

# **UNIFIED SUPPORT FOR QUALITY OF SERVICE METRICS MANAGEMENT IN MOBILE AD HOC NETWORKS USING OLSR**

Djamal-Eddine MEDDOUR, Laurent REYNAUD, Yvon GOURHANT, Bertrand MATHIEU

*France Telecom R&D Division 2, Avenue Pierre Marzin 22307 Lannion CEDEX, France*

**Abstract:** This article focuses on technical issues related to quality of service provisioning at routing layer in ad hoc networks. It describes the design and implementation of a unified support for quality of service (QoS) metrics within the routing protocol OLSR. This is achieved by the extension of both signalling messages and route calculation process. Major benefits of the proposed approach are to allow dynamic enforcing and adaptation of QoS metrics according to policies defined into the network. QoS metrics information are inserted in a generic way within OLSR signalling messages taking advantage of Linux kernel plug-ins. These messages are used by the routing process in order to compute routes with respect to the chosen QoS metrics.

**Key words:** Routing protocol, OLSR, QoS metrics, kernel Plug-ins.

## **1. INTRODUCTION**

Ad hoc networks [1] [2] emerged as an outcome of several projects on mobile computing and packet radio networks financed by the DARPA (Defense Advanced Research Project Agency) during late 60s [3]. One of the key concepts of this paradigm is that the terminals communicate directly between themselves in a multi-hop fashion in an infrastructure-less environment. Each terminal will thus behave both as a router as well as an end-terminal. Consequently, new routing protocols are designed and applied in order to handle terminal mobility and enable multi-hop communication for the network.

In conjunction with the maturity observed on the routing protocol design [4][5], implementation and test, the growing need for multimedia application highlights the need to set up an accurate quality of service support for ad hoc networks. The quality of service provisioning in ad hoc networks is a difficult goal to achieve and the inherent properties of ad hoc network (shared links, limited resources, malign users ...) exemplify the complexity to enforce an effective solution. Nevertheless, several solutions treating QoS provisioning in ad hoc networks have been proposed at various levels of the network stack, more particularly at the routing level. At the routing level, the quality of service provisioning is addressed through the optimization of one or several metrics in the route calculation process. Many metrics may be considered like energy [6], node load [7], bandwidth or delay [8][9], node capacity [10], link stability [11]....

To recapitulate, what commonly follows is a set of two distinct QoS solutions:

- Firstly, most commonly admitted solutions narrow the scope of the problem to a very specific issue. (E.g. security, Energy, link stability ...). This brings good overall performance with respect to the chosen metric, but it targets only a very specific issue of QoS in ad hoc networks, and cannot be considered as a global QoS solution. There is a multiplication of the different QoS models that target different environments.
- Secondly, there are some solutions which try to define statically an extensive set of metrics combinations [12]. However, the definition of an optimal QoS metrics combination is still an NP-hard problem [13] which means that the greater the amount of metrics in a combination is, the more complex the combination maximization will become. At most, what can be expected with these approaches is a comprehensive and complex QoS model that, unfortunately, when applied to a given ad hoc routing protocol, will often dramatically decrease the overall performance. This performance loss is of concern if one or many of the QoS criteria are inappropriate within the ad hoc network where the model is deployed.

However, the diversification of application requirements and the dynamic context of ad hoc networks highlight the insufficiency of such approaches which may consider only one metric or metrics heuristic with a given routing protocol. In order to deal with these limitations, a unified framework for quality of service metrics support is desirable as an alternative to meet the need of a dynamic strategy to set up and adapt QoS metrics at routing level following policies defined in the network. To achieve this goal, we propose a QoS metric management architecture that operates distinctly from the routing module. QoS metrics information are inserted on the routing signaling messages in a generic way taking advantage of Linux kernel

plugins, and these enhanced messages are therefore used by the routing algorithm itself or transferred to an external module in order to compute routes with respect to the chosen QoS metrics.

## 2. MOTIVATIONS

Existing QoS approaches for ad hoc networks advocate the use of a specific metric or a combination of several metrics applicable to explicit use-cases (bandwidth, delay, reliability, energy etc). These solutions utilize static QoS management. When the corresponding QoS metric chosen by the routing protocol does not meet the application requirements, the global QoS efficiency achieved will not be optimal. For instance, the use of delay metric is not judicious if we want to satisfy bandwidth-sensitive applications and vice versa. Moreover, ad hoc network heterogeneity (in term of type of terminal, attached or not to a reliable infrastructure, communication sensibility...) must be taken into account for the design of an effective QoS solution.

To deal with this shortcoming, we propose and implement a new approach that aims at providing an open and generic framework for the management of QoS metrics at routing level. Our approach allows dynamic adaptation of QoS metrics with respect to predefined ad hoc network QoS policies. The use of such a solution allows more effective QoS management. Routing based application requirements is also enabled.

## 3. OUR APPROACH

In this section, we present our approach allowing a dynamic adaptation of the QoS metric at the routing protocol level. Even if the solution principals could be applied for any routing protocol, we advocate in our case the use of the proactive routing protocol OLSR.

### 3.1 Solution requirements

In order to enable a dynamic QoS metrics adaptation in ad hoc networks, the following modules are needed:

- **Extended routing protocol:** signalling message should be extended to carry QoS metrics information. The route calculation process is also modified to be able to compute routes with respect to the chosen metrics.

- **Network level QoS metrics manager:** to ensure an effective QoS metrics choice, a network level manager is required. Its decision must be made based on the available network resources and the applications needs; this decision is then transferred to all the nodes involved into the routing process. This mechanism could be centralized or fully distributed (ad hoc network relayed to a fixed infrastructure/pure ad hoc network).

In this work, we address mainly the routing protocol part. For the choice of the QoS metrics, we set up a centralized mechanism that uses a simple algorithm to take its decision (an example of such algorithm is given in section 4.1). Nevertheless, for more reliability, a distributed solution is needed and we intend to investigate such a problem in our future works.

### 3.2 OLSR description

Type	Description
HELLO.	Neighbors detection
TC.	MPR declaration
MID.	interfaces declaration
HNA.	Gateways detection s

Table 1. List of messages used in OLSR

The standard routing protocol OLSR [4] is a link-state protocol which provides optimal routes in term of number of hops. It is particularly suitable for dense networks where the use of its broadcast technique via Multi-Point Relays (MPR) seems more effective in such a context. MPR are a sub set of 1 hop neighbours which allow a node to reach its 2 hops neighborhood. OLSR ensures the construction of routing table following a periodic exchange of signalling messages (see Table 1). These letters are then used to build a graph of the topology which will be used for route calculation.

Furthermore, we can distinguish two main components for route calculation within OLSR

- **Topology maintenance:** in order to handle topology changes due to node motion (mobility, entrance and exit), a periodic exchange of signalling messages is used and then network topology information is gathered, thus, a graph of the topology is built. Moreover, this mechanism encompasses a system that allows minimizing global network overhead by the selection of special nodes on the networks called MPR (Multi Point Relay). In OLSR, only nodes selected as MPR will act as routers and therefore will retransmit networks traffics.

- Path selection: the Dijkstra algorithm is applied on the graph of topology in order to compute best paths between the network nodes.

We note here that the standard OLSR is a non QoS-aware routing protocol since the route selection criterion is the number of hops only.

### 3.3 Architecture of the proposed solution

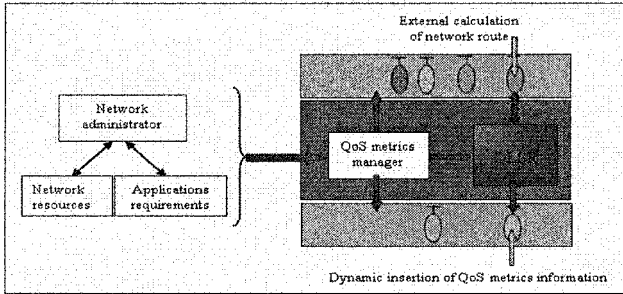


Figure 1. Proposed architecture

The complete architecture is presented in figure 1. A central entity playing the role of a network administrator monitors the available network resource and the application requirements. It takes its decision based on this information and the pre defined QoS policies. This decision is then transmitted to and enforced by the network nodes.

The main idea of our approach is to re-use the existing signalling messages of OLSR without redefining new messages. To be able to manage any type of metric, the following extensions are introduced:

- Extending existing signalling message TC in manner that can transport QoS metrics related information. The core OLSR is therefore modified in order to enable the treatment of the new TC message.
- Dynamic insertion and calculation of QoS metrics information via external modules. Moreover, we introduce the possibility to manage several metric through an external route calculation module. Indeed, the routing protocol transmits to this module the graph of topology enriched by QoS metrics information. Therefore, this module can directly update the kernel routing table.
- The routing protocol connected to a component so-called "QoS metrics manager" which transmits to the routing protocol the QoS metrics to apply in the network. Furthermore, it is responsible for the set up of the needed modules for both dynamic QoS insertion and route calculation.

### 3.3.1 Extended TC messages

The integration of QoS support within OLSR consists on building a topology graph enriched by QoS metrics information. As the construction of such a graph is based on TC messages, the latter will thus be used to transport QoS information. A "normal" TC message includes these fields:

- ANSN: presents a number of sequences in order to identify the most recent TC;
- Reserved: Field not used, fixed by defect at 0;
- Advertised Neighbour Main Address 1..N: The addresses of MPR nodes of the TC message transmitter.

We propose to introduce a QoS metrics support within the TC message. The Field RESERVED is used to indicate the number of neighbours transported in the message TC and to delimit the part of the original TC message and the extension. The extension part contains the following fields (see figure 2):

- For each address, we associate the value of metric defined below in the message (Example: bandwidth or delay towards this neighbour).
- Nb\_Metrics: an integer of 1 byte, it indicates the number of metrics transported by the message TC, It is followed by a field on 1 byte not used;
- A pair (metric type, metrics id) of integers of 1 byte each one. It represents respectively the type of metric, and its identifier.

### 3.3.2 Dynamic insertion of QoS metrics information

In order to achieve dynamic insertion of QoS metrics information, we advocate the usage of PromethOS [14]. PromethOS is an extension of the standard Linux mechanism Netfilter which offers a support to set up on the fly software components called "plug-ins", which has the capacity to apply treatments to the received packages. Plug-ins are kernel modules that are used to reprogram the behaviour of the IP stack and are installed and controlled by PromethOS framework.

Therefore the insertion of QoS metrics values will be done via the use of a PromethOS plug-in. This is useful to intercept the TC messages during transmission, and to modify them in order to include various data related to QoS metrics.

Once the decision concerning the QoS metrics to apply is transmitted to the "*QoS metrics manager*", it deploys the component "*metrics calculation*" which calculates the QoS metric value and is carried out in user space. Thereafter, these values will be transferred to the kernel module "*metrics\_add*" which inserts them in the TC message. The frequency of

calculation of these parameters depends on the sensitivity of the metric. Metrics like the bandwidth, the time or link stability should be calculated more frequently than metric like energy.

ANSN		Nb_Neighbor	
Advertised Neighbor Main Address			
Advertised Neighbor Main Address			
...			
Nb_Metric		Reserved	
Type_M <sub>1</sub>		Id_M <sub>1</sub>	
Type_M <sub>2</sub>		Id_M <sub>2</sub>	
...			
Val (neighbor <sub>1</sub> _M <sub>1</sub> )			
Val (neighbor <sub>1</sub> _M <sub>2</sub> )			
...			
Val (neighbor <sub>1n</sub> _M <sub>n</sub> )			

Extension {

Figure 2. Measured delay for all approaches (performed on the time before metrics changing)

### 3.3.3 Route calculation

The new TC messages once collected on the level of OLSR will be used to build a graph of the topology, each element of the graph being represented in the following way < @source, @destination, metric1, metric2..... >. The route calculation will be done based on the chosen metric.

In order to be able to compute routes based on multiple metrics and overcome limitation present in the current algorithm, we provide an API that allows an external route calculation. An external module that implements a heuristic of route calculation methods based on needed metrics should be used. Information concerning the graph of topology is transferred to this module. Therefore, routes are calculated and injected into the Linux kernel.

## 4. EXPERIMENTATION

The experimentation presented here shows how a dynamic adaptation of QoS metrics can respond efficiently to the observed changes of network conditions. We integrated the proposed scheme within the OLSR implementation provided by [15].

Figure 3 depicts the ad hoc network topology used for our experimentation. The network is composed of 7 nodes running the routing protocol OLSR including our modification, implemented under Linux. All nodes are equipped by 802.11b wireless cards with a maximum of 11Mbps/s

data rates. Within this topology, a source node transmits several traffics with different QoS requirements towards the destination.

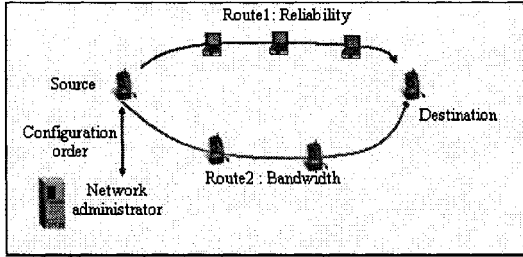


Figure 3. Testbed configuration

## 4.1 Scenario

Figures 4 and 5 show how the use of bandwidth metric allows us to obtain good bandwidth level. This will be advantageous for the applications requiring important bandwidth. Nevertheless it penalizes applications which are delay sensitive. We could observe the same phenomenon when using the metric delay. Our solution based on the use of a dynamic QoS metric will enable us to meet application needs, by respecting the constraints of the applications (in term of metric) and to obtain the best compromise.

Two paths between the source node and the destination node with different capacities are set up, the first one (Route 1) composed of 3 PC with high capacities proposing more reliability but less bandwidth, the second one (Route 2) composed of 2 PDA with low capacities but it contains less hops, it hence offers more bandwidth but a less reliable path. The reliability in our case is related to terminals capacities.

In this topology, we consider two kinds of traffic conveyed in the network between the source node and the destination node. The first one uses TCP and requires bandwidth (example: HTTP, FTP), the second using UDP requires real time and more reliability (Example: multimedia...).

In order to enforce the choice of QoS metrics, a module emulating the network administrator is integrated in the source node. It sets up a system called **Event\_collector** that gathers the application related QoS requirements. The following algorithm is then run to decide on the QoS metrics that are to apply to the network (note that such an algorithm must be modified to fit the requirements of each ad hoc network. For instance, secure routes should be privileged in the case of a military based communication).



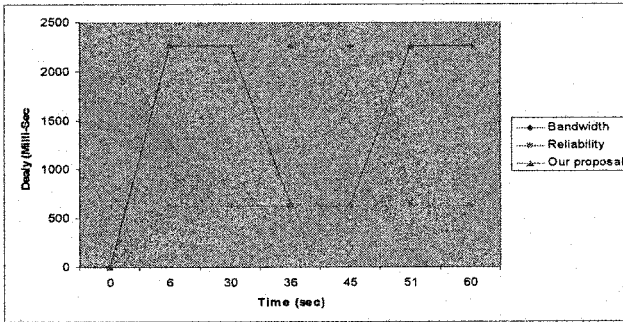


Figure 4. Measured delay (performed on the time before metrics changing)

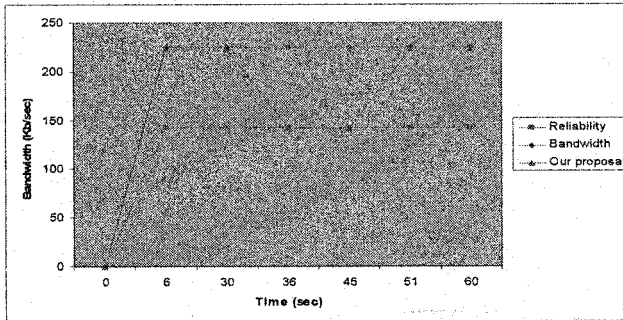


Figure 5. Measured bandwidth (measures performed on the time before metrics changing)

This algorithm tests whether all the applications have chosen the correct metric i.e. to identify if there is no conflict between the expressed QoS needs. Otherwise, the reliability metric will be chosen by the network administrator. Therefore, following the QoS metrics chosen by the network administrator, traffic will be routed through route 1 (reliability) or route 2 (bandwidth).

We can thus notice the effectiveness of our approach from the obtained results. Moreover, they show clearly the need for a dynamic management of QoS metric in ad hoc networks. Compared to a standard approach with a static QoS metric, our proposal offers needed flexibility with respect to the QoS policy defined in the network.

## 5. CONCLUSION

In this article, we have described a novel approach for the design and the implementation of QoS support at routing level. Our proposal enables dynamic adaptation of QoS metrics to meet predefined ad hoc network QoS policies.

The evaluations carried out show clearly the generated QoS profit generated by the use of a dynamic metric compared to a classic approach. This profit is achieved with a negligible network overhead. Also, our approach keeps compatibility of the non-QoS aware basic protocol, which is relating to the fact that QoS information is inserted in a separate way from neighborhood information.

## REFERENCES

- [1] IETF MANET Working Group. MANET Charter, 2000.
- [2] IRTF RRG Ad hoc Network Systems Research Subgroup.
- [3] J. Freewerby et al, "A DoD Perspective on Mobile Ad Hoc Networks", Ad Hoc Networking, ed. C. E. Perkins, Addison-Wesley, 2001, pp. 29–51
- [4] T. Clausen, P. Jacquet "Optimized Link State Routing Protocol (OLSR)". RFC3626.
- [5] D. B. Johnson et al "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks" IETF Draft, April 2004.
- [6] J. Gomez et al "PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks", ACM/Kluwer Journal on Wireless Networks (WINET), September 2003.
- [7] S.-J. Lee et al "Dynamic Load-Aware Routing in Ad hoc Networks", IEEE ICC 2001.
- [8] H. Badis et al "QoS for Ad Hoc Networking Based on Multiple Metrics: Bandwidth and Delay", IFIP MWCN 2003
- [9] C. Perkins, E. Belding-Royer, and S.R. Das "Quality of Service in Ad hoc On-Demand Distance Vector Routing", IETF Internet Draft, draft-ietf-manet-qos-00.txt, July 2000.
- [10] R. Meraihi et al "Cross-layer QoS and terminal differentiation in ad hoc networks for realtime service support", IFIP MedHocNet 2003,
- [11] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm", IEEE Journal on Selected Areas in Communications,
- [12] X. Yuan, "Heuristic Algorithms for Multi-Constrained Quality of Service Routing", IEEE/ACM Transactions on Networking, April 2002
- [13] Z. Wang et al "Quality-of-service routing for supporting multimedia applications", IEEE JSAC, September, 1996.
- [14] R. Keller et al "PromethOS: A Dynamically Extensible Router Architecture Supporting Explicit Routing", Proceedings of IWAN 2002
- [15] <http://qolsr.lri.fr> : OLSR implementation from the LRI.