A Reservation Scheme Satisfying Bandwidth QoS Constraints for Multirate Ad-hoc Networks

Llorenç Cerdà[1], Michael Voorhaen[2], Rafael Guimarães[1], José-M Barceló[1],
Jorge García[1], Julián Morillo[1] and Chris Blondia[2]

[1]Technical University of Catalonia
Computer Architecture Dept.
Jordi Girona 1-3,
E-08034 Barcelona, Spain

[2]University of Antwerp
Dept. Mathematics and Computer Science
Middelheimlaan 1,
B-2020 Antwerpen, Belgium

***Abstract*—Achieving QoS (Quality of Service) in Mobile Ad-hoc NETworks (MANET) has been a research topic in the last years. In this paper we describe a QoS reservation mechanism for *Multirate Ad-hoc Networks*. By multirate we refer to those networks where the mobile nodes can dynamically switch between several link rates. This allows the mobile nodes to select the transmission rate with better performance for every neighbor. The mechanism is targeted for sources requiring a bandwidth allocation.**

I. INTRODUCTION

MANETs (Mobile Ad-hoc NETworks) have characteristics such as flexibility, fast and easy deployment, robustness which make them an interesting technology for military, public safety, emergency and disaster applications. Providing QoS (Quality of Service) in a MANET is, however, a difficult task because: (i) the capacity of the physical links is variable depending on factors such as the distance, signal to noise ratio, interference, etc, (ii) the transmission media is shared between different nodes that have to be synchronized, (iii) MANET nodes are generally mobile and the network topology may change, and (iv) high signaling overhead due to the recovery of already hard-QoS reservations may be a problem due to the scarce transmission resources.

There are several QoS frameworks for MANETs proposed in the literature addressing some of the aspects for QoS support. Authors of [1] present a framework called *Flexible Quality of Service Model* (FQMM) that combines a reservation procedure for high priority traffic with a service differentiation for low-priority traffic. However, this hybrid provisioning scheme does not take into account the characteristics of MANETs and all the drawbacks of the IntServ and DiffServ remain. Other proposals are less general and address some of the aspects to be taken into account in a QoS framework for MANETs. *In-band signaling support for QOS in mobile ad hoc networks* (INSIGNIA) [2], is an in-band signaling protocol designed explicitly for MANETs which must be integrated with an ad-hoc routing protocol. *COre-Extraction Distributed Ad Hoc Routing Algorithm* (CEDAR) [3], is a protocol proposed to reduce the control overhead by defining a backbone and *Hierarchically-organized Multihop Wireless Networks* (HMWN) [4], is defined for cluster networks. A reservation scheme with AODV can be found in [5]. Another QoS approach based on measurements is presented in [6].

In this paper we treat the problem of achieving a reservation taking into account the available bandwidth in a coverage area and the traffic generated and forwarded by the neighbors and interferent MNs in that coverage area. Furthermore, we apply our reservation to *Optimized Link State Routing Protocol* (OLSR) [7], although the reservation scheme can be applied to other ad-hoc routing protocols. The results show the feasibility of our scheme for guaranteeing the QoS requirements.

II. BANDWIDTH QOS CONSTRAINT

Without loss of generality, through this paper we shall assume two traffic classes: with QoS and best effort. Furthermore, we shall assume that the MAC is able to isolate traffic classes such that QoS connections have priority over best effort, e.g. by using 802.11e.

We shall refer to the normalized QoS traffic generated or in transit at MN_i as the *bandwidth reservation* (x_i) at a MN_i. I.e. if r_{ij} is the amount of QoS traffic in bits per second (bps), sent from MN_i to MN_j at a link rate v_{ij} bps, then:

$$x_i = \sum_{j \in \mathcal{N}_i} r_{ij}/v_{ij} \tag{1}$$

Where $\mathcal{N}_i$ is the set of neighbor MNs of MN_i, i.e. $\{MN_j | j \in \mathcal{N}_i\}$ is the set of nodes in range with MN_i. We assume that all transmissions use the same channel, i.e. one node cannot simultaneously received frames from two neighbors. Therefore, x_i can be interpreted as the channel occupancy by the QoS traffic sent by MN_i.

Due to the interference among MNs, two MNs may not be allowed to simultaneously transmit. Therefore, we define the *QoS Load Demand* at MN_i as the channel occupancy to send all

---

This work was supported by the Ministry of Edu. of Spain under grant CICYT TIC-2001-0956-C04-01, by the Generalitat de Catalunya under grant CIRIT 2001-SGR-00226, by the Fund for Scientific Research Flanders under Scientific Research Community Broadband communication and multimedia services for mobile users, by DWTC Belgium under project IAP P5/11 MOTION (Mobile Multimedia Communication Systems and Networks), by IWT under project 020152–End-to-end QoS in IP based Mobile Networks, by the European NoE EuroNGI and by the European project WIDENS.

QoS traffic of MN_i and the nodes intefering with MN_i, observed from MN_i. Our goal is a *bandwidth reservation scheme* (e.g. peak rate allocation) subject to the following *QoS constraint*:

$$LD_i \leq Q, \forall i. \tag{2}$$

In other words, we want that the channel occupancy to send QoS traffic observed from any MN is $\leq Q$. Through this paper we shall assume a MAC where it makes sense defining LD_i as:

$$LD_i = \sum_{j \in \mathcal{N}_i^+} x_j \tag{3}$$

Where $\mathcal{N}_i^+$ is the set of MN_i and its neighbors. Such *Load Demand* corresponds to a MAC where two neighbor MNs could not simultaneously transmit a packet. Note that equation (3) would not be accurate for a MAC as 802.11 using RTS/CTS. This is because all nodes receiving not only RTS but CTS are silent. Therefore, the load demand for an RTS/CTS MAC should be defined not only by the traffic transmitted by the neighbors (as we assume in this paper), but also by the traffic received by them.

Of course, due to collisions and other MAC mechanisms, QoS traffic transmitted by the MNs may consume more bandwidth than the QoS Load Demand given by (3). We shall assume that the parameter Q is dimensioned to cope with this, such that delays are acceptable for QoS connections.

It is convenient to define the *Maximum Available Bandwidth* at MN_i (MAB_i) as:

$$MAB_i = Q - LD_i \tag{4}$$

Thus, the *QoS constraint* becomes:

$$\text{QoS constraint: } MAB_i \geq 0, \forall i. \tag{5}$$

III. Reservation Approach

We define the *available bandwidth* AB_i to allocate new reservations at MN_i as:

$$AB_i = \min\{MAB_j\}, j \in \mathcal{N}_i^+ \tag{6}$$

We shall use the notation $MN_i \rightarrow MN_j$ to denote two consecutive MNs belonging to the path to be reserved for a new QoS connection. Suppose that a new QoS connection of r bps has to be established. We claim that if the path to be reserved does not follow unnecessary jumps (i.e. for all nodes MN_l belonging to the reserved path, it holds that if $\cdots MN_i \rightarrow MN_j \rightarrow MN_k \cdots$, then $MN_i, MN_k \in \mathcal{N}_j$, $MN_l \notin \mathcal{N}_j, \forall l \neq i, j, k$), then, the QoS constraint given by (5) is satisfied if the following CAC conditions hold:

- For the MN_i originating the new QoS connection:
  - If the destination (MN_j) is a neighbor of MN_i, then $AB_i \geq r/v_{ij}$.
  - If the destination is not a neighbor of MN_i and the connection follows the path $MN_i \rightarrow MN_j \rightarrow MN_k$, then $AB_i \geq r/v_{ij} + r/v_{jk}$.
- For all the transit MN_j (located along the path between the source and the destination):
  - If the destination (MN_k) is a neighbor of MN_j and the connection follows the path $MN_i \rightarrow MN_j \rightarrow MN_k$, then $AB_j \geq r/v_{ij} + r/v_{jk}$.
  - If the destination is not a neighbor of MN_j and the connection follows the path $MN_i \rightarrow MN_j \rightarrow MN_k \rightarrow MN_l$, then $AB_j \geq r/v_{ij} + r/v_{jk} + r/v_{kl}$.

Proof: Assume that the new QoS connection is accepted and use (4) and (6) to verify that the QoS constraint is satisfied.

For instance, suppose the case when the new QoS connection of r bps is generated at MN_i and the destination MN_k is not one of its neighbors (see Fig. 1). Assume that the CAC accepts this connection along the path $MN_i \rightarrow MN_j \rightarrow MN_k$. Then, x_i and x_j will be respectively increased by r/v_{ij} and r/v_{jk}. Thus, the MAB of MN_i, MN_j and all their common neighbors will be decreased by $r/v_{ij} + r/v_{jk}$. Since $AB_i \geq r/v_{ij} + r/v_{jk}$ and $AB_j \geq r/v_{ij} + r/v_{jk}$ holds from the previous CAC conditions, equation (6) implies $MAB_l \geq 0, \forall l$.■

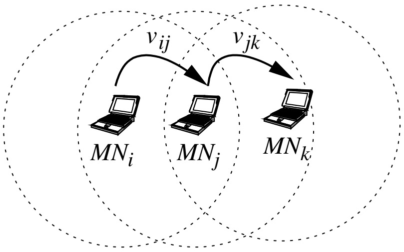

Fig. 1. Example of a connection generated at MN_i with destination MN_k.

IV. Implementation

The reservation scheme described in the previous sections requires that each MN_i knows two quantities from their neighbors $\mathcal{N}_i$: their reservation (x_j, $j \in \mathcal{N}_i$) and their maximum available bandwidth (MAB_i, $j \in \mathcal{N}_i$). This could be implemented by means of each MN_i broadcasting *HELLO packets* with x_i and MAB_i.

In the following we describe how to integrate our reservation scheme in the OLSR [7] routing protocol.

(i) OLSR HELLO messages are modified such that each MN_i advertises x_i and MAB_i to their neighbors.
(ii) Each MN_i collects the QoS messages from their neighbors to compute AB_i according to (6).
(iii) OLSR TC messages were modified to advertise AB_i and v_{ij} next to advertising the MPR selectors. This way each node has knowledge of the network topology and the bandwidth available in the network.
(iv) In order to find a route that meets the QoS requirements, we modified the OLSR route selection algorithm to find a shortest hop path that has enough bandwidth to meet these requirements. Since the TC messages also advertise AB_i and v_{ij} the originating node has enough information to decide if enough resources are available. See section IV-A.
(v) The reservation of the bandwidth at the intermediate nodes is done by adding the requested rate to the IP header (e.g. by using an IP option). This way an intermediate node which has not yet seen the flow will be able to allocate the bandwidth. The requested rate can be advertised for

```
The following pseudo algorithm describes how the CAC is integrated into the OLSR route selection
algorithm:

(1) Add all one hop neighbors registered as symmetric to the routing table with a hop-count of 1 and for
    which the CAC allows this route.
(2) For each symmetric one-hop neighbor, add all two hop neighbors registered on that neighbor that
    has:
        - not already been added to the routing table.
        - a symmetric link to the neighbor.
        - been allowed by the CAC module.
    These Entries are added with a hop-count of two and next-hop as the current neighbor.  Set n equal to
    two.
(3) Then, for every added node N in the routing table with hop-count n add all entries from the TC set
    where:
        - the originator in the TC entry is N
        - the destination has not already been added to the routing table
        - the CAC determined that enough resources are available among the route
    New entries are added with a hop count of n+1 and next-hop as the next-hop registered on N's routing
    entry.
(4) Increase n with one and do step 3 over until there are no entries in the routing table with hop-count
    equal to n or if a route to the destination was found
```

Fig. 2. Integration of the CAC Algorithm in OLSR.

a certain amount of time, number of packets or until an ACK is received to say that the flow has been set up. This is done to make this approach robust to packet loss.

For the remainder of the paper we will call our OLSR implementation extended with the QoS signaling QOLSR. We wish to stress that our protocol has little in common with [8], only the idea of extending OLSR with QoS support. The signaling introduced in QOLSR is similar to the INSIGNIA protocol [2], although the full feature set of INSIGNIA is not implemented since this was not necessary for the goal of this paper.

A. CAC Integration in OLSR

As explained in the previous section each node will gather the necessary information for performing the CAC from the TC packets. We modified the default route selection algorithm from OLSR to be able to compute a route for QoS flows that meets the bandwidth requirements. The CAC is performed during this route computation to remove intermediate routes that do not have enough resources available. Figure 2 shows the route computation algorithm in pseudo code. We will now apply this algorithm to a small example that will show how the CAC is implemented. Suppose N_i is the source and node N_l is the destination. The network topology is as follows:

$$N_i \rightarrow N_j \rightarrow N_k \rightarrow N_l$$

In step (1) the check $AB_i \geq r/v_{ij}$ is performed, regardless if N_j is the destination or not. If a neighbor is not the destination then the check $AB_i \geq r/v_{ij} + r/v_{jk}$ should be performed but since we do not know the destination of the 2nd hop this is not yet possible. If the first check succeeds the route is added. RAB_i (the remaining available bandwidth for MN_i) is set to $AB_i - r/v_{ij}$. RAB_j is set to $AB_j - r/v_{ij}$

In step (2) we will only add the route to N_k if the checks $RAB_i \geq r/v_{jk}$ and $RAB_j \geq r/v_{jk}$ succeed. If the CAC succeeds and N_k is the destination then a route has been found. If N_k is not the destination we set RAB_j to $RAB_j - r/v_{jk}$ and we set RAB_k to $AB_k - r/v_{jk}$. The partial route is then added to the table along with RAB_j and RAB_k.

In step (3) $RAB_j \geq r/v_{kl}$ and $RAB_k \geq r/v_{kl}$ are checked to see if the flow can be allowed into the network. If N_l is not the destination but the CAC succeeds then RAB_k is set to $RAB_k - r/v_{kl}$ and RAB_l to $AB_l - r/v_{kl}$. Step (3) can then be repeated until a destination is found by letting N_k become N_j and N_l become N_k. In case N_l is the destination a route has been found and the algorithm will stop.

V. SIMULATION RESULTS

We have added our reservation scheme in the OLSR implementation available in [9] for the network simulator [10]. We have simulated the following scenario:

- MAC: 802.11 with RTS/CTS disabled and multi-rate on.
- Multi-rate parameters: for a distance less than 100 meters the rate is 11 Mbps; for a distance between 100 and 300 meters the rate is 2 Mbps.
- CBR connections sending packets of size 500 bytes and rate 32 kbps.
- The QoS constraint for CBR connections is Q = 12.5%.
- 40 MNs randomly placed over a square 1000x1000 meters.
- MN coverage of 300 meters (i.e. any two MN at a distance $\leq$ 300 meter are in range).
- Each pair of nodes initiates a unidirectional call staggered 15s. Thus, 20 calls are initiated (20 x 32 = 640 kbps).
- The simulation time is 700 s, including a 100 s startup time giving OLSR the time to exchange routing information before starting the applications.
- The nodes don't move.

Using the following parameters for the OLSR protocol:

- HELLO_INTERVAL: 0.5 seconds
- TC_INTERVAL: 2 seconds
- NEIGHB_HOLD_TIME: 5 x HELLO_INTERVAL
- TOP_HOLD_TIME: 3 x TC_INTERVAL
- DUP_HOLD_TIME: 25 seconds

In the following we explain the results obtained using OLSR/QOLSR.

Figure 3 shows the evolution of the connections established with each protocol. Note that all connections are established

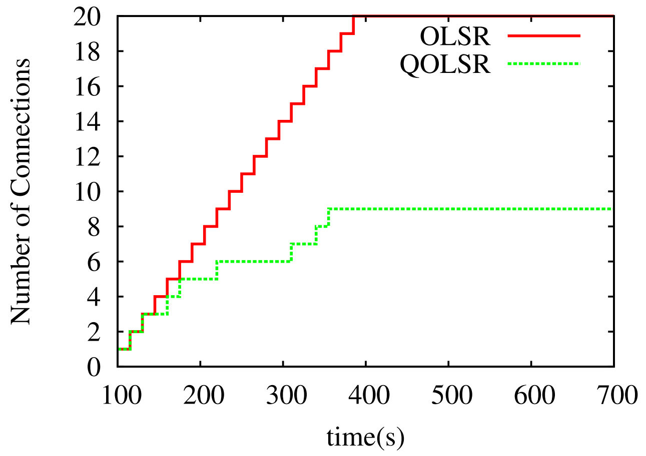

Fig. 3. Connection setup.

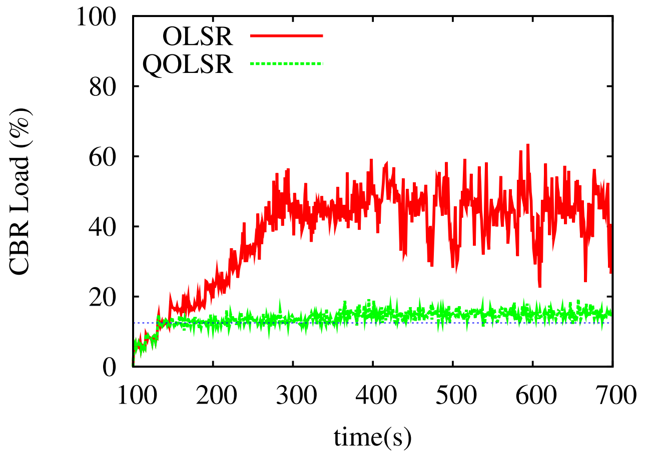

Fig. 4. Maximum occupancy.

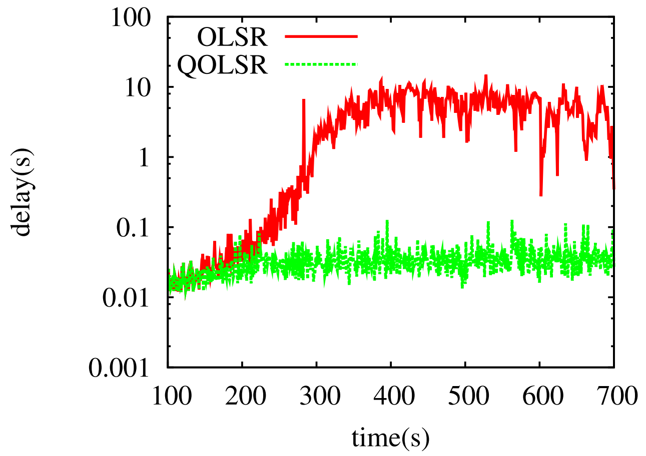

Fig. 5. Maximum delay.

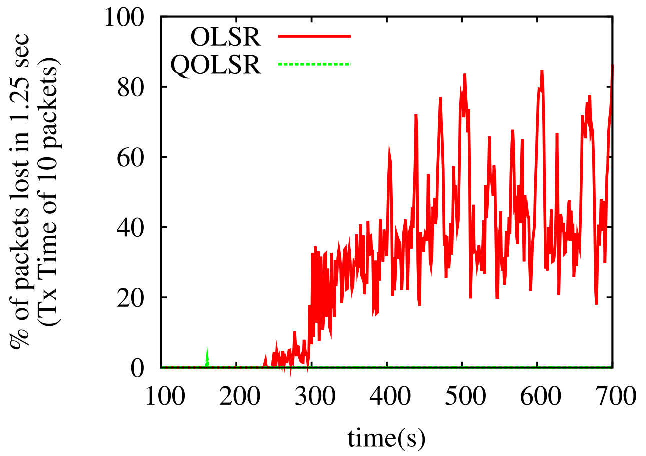

Fig. 6. Maximum loss.

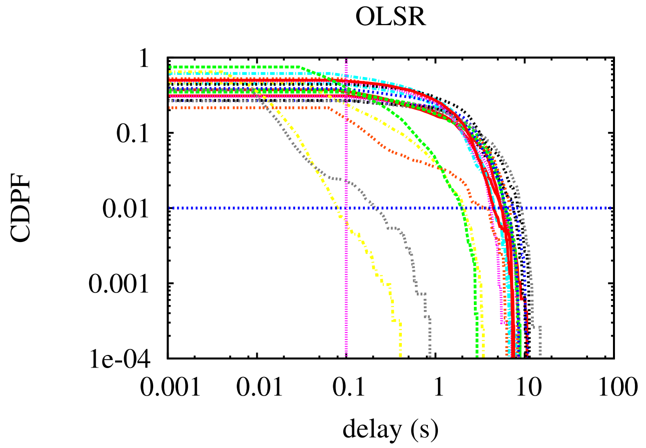

Fig. 7. OLSR delay histogram.

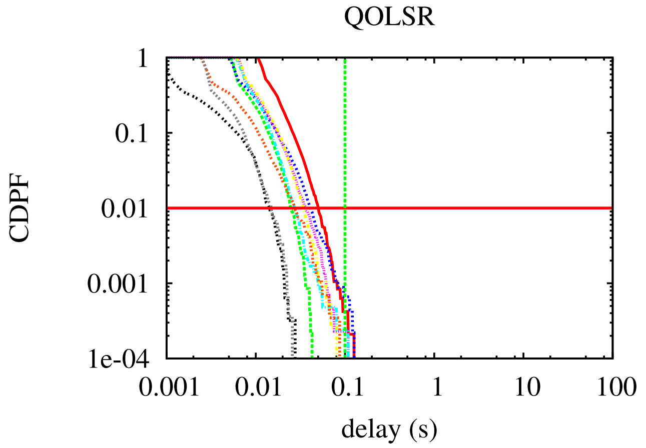

Fig. 8. QOLSR delay histogram.

with OLSR but only 9 with QOLSR (the others are blocked). Since these results are influenced by many random factors, we repeated the simulation many times each time using different node placements. Figure 9 shows the results of 400 simulation runs, each time using different node placements. QOLSR has an average of 7.5 connections and a standard deviation of 2 connections.

Figure 4 plots at each time t the maximum CBR Load (defined in section II as *QoS Load Demand*) observed by the most congested node at this moment. This load is measured at each node counting the transmission time of MAC frames carrying CBR packets that are seen (transmitted or received) by the node (including the collisions). This figure validates that the QoS constraint is satisfied, i.e. the maximum CBR Load is $\leq 12.5\%$.

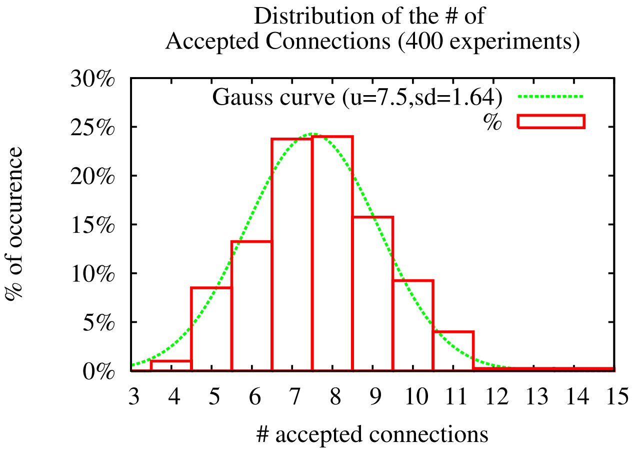

Fig. 9. % of accepted connections by QOLSR

This value is exceeded a bit among other reasons, because of the headers (the 500 bytes packet size does not include the IP header, neither the 802.11 header, thus, the occupancy at the MAC is in fact $12.5 \times 572/500 = 14.3\%$). Thus, we conclude from Fig. 4 that the QoS constraint is satisfied.

Fig. 4 also shows that, using OLSR, the MAC becomes congested at around 300 seconds (when only 14 of the 20 connections have started). This may be noticed by the fact that the CBR Load does not increase any more, although new connections are established. This is confirmed by Figs. 5 and 6. The first one depicts the maximum end-to-end delay of CBR packets, and the second one depicts the maximum percentage of packets lost by the connections, measured in intervals of 1.25 s (the transmission time of 10 packets). These figures shows us that QOLSR is not only successful in avoiding network congestion, but also in avoiding the packet losses and increased delays that occur when the network becomes congested. Compared to QOLSR, OLSR behaves much worse since it loses up to 80% of the packets at some instances.

It is also interesting to know how may connections are suffering from congestion. Figures 7-8 show transmission delay Complementary Distribution Probability Functions (CDPF), i.e. prob$\{$transmission delay $> x\}$, for all the established connections (20 with OLSR and 9 with QOLSR). Figure 7 shows us that using OLSR, except for two flows all the others have a 10% chance of having at least a 1s delay. With QOLSR on the other hand the majority of the flows have end-to-end delays smaller than 0.1 ms with a probability higher than 99%.

VI. CONCLUSIONS AND FURTHER WORK

In this paper we have described a bandwidth reservation scheme for ad-hoc networks that satisfies the following QoS constraint: “The load demand offered to the wireless media by the QoS traffic observed at any mobile node (MN) is $\leq Q$”. The parameter Q is dimensioned such that delays are acceptable for QoS connections. Our reservation scheme only requires that MNs know the normalized bandwidth reservation and maximum available bandwidth of their neighbors. These quantities can be easily advertised by means of hello packets. We also give the CAC rules that MNs should apply to new connections requiring QoS.

We have described how to integrate our reservation scheme in the OLSR ad-hoc routing protocol and we have implemented the protocol using the ns simulator. We have have simulated OLSR with and without our reservation scheme. The following items summarize our findings:

- Ad-hoc networks can easily become congested by QoS traffic (opposite to TCP, this kind of traffic typically doesn’t have congestion control mechanisms).
- Congestion can easily extend to most of the network introducing high delays and losses, thus, damaging most of the connections having QoS requirements.
- Our reservation scheme provides a feasible way to avoid congestion, thus, guaranteeing QoS requirements to ongoing connections.

Further work: In the simulations carried out in this paper we have used static MNs (without movement). If MNs move, they may enter in coverage with new MNs, producing *QoS violation* (i.e. breaching the QoS constrain). A mechanism in needed to cope with this situation. For instance, the MNs receiving hello packets from a new MN such that a QoS violation occurs, may send a *Route Error* to some connections such that they look for another path that fulfills the QoS constrains. Other reasons may produce QoS violations, e.g. due to transient periods, or due to the establishment of a path having unnecessary jumps inside the coverage of another MN.

Another problem arises when a link is broken and a set of connections that traverse that node loose the reserved path to their destination. A reservation recovery must be initiated on the nodes whose flows have lost the QoS reservations. Furthermore, a mechanism to free the existing reservations on the broken path is needed. This mechanism may use timers that free a reservation in a node if the interval of time after forwarding a packet belonging to a flow is higher than certain value.

REFERENCES

[1] H. Xiao, W. G. Seah, A. Lo, and K. C. Chua, “A flexible quality of service model for mobile ad hoc netowrks,” in *Proceedings of the IEEE Vehicular Technology Conference, Tokio, Japan*, May 2000.

[2] S. Lee, G. Ahn, and A. T. Campbell, “Improving UDP and TCP performance in mobile ad hoc networks with INSIGNIA,” in *IEEE Communications Magazine*, June 2001.

[3] R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: A COre-Extraction Distributed Ad Hoc Routing Algorithm,” in *IEEE JSAC, Special Issue on Ad Hoc Networks, Vol 17, N^o 8*, August 1999.

[4] R. Ramanathan and M. Streenstrup, “Hierarchically-organized, multihop wireless networks for quality of service support,” in *ACM Mobile Networks and Applications, Vol 3, N^{o}1*, June 1998.

[5] Q. Xue and A. Ganz, “Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks,” *Journal of Parallel and Distributed Computing*, no. 63, pp. 154–165, 2003.

[6] M. Kazantzidis, M. Gerla, and S. Lee, “RFC 3697: Permissible throughput network for adaptative multimedia in AODV MANETs,” in *IEEE ICC 2001*, 2001.

[7] P. J. T. Clausen, “RFC 3626: Optimized Link State Routing Protocol (OLSR),” Oct 2003.

[8] H. Badis and K. A. Agha, “QOLSR, QoS routing for Ad Hoc Wireless Networks Using OLSR,” *European Transactions on Telecommunications*, vol. 15, no. 4, 2005.

[9] “NRLOLSR Implementation from the NRL (Naval Research Laboratory).” [Online]. Available: http://pf.itd.nrl.navy.mil/projects.php?name=olsr

[10] “The Network Simulator ns-2.” [Online]. Available: http://www.isi.edu/nsnam/ns