# Implementation and Performance Evaluation of a Quality of Service Support for OLSR in a Real MANET

Dang-Quan Nguyen
INRIA Rocquencourt
78153 Le Chesnay, France

Pascale Minet
INRIA Rocquencourt
78153 Le Chesnay, France

Cedric Adjih
INRIA Rocquencourt
78153 Le Chesnay, France

Thierry Plesse
DGA/CELAR
35174 Bruz, France

## ABSTRACT

Military applications in tactical networks have quality of service (QoS) requirements. The purpose of this paper is to show that these requirements can be fulfilled by our QoS support mechanisms, implemented in a MANET based on the IEEE802.11b and the OLSR protocols. After a brief presentation of our solution, we describe its implementation on a MANET testbed made of 18 routers. Results of performance evaluation obtained from measurements are reported and compared with simulation's. They allow to conclude that this QoS support improves the quality perceived by the users in terms of delivery rate and granted throughput. The flooding optimization present in OLSR is also preserved. With QoS support, simulation results show that the network is able to admit more user flows than without it. The overhead of this QoS support is evaluated and extensive simulations also show that it is scalable to large networks.

## Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*routing protocols*

## Keywords

routing, quality of service, QoS, MANET, OLSR

## 1. CONTEXT AND MOTIVATIONS

A MANET, Mobile Ad hoc NEtwork, is a collection of autonomous mobile nodes communicating over a wireless medium without requiring any pre-existing infrastructure. These nodes are free to move about arbitrarily. MANETs exhibit very interesting properties: they are self-organizing, decentralized and support mobility. Hence they are very good candidates for tactical networks in military applications. In order to allow the communication between any two nodes in a multihop MANET, a routing protocol is required. The IETF MANET working group has standardized four routing protocols. Among them, OLSR (Optimized Link State Routing) [1] is a proactive protocol where nodes periodically exchange topology information in order to establish a route to any destination in the network. A comparative performance evaluation between OLSR and AODV is reported in [9].

OLSR [1] is an optimization of a link-state routing protocol. It is based on the concept of *multipoint relays* (*MPRs*). Using *MPRs* reduces the size of control messages: instead of declaring all links in the neighborhood, a node only declares the set of links to the neighbors that have selected it as *MPR*. The use of *MPRs* also minimizes flooding of control traffic, known as *MPR flooding*. Indeed only *MPRs* forward flooded messages. This technique significantly reduces the number of retransmissions of any flooded message [11]. An OLSR node detects its one-hop and two-hop neighbors by means of periodic *Hello* messages. This node independently selects its own set of *MPRs* among the one-hop neighbors in such a way that the *MPRs* cover all two-hop neighbors. Each node also maintains information about the network topology obtained from *Topology Control (TC)* messages, generated by *MPRs* and disseminated by *MPR flooding*. The routing table is computed using Dijkstra's algorithm.

More and more applications have Quality of Service (QoS) requirements in terms of minimum throughput, maximum end-to-end delay or loss probability. With regard to military IP network's characteristics (strategic, operative and tactical), it is essential to provide an end-to-end QoS to the end-users (soldiers). The intrinsic qualities of MANETs make them very suitable for such mobile networks. As on the fixed networks, various types of user's traffic coexist, e.g.: data, voice and video. These types of traffic have different properties and military operational constraints. They must receive appropriate treatment according to their importance, e.g.: "flash" messages must be delivered as fast as possible. As a consequence, MANETs should support QoS in order to be considered for military tactical networks. That is why, we propose in this paper a solution to enhance OLSR with QoS support and report the performance evaluation obtained from real experimentations and simulations.

The introduction of QoS support into MANETs becomes complex when one considers MANET's specificities:

- **The existence of radio interferences makes the bandwidth reservation NP-hard [3]**. When a node $N$ transmits a packet, all distant nodes up to two hops from $N$ are backed-off, assuming that the interference caused by a transmitter is limited to two hops. Hence the bandwidth consumed by a flow on a node $X$ depends on how many times $X$ belongs to the interference area of a transmitter of

*QoSim* 2009, Rome, Italy

this flow. When the path of a flow $f$ is a straight line (e.g. shortest path), a node $X$ belongs at most five times to the interference area of $f$. Indeed this bound is reached at the third transmitter on the flow $f$, at least 5-hop long: one for the node $X$ itself, plus two for its two predecessors on the flow path and two more for its two successors.

- **Network resources in MANETs are scarce.** For example, the 802.11b medium bandwidth is limited to 11 Mbps. Mobile nodes also have low energy capacity and limited memory space. Therefore, one should consider not to use complex algorithms and resource demanding protocols when designing solution for MANETs.
- **MANET's topology is highly dynamic.** It is due to the variation of radio propagation and node mobility. New nodes may join or leave the network at anytime. Thus, radio links may form or break frequently.

Those specificities make the QoS support in MANETs very complex, and explain why traditional QoS supports designed for a wired network does not fit MANETs. As a consequence, our goal of QoS support in MANETs is not to provide hard QoS guarantees but rather to optimize resource utilization while trying to satisfy the requirements of real-time applications. The proposed QoS support should:
(i) be interference-aware,
(ii) use simple algorithms and protocols with low overhead,
(iii) support multihop routing and topology changes.

In this paper, we focus more particularly on the admission control and the QoS routing, both being interference aware. With regard to [6], we show how to compute the local available bandwidth and tune the leaky bucket parameters at each node in section 2. In section 3, we present an implementation of our QoS support on a MANET testbed made of 18 routers, each equiped with an 802.11b network interface and running the OLSR protocol. The performances of this QoS support are measured on this platform for different scenarios and reported in section 4. Those results are also compared with simulation's in which all the testbed scenarios are reproduced by our NS-2 simulator. In section 5, we give an analytical evaluation of the overhead of this solution. Extensive simulations in section 6 show that it is scalable to large networks. We conclude this paper in section 7.

## 2. THE PROPOSED SOLUTION

Many existing works concern QoS support in MANETs. Some of them ([2, 5]) set a framework for QoS support but do not consider the problem of radio interferences in MANETs. While other works ([4, 8]) propose interference-aware routing protocols for MANETs, their mechanisms lack of justification and quantitative analysis in order to be compared with other solutions. In our QoS support described in this section, we take into consideration the interference problem and justify our design choices. We also provide extensive performance analysis of our solution.

As resources are scarce in MANETs, our extension of OLSR [6] keeps the optimizations present in OLSR, which relies on two principles:

- a partial topology knowledge: the advertised link set is a subset of the whole topology;
- an optimized flooding, called MPR flooding: it is based on the concept of multipoint relays.

Our solution of QoS support includes five components as illustrated by Figure 1 and allows cross-layering between the MAC and routing layers. In order to select a route that meets a bandwidth requirement, two tasks are required.
(i) First, the routing protocol must know the amount of bandwidth available at each node, denoted $r_i$ with $i$ the index of the considered node. We recall that this bandwidth is shared with all nodes up to two hops. QoS signaling is used to disseminate this bandwidth information $r_i$, using *Hello* and *TC* messages with MPR flooding.
(ii) Secondly, the subset of links advertised by *TC* messages must contain all links to "best-bandwidth" one-hop neighbors, i.e. the one-hop neighbors having the largest $r_i$. We add a modified version of the MPR selection, named QoS-MPR selection, to consider the bandwidth criterion: each node $N$ selects its QoS-MPRs in such a way that each two-hop node is reached by a QoS-MPR providing the largest available bandwidth $r_i$. In our solution, QoS-MPR selection exists alongside with MPR selection. The former allows nodes to find routes that meet bandwidth requirements if they can be met. The latter ensures MPR flooding of the routing protocol. Thus, this QoS support preserves the flooding optimization using MPR.

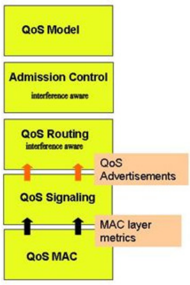

**Figure 1: Five components of QoS support.**

In [8], the authors replace MPR selection by QoS-MPR selection and use QoS-MPR to ensure flooding. We show in section 5 that this kind of flooding generates an amount of overhead that is significantly higher than that of MPR flooding (by 20% or 30% depending on the network size).

This extension of OLSR with QoS support, illustrated in Figure 1, would demonstrate better performances if a QoS MAC layer was used. An ideal QoS MAC layer should be deterministic, i.e. grant medium access to the waiting packet having the highest priority among all packets in the shared medium. Also, it should provide information on QoS parameters at MAC level, e.g.: the local available bandwidth $r_i$. Nevertheless, we show in section 4 that our QoS solution does improve the quality of user flows that rely on the protocol IEEE 802.11b, even in the absence of QoS MAC layer. A QoS MAC would provide a better differentiation between traffic classes.

### 2.1 QoS models

In our solution, we distinguish four classes of traffic, listed in decreasing priority order:

- **Control traffic**: this class contains routing protocol messages. They are not subject to QoS control admission process and are forwarded with the highest priority at any node.
- **Delay traffic**: this class contains user's data traffic that has delay requirements, like voice flows. They are subject to QoS control admission process. If accepted, delay traffic is forwarded with a priority higher than bandwidth traffic.
- **Bandwidth traffic**: this class contains user's data traffic that has throughput requirements, like video flows. They are

subject to QoS control admission process. Bandwidth traffic is forwarded with higher priority than Best-Effort traffic.

• **Best-Effort traffic**: this class contains all other types of traffic, which do not have QoS requirements.

In the following, we denote by *QoS flows* all user's flows having delay or throughput requirements. Best-Effort traffic is denoted *BE*.

## 2.2 QoS signaling

QoS signaling is in charge of local QoS parameters estimation. QoS signaling disseminates them to all other nodes. In this paper, the QoS parameter considered is bandwidth. The local available bandwidth of a node $N_i$, shared with its neighbors up to 2-hop shared available bandwidth is measured locally at the MAC level.

DEFINITION 1. *The bandwidth available at a node $N_i$, denoted $r_i$, represents the bandwidth that $N_i$ can use for additional transmissions.*

If the IEEE 802.11b protocol is used, then $r_i$ is equal to the percentage of time the medium is not busy and $N_i$ is not in back-off state, multiplied by the medium capacity. We will see in Section 3 how this value is obtained.

The available bandwidth is first disseminated locally using *Hello* and then globally to the entire network with *TC* messages using MPR flooding.

• A node broadcasts its $r_i$ and all $r_j$ of its 1-hop neighbors in *Hello* message. Thus, a node knows the value of $r_i$ of all nodes in its 1-hop and 2-hop neighborhood. This information is used to select the QoS-MPR set, as described later.

• A QoS-MPR node floods to the network its *TC* message, which includes:

- the local available bandwidth $r_i$ of its QoS-MPR selectors.
- the minimum of bandwidth available in its interference area, denoted as $\gamma_i$, i.e. $\min\{r_j\}$ over all 1-hop and 2-hop neighbors.

With QoS support, each node selects two sets of MPR:

• **Classical MPR set**. It is done as specified in the OLSR RFC [1].

• **QoS-MPR set**. A node $N_i$ selects this set in order to be reached from any 2-hop neighbor via a 1-hop neighbor having largest $r_i$. The selection of QoS-MPRs is inspired from the works in [7] and is described in the next subsection.

## 2.3 QoS-MPR Selection

In order to find routes that meet a bandwidth demand, each node must have the knowledge of the partial QoS topology of the network (i.e. partial topology enhanced with QoS information on the nodes). This QoS topology is required by QoS routing protocols designed for ad hoc networks such as QOLSR [8]. Each node selects its QoS-MPR set and floods them to the entire network. The idea of QoS-MPR selection is to extract from the set of 1-hop neighbors a minimum subset of nodes that have the largest available bandwidth $r_i$. This subset must cover all 2-hop neighbors. So that any node can be reached from any of its 2-hop neighbors via a 1-hop neighbor having the largest $r_i$.

• **QoS-MPR selection algorithm:**

1. Sort all 1-hop neighbors by decreasing order of $r_i$.
2. Consider each 1-hop neighbor in that order: this neighbor is selected as QoS-MPR if it covers at least one 2-hop neighbor that is still uncovered by all the selected QoS-MPRs. Continue until all 2-hop neighbors are covered.

## 2.4 QoS routing

The purpose of QoS routing is to find a route between a source and a destination that meets the throughput requirement of a QoS flow. It uses information of the partial QoS topology gathered by QoS signaling to select routes. As QoS routing must be interference-aware, we define in section 3.1 a rule to convert an application throughput requirement into the amount of available bandwidth needed at the 802.11b MAC level. This amount of bandwidth, denoted as $b$, takes into account all interferences generated by the flow's retransmissions on its route. Thus, a route $\mathcal{R}$ for a QoS flow is *eligible* if and only if any node $N_i$, with $N_i$ in $\mathcal{R}$ or in its interference area (i.e. distant at most 2 hops from a sender node in $\mathcal{R}$), has a sufficient available bandwidth: $\gamma_i \geq b$.

On the other hand, the shortest route tends to minimize the network resources used for retransmissions. Hence, our QoS routing algorithm selects the shortest route among all eligible routes. If tie, the shortest route having the largest $\min_{\mathcal{R}}\{\gamma_i\}$ is chosen. We apply Dijkstra's algorithm [10] to the partial QoS topology in which all nodes $N_i$ having $\gamma_i < b$ are not considered.

PROPERTY 1. *This algorithm finds a shortest route that meets the required amount of bandwidth between any two nodes, if it exists. If there are several shortest routes, the one having the largest available bandwidth is chosen.*

PROOF. We prove this property $\mathcal{P}_k$ by induction on the route length of $k$ hops. It is easy to check that $\mathcal{P}_k$ holds for $k = 1, 2$. Let assume that $\mathcal{P}_n$ holds. Let $\mathcal{R}^{[n+1]} = (N_0, N_1, \ldots, N_n, N_{n+1})$ a shortest route of $n + 1$ hops from $N_0$ to $N_{n+1}$, and $\mathcal{R}^{[n+1]}$ meets the bandwidth requirement $b$. From $\mathcal{P}_n$, the algorithm finds a shortest route $\mathcal{R'}^{[n]} = (N_0, N'_1, \ldots, N'_{n-1}, N_n)$ of $n$ hops from $N_0$ to $N_n$. $\mathcal{R'}^{[n]}$ meets the bandwidth requirement $b$ and has the largest available bandwidth. By the minimality of $\mathcal{R}^{[n+1]}$ and $\mathcal{R'}^{[n]}$ lengths, node $N'_{n-1}$ is a 2-hop neighbor of $N_{n+1}$. Let $N'_n$ the QoS-MPR of node $N_{n+1}$ that covers $N'_{n-1}$ and has the largest available bandwidth (i.e. $r'_n \geq r_n$). $N'_n$ exists by the selection of QoS-MPR at node $N_{n+1}$. The link $(N'_n, N_{n+1})$ is also known to all nodes by *TC* messages generated from $N'_n$. Therefore, the algorithm finds the route $\mathcal{R'}^{[n+1]} = (N_0, N'_1, \ldots, N'_{n-1}, N'_n, N_{n+1})$ of $n+1$ hops from $N_0$ to $N_{n+1}$. As $\mathcal{R'}^{[n+1]}$ has an available bandwidth larger than $\mathcal{R}^{[n+1]}$, $\mathcal{P}_{n+1}$ holds. This concludes the proof. ☐

If a route $\mathcal{R}$ is found, it is used to forward all packets of the QoS flow. Other QoS routing proposals like QOLSR [8] does not impose the route $\mathcal{R}$. They let the QoS flow's packets to be routed hop-by-hop through undetermined nodes once the QoS flow is accepted into the network. In section 4.2, we show by simulations that imposing the route $\mathcal{R}$ significantly improves packet's jitter and throughput stability. In practice we use the *source routing* option provided by the IP header to impose $\mathcal{R}$ to all QoS flow's packets. Thus, no extra signaling protocol is required.

In order to deal with resource releases or changes in topology, the source of the QoS flow recomputes periodically a new route $\mathcal{R}'$. The new route $\mathcal{R}'$ replaces the old one $\mathcal{R}$ if and only if $\mathcal{R}$ is no longer valid (e.g.: due to link breaks), or $\mathcal{R}'$ is strictly shorter than $\mathcal{R}$.

BE (Best Effort) traffic is routed hop-by-hop, according to the routing table computed at each node, like in OLSR. However this routing table is computed based on QoS-MPRs and not on MPRs.

## 2.5 QoS admission control

The purpose of QoS admission control is twofold:

- Decide if a new QoS flow can be accepted into the network, based on its amount of required bandwidth $b$ and on the QoS topology.
- Protect QoS flows from BE traffic by limiting the BE traffic transmission.

When a source node detects a new coming QoS flow with an amount of required bandwidth, it performs the QoS admission control. If there is a route $\mathcal{R}$ that meets this bandwidth requirement as described in section 2.4, then the flow is accepted. Otherwise, it is rejected. We notice that the QoS admission control is performed by the source. No extra signaling is needed.

In order to limit BE traffic, we use a leaky bucket. Our goals of shaping BE traffic are:

- Bandwidth allocated to QoS flows is preserved.
- Nodes located in the same interference area dynamically share the available bandwidth left by all QoS flows.
- This share is computed at each node in a distributed manner. No centralized entity is required.

The following property formally defines the rate of the leaky bucket used at each node to regulate the BE traffic injected (i.e generated or forwarded) by this node:

Property 2. *For any node $N_i$,*

- *Let $\mathcal{N}_i \triangleq \{N_i\} \bigcup \mathcal{N}_i^1 \bigcup \mathcal{N}_i^2$ be the set of $N_i$, its 1-hop and 2-hop neighbors.*
- *Let $\gamma_i$ be the minimum value of the available bandwidth left by QoS flows in $\mathcal{N}_i$: $\gamma_i \triangleq \min_{N_j \in \mathcal{N}_i}\{r_j\}$.*
- *Let $\sigma_j$ be the amount of bandwidth consumed by BE traffic of a node $N_j \in \mathcal{N}_i$.*

*The maximum amount of bandwidth available to BE traffic at each node in $\mathcal{N}_i$, denoted $\kappa_i$, is defined by:*

$$\sum_{N_j \in \mathcal{N}_i} \min(\sigma_j, \kappa_i) = \gamma_i$$

*If node $N_i$ knows $\kappa_j$, the bandwidth available to BE traffic for each $N_j \in \mathcal{N}_i$, then it can set $\rho_i$, the rate of its leaky bucket:*

$$\rho_i = \min_{N_j \in \mathcal{N}_i}\{\kappa_j\}$$

Proof. It is trivial that the leaky bucket rate computation can be done by each node and does not need centralized entity. All necessary information on bandwidth can be broadcast to 1-hop and 2-hop neighbors using *Hello* messages. We now prove that the bandwidth of QoS flows is not affected by BE traffic bandwidth consumption.

Let $N_i$ be a node that carries QoS flows. Let $N_j \in \mathcal{N}_i$, by the leaky bucket definition: $\sigma_j \leqslant \rho_j$, which implies: $\sigma_j \leqslant \min_{N_k \in \mathcal{N}_j}\{\kappa_k\} \leqslant \kappa_i$. The sum of $\sigma_j$ over $\mathcal{N}_i$ yields:

$$\sum_{N_j \in \mathcal{N}_i} \sigma_j \leqslant \sum_{N_j \in \mathcal{N}_i} \min(\sigma_j, \kappa_i) \leqslant \gamma_i$$

Therefore, the sum of bandwidth consumption by all nodes in the interference area of node $N_i$ is smaller than its available bandwidth left by QoS flows. This bound is the best possible when we consider a network where all nodes up to two hops interfere with each other. ☐

We give hereby an algorithm to compute the leaky bucket rate at each node with complexity of $\mathcal{O}(n \log n)$ for an average number of $n$ neighbors per node:

```
(1) Sort nodes in $\mathcal{N}_i$ by increasing order of $\sigma_j$.
    Let $\{N_1, N_2, \ldots, N_n\} = \mathcal{N}_i$ be the sequence
    of nodes after sorting.
(2) $k := 1$. While $(k \leq n)$ and $(\sigma_k < \frac{\gamma_i}{n-k+1})$, do:
    (2.1) $\gamma_i := \gamma_i - \sigma_k$
    (2.2) $k := k + 1$
(3) If $(k > n)$, then $\kappa_i := \sigma_n$; else $\kappa_i := \frac{\gamma_i}{n-k+1}$
(4) $\rho_i := \min_{N_j \in \mathcal{N}_i}\{\kappa_j\}$
```

# 3. IMPLEMENTATION OF QOS SUPPORT

## 3.1 Technical obstacles

We identified the following technical obstacles when we implemented our mechanisms of QoS support on a MANET testbed equiped with 802.11b network interfaces:

- **How to obtain $r_i$, the amount of available bandwidth at each node ?** This value is not provided by the 802.11b MAC layer. We use the payload of traffic generated and relayed by all nodes in the 1-hop and 2-hop neighborhood to estimate $r_i$ with the following rule:

Rule 1. *Let us consider a node $N_i$. Let $\tau_j$ be the payload of traffic generated and relayed by a node $N_j \in \mathcal{N}_i$. Let $\mathcal{B}$ be the bandwidth capacity of the medium (e.g. 11Mbps). The amount of available bandwidth of $N_i$ is estimated as follows.*

$$r_i \simeq \mathcal{B} - \sum_{N_j \in \mathcal{N}_i} \tau_j$$

- **How to convert an end-to-end throughput required by a QoS flow to the amount of bandwidth needed at the 802.11b MAC layer?** The amount of bandwidth needed at the 802.11b layer must include all interferences created by the relays of this QoS flow. This value depends on how many times a node $N$ belongs to the interference area of this flow. We assume in section 1 that a node belongs at most five times to the interference area of a flow on a straight line. The following heuristic may apply: for a given amount of bandwidth at the 802.11b MAC layer, if the QoS routing algorithm finds a route $\mathcal{R}$ that meets this bandwidth requirement, then we check that no node in the network belongs more than five times to the interference area of $\mathcal{R}$. This heuristic is analyzed with more details in [12]. In most cases, this bound is shown to be sufficient.

Rule 2. *Let $\Phi$ be the end-to-end throughput required by the QoS flow. The equivalent amount of bandwidth needed at the 802.11b MAC layer is estimated by:*

$$b \simeq \alpha \times \Phi \times \min(5, k)$$

*where $\alpha$ is a constant that takes into account the overhead of IP and MAC layers, including MAC acknowledgement and possible MAC layer's retransmissions.*

*$k$ is the shortest route's length from the source to the destination when there is no bandwidth constraint. $k$ is obtained from the OLSR routing table.*

## 3.2 Modules implemented to support QoS

To implement our QoS support, we have developped the following modules:

- **Quality of service Source Routing (QSR)**: when a new QoS flow is detected at its source, QSR asks QoS routing in OLSR to find a shortest route $\mathcal{R}$ that meets the required bandwidth. If $\mathcal{R}$ is found, then QSR will capture all packets of this flow, add the route $\mathcal{R}$ to their IP header and set the *source routing* option before reinjecting them into the network. They are handled by the IP layer at the intermediate nodes and follow exactly the route $\mathcal{R}$.

• **Object-oriented OLSR (OOLSR)**: is in charge of detecting neighbors and disseminating topology in order to build routing table.
− Each node computes its payload $\tau_i$ for each traffic class (control, QoS and BE) from the counters of the Linux kernel. It then deduces the values of $r_i$ according to the rule 1, and of $\gamma_i$, the minimum of $\{r_j\}$ over all 1-hop and 2-hop neighbors.
− The values of $\tau_i$ for each traffic class and $r_i$ are appended to the *Hello* messages broadcast by node $N_i$. This node also relays these values of its 1-hop neighbors in its *Hellos*. The *TC* messages generated by $N_i$ contain the values of $\gamma_i$ and $r_j$ for each $N_j$ a QoS-MPR selector of node $N_i$.
− Upon receipt of a *Hello* or *TC* message, the OOLSR module updates its QoS neighborhood or topology table. The QoS-related information is stored in these tables. If the status of a link is changed, the routing table is recomputed.
− This module also computes the leaky bucket rate $\rho_i$. This leaky bucket is used to shape BE traffic.

• **Linux's Traffic Control**: manages the packet queues. There are three queues in our implementation: control, QoS and BE. Packets in these queues are managed according to their priority: control packets first, then QoS packets and then BE packets. The BE queue is shaped by a leaky bucket: we use Linux *Token Bucket Filter* (TBF) whose rate is $\rho_i$.

## 3.3 The platform and its parameters

The CELAR MANET/OLSR platform, illustrated by Figure 2 has 18 nodes. These nodes include IP routers and laptops. Each node is equiped with an 802.11b network card and runs the OLSR protocol with QoS support. Ten nodes are placed in the central tower of the CELAR building, and two in a shelter outside, denoted as *ALGECO* on Figure 2. The other nodes are mobile and embedded in vehicles outside the building. This MANET is connected to a wired network via a router acting as OLSR/OSPF gateway.

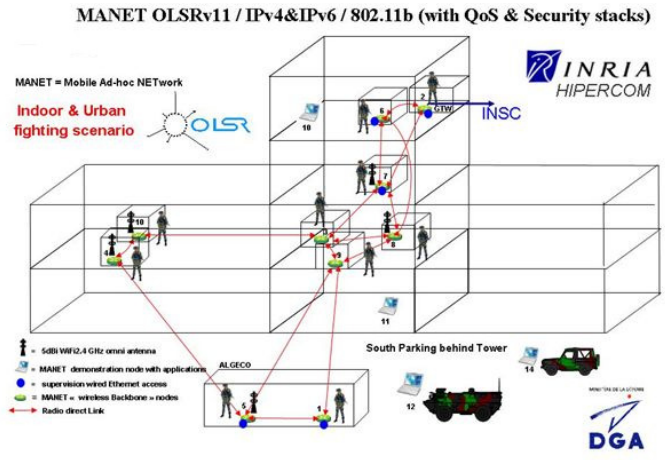

**Figure 2: The CELAR MANET/OLSR platform.**

During the setup of this platform, the signal thresholds used in the link hysteresis are set to -39dB (low) and -30dB (high). These values have been tuned in wireless cards. They allow us to keep only stable wireless links. We made some measurements to determine the maximum IP throughput on a wireless link. It turned out to be 5Mbps. The parameter $\alpha$ in rule 2 is set accordingly.
We assume that the throughput required by a QoS flow depends on the user application. This is defined by the destination port number in the IP header, e.g.: an audio flow from Robust Audio Tool (RAT) has a destination port number of 5010 and a required end-to-end throughput of 128kbps.

## 4. PERFORMANCE EVALUATION

The OLSR parameters are set as recommended in [1]: *Hello* interval is 2s and *TC* interval is 5s. In all the following scenarios experimented on the MANET testbed, all flows are CBR with a packet size set to 1000 bytes. Each flow lasts 5 minutes. The measures obtained from the testbed are compared with NS-2 simulation results. With simulations, the reception range of a node is 250m and the carrier sense range is 550m. We reproduced in simulations the same network topology as in our testbed. The RTS/CTS option is disabled both in simulations and tests. Simulation results are averaged on ten runs.

## 4.1 A better QoS

In order to show benefits provided by QoS support, we measure the quality of user's flows, without QoS support in the first place, then with QoS support, for the same network configuration and scenario. The quality of service is measured in terms of throughput and delivery rate obtained at the destination. The network topology is illustrated by Figure 3, where the number near each router is this depicted in Figure 2 illustrating the real platform.

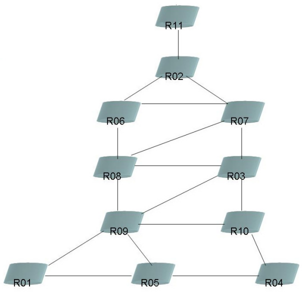

**Figure 3: Links between the routers**

In this scenario, we have two flows $f_1$ and $f_2$, each requires a throughput of 200kbps and is two-hop long. We then introduce two more flows $f_3$ and $f_4$ at 200kbps and three-hop long. Finally, we introduce two more flows $f_5$ and $f_6$ at 300kbps and five-hop long. We focus on flows $f_1$ to $f_4$ and distinguish two cases:
• Without QoS support, the quality of all four flows is very low as shown in Table 1. Their packets have a very high loss rate (up to 67%). The simulation results are slightly better, because the simulator does not take into account the environmental factors such as physical obstacles. . .
• With QoS support, the flows $f_1$ to $f_4$ receive a quality of service close to this requested. The throughput measured at the flow destination is close to this requested and the loss rate is small, as shown by Table 2. Simulation results are close to the measures obtained on the real platform.

| flow | src | dest | hop | src rate | Measured | | Simulated | |
|---|---|---|---|---|---|---|---|---|
| | | | | | thr. | loss | thr. | loss |
| $f_1$ | R11 | R06 | 2 | 200 | 157 | 20% | 164 | 17.8% |
| $f_2$ | R06 | R11 | 2 | 200 | 130 | 34% | 166 | 17.2% |
| $f_3$ | R05 | R07 | 3 | 200 | 149 | 25% | 153 | 23.3% |
| $f_4$ | R07 | R05 | 3 | 200 | 111 | 48% | 145 | 27.6% |
| $f_5$ | R01 | R11 | 5 | 300 | 92 | 72% | 101 | 66.5% |
| $f_6$ | R11 | R01 | 5 | 300 | 88 | 77% | 99 | 67% |

**Table 1: Throughput and loss rate without QoS support.**

| flow | src | dest | hop | src rate | Measured | | Simulated | |
|---|---|---|---|---|---|---|---|---|
| | | | | | thr. | loss | thr. | loss |
| $f_1$ | R11 | R06 | 2 | 200 | 193 | 3% | 192 | 3.8% |
| $f_2$ | R06 | R11 | 2 | 200 | 191 | 4% | 192 | 3.8% |
| $f_3$ | R05 | R07 | 3 | 200 | 197 | 1% | 191 | 4.2% |
| $f_4$ | R07 | R05 | 3 | 200 | 200 | 0% | 188 | 5.9% |
| $f_5$ | R01 | R11 | 5 | 300 | 39 | 87% | 77.7 | 74.1% |
| $f_6$ | R11 | R01 | 5 | 300 | 40 | 86% | 88.5 | 70.5% |

**Table 2: Throughput and loss rate with QoS support.**

We can conclude that both simulation results and testbed measures confirm the benefits provided by QoS support. The required throughputs of QoS flows are met. Also, the loss rate is satisfactory.

## 4.2 A higher number of QoS flows accepted

In this scenario, we have four flows, each requires a throughput of 150kbps. They have the same source R11, and the same destination R05 five hops away from R11.

- Without QoS support, the results are reported in Table 3. Testbed measures show a rather high loss probability (18%). Simulation results are slightly better for the same reasons as in the previous scenario.

| flow | src | dest | hop | src rate | Measured | | Simulated | |
|---|---|---|---|---|---|---|---|---|
| | | | | | thr. | loss | thr. | loss |
| $f_1$ | R11 | R05 | 5 | 150 | 123 | 17% | 130 | 13.2% |
| $f_2$ | R11 | R05 | 5 | 150 | 123 | 18% | 129 | 13.7% |
| $f_3$ | R11 | R05 | 5 | 150 | 123 | 18% | 128 | 14.5% |
| $f_4$ | R11 | R05 | 5 | 150 | 124 | 18% | 127.5 | 15% |

**Table 3: Throughput and loss rate without QoS support, 4 flows.**

If only three flows are present and no QoS support is provided, testbed measures show in Table 4 that their quality is acceptable (3% of lost packets). Hence, without QoS support, the network only accepts 3 flows out of 4. Each flow is 5-hop long and has a 150kbps bit rate.

| flow | src | dest | hop | src rate | thr. | loss |
|---|---|---|---|---|---|---|
| $f_1$ | R11 | R05 | 5 | 150 | 145 | 3% |
| $f_2$ | R11 | R05 | 5 | 150 | 145 | 3% |
| $f_3$ | R11 | R05 | 5 | 150 | 145 | 3% |

**Table 4: Throughput and loss rate without QoS support, 3 flows.**

- With QoS support, the quality of all 4 flows is improved as shown in Table 5. Packet loss does not exceed 12%. QoS support allows to accept all 4 flows instead of 3. This improvement of 33% is provided by the IP *source routing* option. Indeed there is no alternative route in this configuration, as illustrated by Figure 3. This option avoids frequent route oscillations encountered by hop-by-hop routed packets. It provides a better routing stability. Simulation results confirm our measures on testbed.

| flow | src | dest | hop | src rate | Measured | | Simulated | |
|---|---|---|---|---|---|---|---|---|
| | | | | | thr. | loss | thr. | loss |
| $f_1$ | R11 | R05 | 5 | 150 | 133 | 11% | 134 | 10.7% |
| $f_2$ | R11 | R05 | 5 | 150 | 133 | 11% | 134 | 10.7% |
| $f_3$ | R11 | R05 | 5 | 150 | 131 | 12% | 132 | 11.9% |
| $f_4$ | R11 | R05 | 5 | 150 | 134 | 10% | 130 | 13.2% |

**Table 5: Throughput and loss rate with QoS support, with all 4 flows.**

## 4.3 Avoidance of saturated regions

We add two more laptops into our scenario, denoted as V11 and V12 (see Figure 4). They extend the network topology. It becomes then possible to find a route that does not cross a region saturated by interferences, by choosing routes that have the largest available bandwidth. We recall that interferences are supposed to affect all nodes up to two hops away from the transmitter.

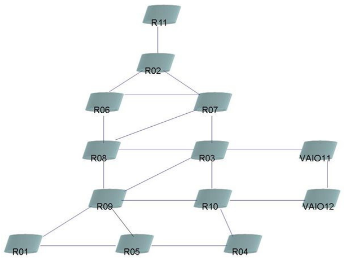

**Figure 4: The configuration with 2 additional VAIOs**

.

In this scenario, a flow $f_1$ is sent from R11 to R08 via R02 and R06 with a throughput of 480kbps, another flow $f_2$ from R11 to R07 via R02 at 480kbps. Consequently, the nodes R11, R02, R06, R07, R08, R09 and R03 are saturated. Then a flow $f_3$ is sent from V11 to R01 at 480kbps.

- Without QoS support, flow $f_3$ takes the shortest route (V11, R03, R09, R01). Hence, $f_3$ generates interferences on the destinations of $f_1$ and $f_2$. This explains the low throughput and significant packet losses of $f_1$ and $f_2$ as shown in Table 6.

| flow | src | dest | hop | src rate | thr. | loss |
|---|---|---|---|---|---|---|
| $f_1$ | R11 | R08 | 3 | 480 | 316 | 34.2% |
| $f_2$ | R11 | R07 | 3 | 480 | 321 | 33.2% |
| $f_3$ | V11 | R01 | 3 | 480 | 462 | 3.7% |

**Table 6: Results obtained without QoS support.**

- With QoS support, all flows are sent as QoS flows. Flow $f_3$ takes a longer route to avoid the saturated region: V11, V12, R10, R04, R05, R01. As a consequence, it does not generate interferences on $f_1$ and $f_2$. The quality of all three flows are preserved as shown in Table 7. These results have been obtained only by simulations.

| flow | src | dest | hop | rate | thr. | loss |
|---|---|---|---|---|---|---|
| $f_1$ | R11 | R08 | 3 | 480 | 453 | 5.7% |
| $f_2$ | R11 | R07 | 3 | 480 | 450 | 6.2% |
| $f_3$ | V11 | R01 | 5 | 480 | 469 | 2.2% |

**Table 7: Results obtained with QoS support.**

This remarkable improvement of flow's quality can be explained by the fact that this QoS support takes radio interferences into consideration. While QoS signaling gathers information on the saturated regions in the network, QoS routing allows flows to avoid them.

# 5. OVERHEAD ANALYSIS

In this section, we analyze the average number of QoS-MPRs selected per node. We then compare the number of retransmissions of QoS-MPR and MPR flooding techniques and deduce the overhead generated by this QoS support.

## 5.1 Analysis of QoS-MPRs selection

We can compute the average number of QoS-MPRs selected per node.

PROPERTY 3. *Let $n$ be the average number of 1-hop neighbors per node. Let $m$ be the average number of QoS-MPRs selected per node. We have:*

*(i) $m = \mathcal{O}(\log n)$, on a linear network domain.*

*(ii) $m = \mathcal{O}(n^{\frac{1}{3}} \log n)$, on a planar domain.*

PROOF. We prove this property based on the following remark. The selection of QoS-MPRs is equivalent to the arrival process of pre-sorted 1-hop neighbors. Let $\{N_1, \ldots, N_n\}$ be the set of node $N$'s 1-hop neighbors, sorted by decreasing value of bandwidth available.

(i) Let us consider a linear domain. In the $i^{\text{th}}$ step of the arrival proccess, node $N_i$ is selected as QoS-MPR iff its distance to node $N$ is greater than the distance of any precedent node to $N$, so that $N_i$ can cover new 2-hop nodes. Therefore, the probability that a new 1-hop neighbor is selected as QoS-MPR is $\frac{1}{2^m}$, if $m$ nodes are already selected as QoS-MPRs. Let d$n$ be an infinitesimal quantity of nodes that arrive into node $N$'s 1-hop neighborhood. Let d$m$ be the number of node selected as QoS-MPRs. We have: $\mathrm{d}m = \frac{1}{2^m}\mathrm{d}n$. This implies $m = \mathcal{O}(\log n)$.

(ii) On a planar domain, the QoS-MPR set must include a minimal subset of nodes, denoted as $\mathcal{S}_1$ that cover all 2-hop neighbors. When $n$ is large enough, we may assume that they are located on the border of node $N$'s coverage. It is known from [11] that the size of $\mathcal{S}_1$ is $\mathcal{O}(n^{\frac{1}{3}})$.

We consider the subset of QoS-MPRs, denoted as $\mathcal{S}_2$ that are not in $\mathcal{S}_1$. They are selected according to their available bandwidth and to their 2-hop neighborhood coverage. Let $N_i, N_{i+1} \in \mathcal{S}_1$ that are located immediately next to each other on the border of $N$'s coverage. We are interested in the number of QoS-MPRs of $\mathcal{S}_2$ in the sector $(N_i, N_{i+1})$ of the disc centered on node $N$. The angle $(NN_i, NN_{i+1}) = \mathcal{O}(n^{-\frac{1}{3}})$ becomes small very quickly when $n$ increases. Thus, we may apply property $(i)$ to this sector when $n$ is large enough. This implies $|\mathcal{S}_2| = \mathcal{O}(n^{\frac{1}{3}} \log n)$. ☐

## 5.2 MPR and QoS-MPR floodings

We now evaluate the average number of retransmissions of a flooded message, using QoS-MPR flooding technique. Note that this technique is the same as MPR flooding in which MPRs are replaced by QoS-MPRs.

PROPERTY 4. *In a planar network domain with $\Gamma$ nodes and density $n$, the number of retransmissions of a flooded message using QoS-MPR flooding technique is $\mathcal{O}(\Gamma n^{-\frac{2}{3}} \log n)$.*

PROOF. The proof for this property is very similar to the one in [11], with the number of MPRs replaced by the number of QoS-MPRs. ☐

It is known from [11] that the number of retransmissions using MPR flooding is $\mathcal{O}(\Gamma n^{-\frac{2}{3}})$. Thus, using QoS-MPR flooding technique would introduce a factor of $\log n$ extra retransmissions. Therefore, using MPR for flooding should be prefered to using QoS-MPR. This conclusion implies that the network will need both MPRs and QoS-MPRs in order to support QoS and to provide optimized flooding.

Our NS-2 simulation result in Figure 5 compares the number of retransmissions of a *TC* message between QoS-MPR and MPR floodings. The network has 1000 nodes with densities varying from 10 to 200 neighbors per node. QoS-MPR flooding generates more retransmissions than MPR flooding.

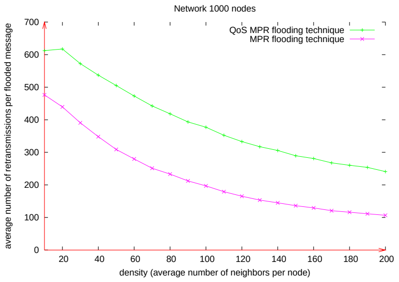

**Figure 5: Number of retransmissions by QoS-MPR and MPR floodings.**

## 5.3 Overhead computation

We are now interested in the topology dissemination locally perceived by each node of the network, i.e. the number of TC messages that each MPR node must retransmit according to the MPR flooding technique. Note that with QoS support, *TC* messages are generated by QoS-MPRs and forwarded by MPRs.

PROPERTY 5. *In a planar network domain with $\Gamma$ nodes and density $n$, the number of retransmissions of TC messages per MPR node is $\mathcal{O}\left(\Gamma n^{-\frac{2}{3}}(1 + \exp(-\lambda n^{\frac{1}{3}}))\right)$, where $\lambda$ is a constant.*

PROOF. From [11], each *TC* message is forwarded $\mathcal{O}(\Gamma n^{-\frac{2}{3}})$ times by MPR nodes. Let $\Gamma_s$ and $\Gamma_s^*$ denote the total number of MPRs and QoS-MPRs, respectively. During a *TC* interval, each QoS-MPR generates one *TC* message. Thus, there are $\mathcal{O}(\Gamma_s \Gamma n^{-\frac{2}{3}})$ *TC* messages, including copies, in the whole network. Therefore, each MPR retransmits $\mathcal{O}(\frac{\Gamma_s}{\Gamma_s^*}\Gamma n^{-\frac{2}{3}})$ *TC* messages, which concludes the proof. ☐

Since our QoS support does not generate any extra message, we conclude that the amount of overhead created by our QoS support is larger than OLSR's overhead by a factor of $(1 + \exp(-\lambda n^{\frac{1}{3}})) \log n$, considering that the size of each *TC* message is larger by a factor of $\log n$.

## 6. SCALABILITY OF THE QOS SUPPORT

In section 4, our simulation results are shown to be a good approximation to the measures made on the real platform, we now use simulations to study the behavior of the QoS support in large and dense MANETs.

Our network has 300 nodes with the average density of 14 neighbors per node (see Figure 6). First, we saturate the center of the network with eight flows: $f_1$ to $f_8$. Then, four additional flows ($f_9$ to $f_{12}$) are introduced to the network, they connect nodes from the upper border to the lower border, as shown in Figure 6. All flows have a throughput of 100kbps.

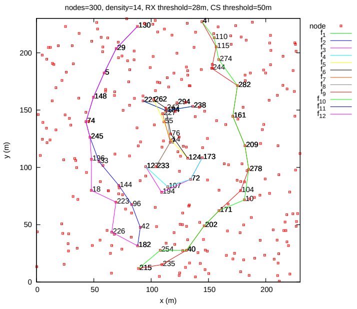

**Figure 6: Routes taken by the 12 flows in the network of 300 nodes, with QoS support.**

- Without QoS support, the flows $f_9$ to $f_{12}$ take the shortest routes. They cross the center of the network which is saturated. As a consequence, their packets encounter high loss probability. Their throughput also suffers from high fluctuation.

- With QoS support, we can see in Figure 6 that the flows $f_9$ to $f_{12}$ avoid the saturated area by taking longer routes, going around the center of the network. As a result, the quality of all flows are preserved, their throughput is maintained close to the requested throughput.

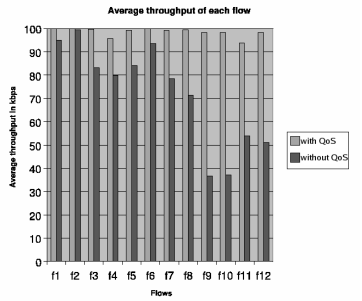

**Figure 7: Average end-to-end throughput per flow.**

- Figure 7 compares the average end-to-end throughput of each flow in both scenarios: with and without QoS support. We can see that our QoS support provides a guarantee of throughput to the QoS flows and also make better use of the bandwidth resources in the whole network.

## 7. CONCLUSION

MANETs show many interesting properties with regard to military applications: they are self-organizing, support mobility and can be quickly deployed. However military applications also have QoS requirements: throughput guarantee, prioritized access ... In this paper, we report the performance of our QoS support designed for MANETs and implemented on a testbed based on 802.11b network equipments and the OLSR protocol. Our QoS support includes four main components: QoS signaling, QoS routing, QoS admission control and QoS models. QoS routing and QoS admission control take into consideration interferences in order to provide a better QoS. Our measures obtained from the testbed are compared with NS-2 simulations. Both results allow to conclude that this QoS support significantly improves the quality of user flows, in terms of throughput and delivery rate. We also give an analytical evaluation on the overhead of our QoS support. We show that this overhead is kept very low. In particular, our QoS support maintains the same optimization degree as MPR flooding. We have also made experiments with mobility [12] that show that this solution improves the QoS perceived by the user as long as mobility is $\leq$ 15m/s.

## 8. REFERENCES

[1] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot: *Optimized Link State Routing Protocol*, RFC 3626, IETF, 2003.

[2] G.S. Ahn, A. Campbell, A. Veres, L.H. Sun: *SWAN: Service Differentiation in stateless Wireless Ad-Hoc Networks*, INFOCOM'2002, New York, June 2002.

[3] G. Allard, L. Georgiadis, P. Jacquet, B. Mans: *Bandwidth Reservation in Multihop Wireless Networks: complexity, heuristics and mechanisms*, International Journal of Wireless and Mobile Computing (inderscience), accepted for publication in May 2004, To appear (ISSN-1741-1084).

[4] C. Chaudet, I. Guérin-Lassous: *BRuIT: Bandwidth Reservation under InTerferences influence*, in European Wireless (EW), pp. 466-472, 2002.

[5] K. Nahrstedt, S. Shah, K. Chen: *Cross-layering architectures for bandwidth management in wireless networks*, Resource management in wireless networking, Edited by M. Cardei, I. Cardei, D. Du, 2004.

[6] D.Q. Nguyen and P. Minet: *QoS support and OLSR routing in a mobile ad hoc network*, in 5th IEEE International Conference on Networking, ICN06, Mauritius, April 2006.

[7] Y. Ge, T. Kunz, L. Lamont: *Quality of Service Routing in Ad-Hoc Networks Using OLSR*, HICSS'03, Big Island, Hawai, January 2003.

[8] H. Badis and K. Al Agha: *QOLSR, QoS routing for Ad Hoc Wireless Networks Using OLSR*, in European Transactions on Telecommunications, vol. 15, n4, 2005.

[9] T. Clausen: *Comparative Study of Routing Protocols for Mobile Ad-hoc NETworks*, INRIA research report, N5135, March 2004, INRIA Rocquencourt, France.

[10] E. W. Dijkstra: *A note on two problems in connexion with graphs*, in Numerische Mathematik, 1959.

[11] P. Jacquet, A. Laouiti, P. Minet, L. Viennot: *Performance analysis of OLSR Multipoint Relay flooding in two ad-hoc wireless network models*, INRIA research report, N4260, September 2001, INRIA Rocquencourt, France.

[12] D.Q. Nguyen: *Support de Qualité de Service dans les Réseaux Ad Hoc*, Ph.D. thesis, September 2006, University of Paris 6, France.