node will never be informed of QoS loss. Conversely, if the margin is too small, useless RERR packets can be generated causing new RREQ broadcast. This undesirable control packet transmission induces an undesired overhead, slowing down data packet exchanges, even if the QoS constraint is initially respected. So, an accurate dimensioning of these margins is very important.

As for standard AODV route error mechanism, the RERR packets are sent to all the precursors stored for all the routes (cf. Figure 9). Note that the NTT or/and the available bandwidth are measured each time a RREQ or a RREP packet is received by a node, which generally corresponds to a change of the network and traffic load (new source, node mobility…) producing a possible loss of QoS.

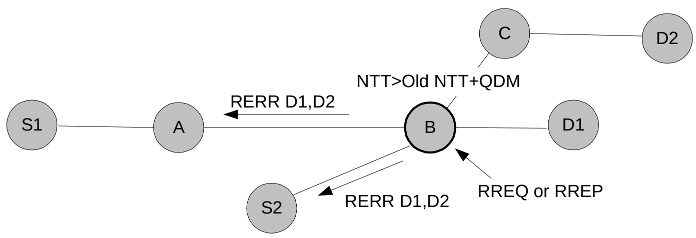

Figure 9. Example of QoS delay lost.

PERFORMANCE EVALUATION

In order to evaluate the performances of our QoS routing protocol, we simulate the proposed mechanisms using NS-2 [11] extended by a complete implementation of IEEE 802.11 [12].

The radio model allows a bit rate of 2 Mbit/s and a transmission range of 250 m. The number of mobile nodes is set to 20 or 50 nodes giving two simulation sets. These nodes are spread randomly in a 670×670m area network and they move to a random destination every 30 s with a speed randomly chosen between 0 and 10 m/s. Simulations run for 300 s. Traffic sources are constant bit rate (CBR) sending 8 packets of 512 bytes per second. Several simulations are realized by varying the source node percentage from 10% to 100%. The QoS constraint is set to 100ms for delay and 100kbit/s for bandwidth. For the QoS loss detection mechanism, the optimal margins are 30ms for QDM and

30kbit/s for QBM. First simulation results shows that these values are optimal and give a good compromise between overhead due to RERR generation and QoS loss detection.

The delay estimation uses a constant processing time in the node equal to 3 ms, a first order (k=1 in (1)) variance correction is applied and the optimum forgetting factor λ is set to 0.2. ACTIVE_ROUTE_TIMEOUT is a constant set to 10s. The performance of our algorithm has been evaluated by measuring the average end to end delay, the average throughput, and the overhead induced.

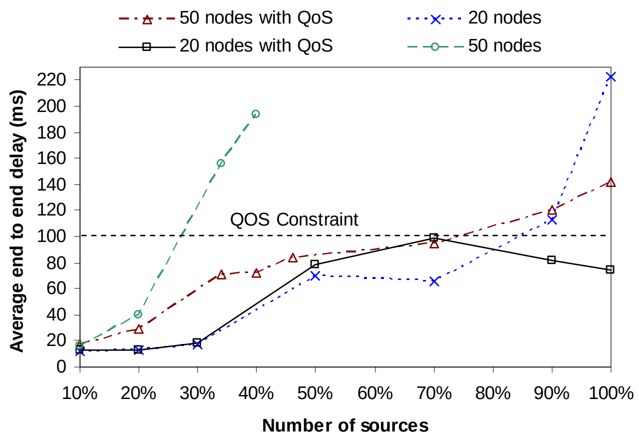

Figure 10. Average end to end delay / Number of sources.

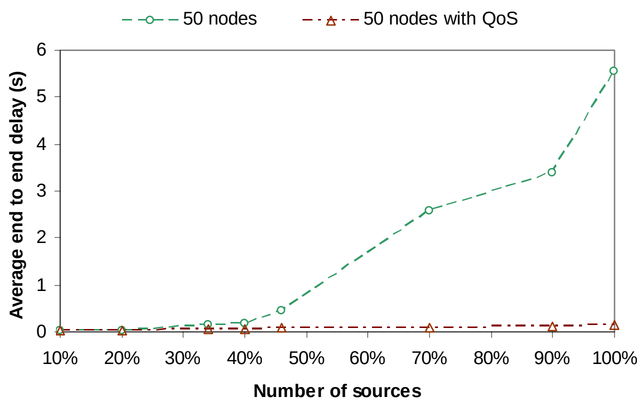

Figure 11. Average end to end delay / Number of sources (50 nodes).

Figures 10 and 11[3] present the average end to end delay for data packets on all QoS routes when the number of QoS sources varies. On a 20-nodes network, the delay remains lower than 100ms with QoS routing whatever the number of sources. On a high density (50 nodes), the QoS constraint of 100ms is respected with QoS routing if the number of sources does not exceed 70%. Without QoS, the delay can reach several seconds (figure 11).

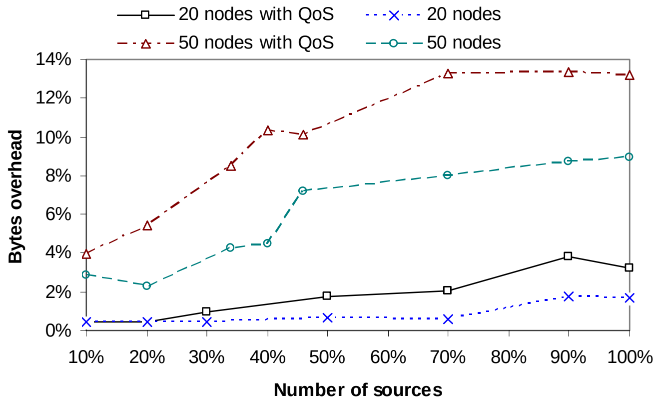

Figure 12. Overhead / Number of sources (delay constaint).

The overhead[5] due to the AODV control messages is slightly higher when using QoS extensions (cf. Figure 12). The increase of this overhead when using QoS extensions remains lower than 6% whatever the density of the network and the number of sources.

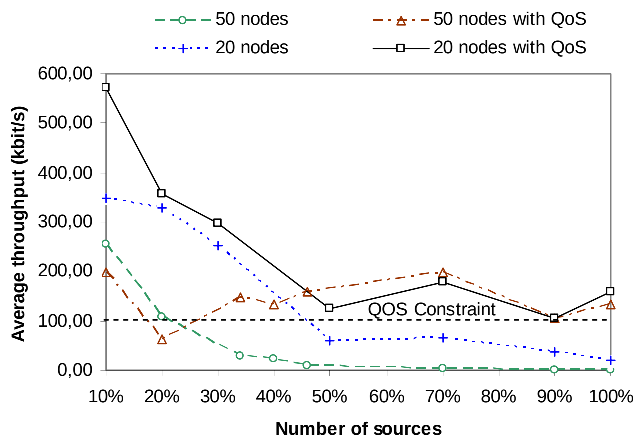

Figure 13. Average throughput / Number of sources.

[3] In Figure 11 the scale is given in s, rather than ms (Figure 10), for better visibility.

[5] The overhead is computed as the bandwidth percentage consumed by the control packets (RREQ, RREP, and RERR).

Figure 13 presents the average throughput on all QoS routes when data packets are sent from a source to a destination. For a 20-nodes network, the bandwidth constraint is respected whatever the number of sources. On a high density network (50 nodes), the QoS routing becomes more efficient when the number of sources, and then the traffic, increases. Note that without QoS extensions, the throughput becomes quickly very weak when the number of sources increases (under the 100kbit/s requested when there are more than 20% of sources).

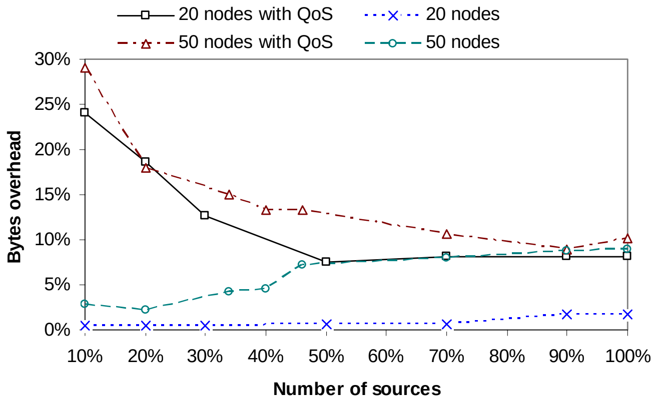

Figure 14. Overhead / Number of sources (bandwidth constaint).

The overhead is higher when using the QoS extension especially for a small number of QoS sources (cf. Figure 14). The traffic generated by AODV control packets, and particularly RERR packets, is relatively important in this case. For a dense network with high offerload, the overhead becomes equivalent. This is a logical result since all new demands are quickly rejected, even by the first encountered node: the overhead is then considerably reduced.

The last results show that the QoS routing algorithm with bandwidth extensions is more suitable for a high density network with an important traffic.

CONCLUSION

In this paper, we have proposed and evaluated a QoS routing solution based on AODV. This solution uses delay and bandwidth measurement and preserves the reactive nature of AODV.

The QoS routes are traced node by node and the proposed routing algorithm uses extensions of the AODV request (RREQ) and reply (RREP) packets. The delay and bandwidth measurements are initiated only on RREQ or RREP arrivals in a node (these times correspond to a network state change: arrival of a new flow). Note that measurements on each data or routing packet would increase the overhead unnecessarily. Corrections are however made in order to take into account variations due to the dynamic nature of ad hoc network and network traffic.

The proposed QoS routing with QoS loss notification gives satisfying results, especially for the delay extension. For the bandwidth extension, good performances are obtained for high density networks with an important offerload. New algorithms for bandwidth measurement and queuing-delay estimation are currently under study. This study will allow choosing the best estimators to run with the proposed QoS routing protocol.

REFERENCES

1. IETF MANET WG (Mobile Ad hoc NETwork) www.ietf.ora/html.charters/manet-charter.html.
2. C. E. Perkins, E. M. Royer, and S. R. Das, “Ad hoc on-demand distance vector routing,” *Internet Draft*, 2002.
3. C. E. Perkins, E. M. Royer "Quality of service for ad hoc on-demand distance vector routing". *IETF Internet Draft*.
4. Kui Wu and Janelle Harms “QoS Support in Mobile Ad Hoc Networks” Computing Science Department University of Alberta.
5. A. Veres, Campbell, A. T, Barry, M and L-H. Sun, "Supporting service differentiation in wireless packet using distributed control", *IEEE Journal of Selected Areas in Communications (JSAC)* Vol. 19, No. 10, pp. 2094-2104, October 2001.
6. H. Xiao, K.G. Seah, A. Lo and K.C. Chua, "A flexible quality of service model for mobile ad-hoc networks", IEEE Vehicular Technology Conference (VTC Spring 2000), Tokyo, Japan, May 2000, pp. 445-449.
7. LAN MAN Standards of the IEEE Computer Society. “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification”. IEEE Standard 802.11, 1997.
8. Kay Römer, "Time Synchronization in Ad Hoc Networks." In Proceedings of MobiHoc 2001, ACM, Oct. 2001.
9. P. Gupta and P. R. Kumar "A system and traffic dependent adaptive routing algorithm for ad hoc networks" , *Proceedings of the 36th IEEE Conference on Decision and Control*, pp. 2375--2380, San Diego, Dec. 1997.
10. M Kazantzidis, “End-to-end versus explicit feedback measurement in 802.11 networks,” *Technical Report N° 010034 UCLA Computer Science WAM Lab*,. 2001.
11. The Network Simulator - NS-2 : http://www.isi.edu/nsnam/ns/
12. The Monarch Project : http://www.monarch.cs.rice.edu/