An Efficient QOLSR Extension Protocol For QoS in Ad hoc Networks

Hakim Badis*† and Khaldoun Al Agha*†
*LRI Laboratory, University of Paris XI, 91405 Orsay, France
†INRIA Laboratory, Rocquencourt, 78153 Le Chesnay, France

***Abstract*— Quality of Service (QoS) support in Mobile Ad hoc NETworks (MANETs) is a challenging task. We have developed the QOLSR protocol, a QoS routing over OLSR protocol introducing more appropriate metrics than the hop distance as bandwidth and delay. This paper discusses a proposed extension to the QOLSR protocol to provide integrated services, i.e., to support real-time as well as the current non-real-time service of IP. This extension is necessary to meet the growing need for real-time service for a variety of new applications, including teleconferencing, remote seminars, telescience, and distributed simulation. We include the various QoS parameters in the IPv6 flow label. The performance of our extension is extensively investigated by simulation. Our results indicate that the attained gain by our proposal represent an important improvement in such mobile wireless networks.**

I. Introduction

An ad hoc network is a dynamic multi-hop wireless network that is established by a group of mobile hosts on a shared wireless channel by virtue of their proximity to each other. Since wireless transmissions are locally broadcast in the region of the transmitting host, hosts that are in close proximity can hear each other and are said to be neighbors. The transitive closure of the neighborhood of all the hosts in the set of mobile hosts under consideration forms an ad hoc network. Thus, each host is potentially a router and it is possible to dynamically establish routes by chaining together a sequence of neighboring hosts from a source to a destination in the ad hoc network.

Most routing protocols for the mobile Ad hoc networks (MANETs) [1], such as OLSR [2], AODV [3], DSR [4], are designed without explicitly considering QoS of the routes they generate. The number of hops is the most common criterion adopted by such proposed routing protocols. It is becoming increasingly clear that such routing protocols are inadequate for multimedia application, such as video conferencing, which often require guaranteed QoS. QoS routing requires not only finding a route from a source to a destination, but a route that satisfies the end-to-end QoS requirement, often given in terms of bandwidth or delay. QoS is more difficult to guarantee in ad hoc networks than in most other networks, because the wireless bandwidth is shared among adjacent nodes and the network topology changes as the nodes move.

The link state routing approach makes available detailed information about the connectivity and the topology found in the network. Moreover, it increases the chances that a node will be able to generate a route that meets a specified set of requirement constraints. OLSR protocol is an optimization over the classical link state protocol for the mobile ad hoc networks. It performs hop by hop routing, i.e., each node uses its most recent information to route a packet.

We have proposed the QOLSR protocol in [5]–[7], which is an enhancement of the OLSR routing protocol to support multiple-metric routing criteria. However, the QOLSR protocol has any distinction between the real-time and non-real-time services of IP.

With the emergence of bandwidth-greedy and/or time-sensitive applications, the need of guaranteed QoS of this applications becomes of prime importance in the networks. For this purpose, many approaches have been developed so far to provide real-time QoS guarantees for time-sensitive applications. In the Internet community, the two widespread approaches are IntServ [8] and DiffServ [9]. In this paper we present a QOLSR's extension for Integrated Services in ad hoc networks including the various QoS parameters (bandwidth, delay, buffer requirements, ...) in the IPv6 Flow Label [10] that may be requested by any application. The IPv6 Flow Label was still experimental, and subject to change, as the requirement for flow support in the Internet were evolving. The last several years of work in IETF on Internet Protocols Quality of Service (IntServ, and DiffServ) has provided a more solid and ample architectural perspective, and framework for the standardization of the IPv6 Flow Label. IntServ and DiffServ present two alternative solutions of resolving QoS problems in the Internet.

This paper talks about the design of Quality of Service (QoS) in IPv6. Though IPv6 main header has a 20-bit Flow Label field for QoS implementation purposes, it has not yet been exploited. Few Internet drafts give various definitions of the 20-bit Flow Label in IPv6, each with its own advantages and disadvantages. This paper provides an efficient Quality of Service using a hybrid approach. we present performance comparison of QOLSR extension routing vs. standard QOLSR routing using a scalable simulation model.

This paper is organized as follows. Section II presents IPv6 Flows, Flow Label and the quality of service mechanisms. Structure and mechanism for the use of the Flow Label are presented in Section III. In section IV we describe our QOLSR protocol, which is an enhancement of the OLSR routing protocol to support multiple-metric routing criteria and routing table calculation. Therefore, we validate the proposal by means of performance evaluation (Section V). Finally, we present our conclusions.

II. IPv6 Flows and Flow Label

A flow is a sequence of packets sent from a particular source, and a particular application running on the source host, using a particular host-to-host protocol for the transmission of data over the Internet, to a particular (unicast or multicast) destination, and particular application running on the destination host, or hosts, with a certain set of traffic, and quality of service requirements.

As IPv6 relies on quality of service mechanisms defined by the Integrated Services Architecture or the Differentiated Services Quality of Service Architecture, it is worth considering those architectures flow definitions. The IPv6 Flow Label is defined as a 20-bit field in the IPv6 header which may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or real-time service.

A. Integrated Services Flows

The Integrated Services architecture defines a flow as an abstraction which is a distinguishable stream of related datagrams that results from a single user activity and requires the same QoS. For example, a flow might consist of one transport connection or one video stream between a given host pair. It is the finest granularity of packet stream distinguishable by the Integrated Services.

For the purpose of traffic control (and accounting), each incoming packet must be mapped into some class; all packets in the same class get the same treatment from the packet scheduler. A class might correspond to a broad category of flows, e.g., all video flows or all flows attributable to a particular organization. On the other hand, a class might hold only a single flow. A class is an abstraction that may be local to a particular router; the same packet may be classified differently by different routers along the path. For example, backbone routers may choose to map many flows into a few aggregated classes, while routers nearer the periphery, where there is much less aggregation, may use a separate class for each flow.

B. Differentiated Services Flows

The Differentiated Services architecture defines a flow or microflow as a single instance of an application-to-application flow of packets, which is identified by the source address, source port, destination address, destination port and protocol id (fields in the IP and host-to-host protocol headers).

Furthermore, this architecture defines a classifier as: a mechanism that selects packets in a traffic stream based on the content of some portions of the packet header. Two types of classifiers are defined. The BA (Behavior Aggregate) Classifier classifies packets based on the DS codepoint only. The MF (Multi-Field) classifier [9] selects packets based on the value of a combination of one or more header fields, such as source address, destination address, DS field, protocol ID, source port and destination port numbers, and other information such as incoming interface.

III. Structure and Mechanism for the Use of the Flow Label

The 20-bits of the Flow Label should be defined in an appropriate manner so that various approaches can be included to produce a more efficient hybrid solution. For example the hybrid approach suggested in [11], in witch the first three bits of the IPv6 Flow Label are used to define the approach used. 000: Default, 001: A random number is used to define the Flow Label, 010: The value given in the Hop-by-Hop extension header is used instead of the Flow Label, 011: PHB ID, 100: A format that includes the port number and the protocol in the Flow Label is used, 101: Various QoS parameters in the IPv6, 110 and 111: Reserved for future use. The next 17 bits are used to define the format used in a particular approach.

In this paper, we interest only to include various QoS parameters in the IPv6 Flow Label that may be requested by any application. The various QoS parameters are: bandwidth, delay or latency, jitter, packet loss, buffer requirements. As packet loss and the jitter are often desired to be of minimum value by any application, these two parameters may not be defined in the Flow Label field itself. Instead, if needed, the Hop-by-Hop EH space can be effectively used to specify these parameters. Bits thus saved in the Flow Label can be effectively used for more demanding purposes. The QoS parameters that are to be included in the Flow Label are: bandwidth (to be expressed in kbps), delay (to be expressed in nanoseconds) and buffer requirements (to be expressed in bytes). If we use the approach suggested in [11], as there are only 17 bits left, the optimal use of the bits is very important so as to obtain the maximum information out of those 17 bits. The first bit out of these 17 bits is used to differentiate between the hard real time and soft real time applications. This bit is set to 0 for soft real time applications (as shown in figure 1) and it is set to 1 for hard real time applications (as shown in figure 2).

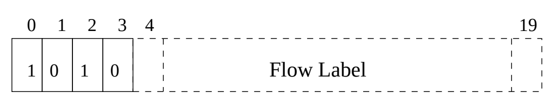

Fig. 1. Soft real time format for the fbw label

Soft Real time applications service is meant for RTT (Real Time Tolerant), which have an average bandwidth requirement and an intermediate end-to-end delay for an arbitrary packet. Even if the minimum or maximum values specified in the Flow Label are not exactly met, the application can afford to manage with the QoS provided.

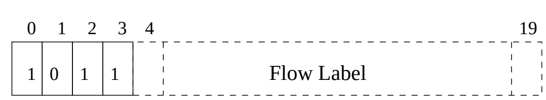

Fig. 2. Hard real time format for the fbw label

Hard Real time applications service is meant for RTI (Real Time Intolerant), which demand minimal QoS parameters like delay, jitter or bandwidth. For example, a multicast real time application (videoconferencing). Delay is unacceptable and ends should be brought as close as possible. For this videoconference (DTVC) case, the required resource reservations are: constant bandwidth for the application traffic and deterministic Minimum delay that can be tolerated. These types of applications can decrease delay by increasing demands for bandwidth. The minimum or maximum values specified in the Flow Label have to be exactly met for these kind of applications.

After keeping one bit for Hard/Soft real time applications, we are left within 16 bits for defining the Flow Label. For the bandwidth parameter, we specify 6 bits out of the 16 bits to be used for specifying the bandwidth value. The application can demand for a minimum or maximum value of bandwidth. So one bit out of these 6 bits is used for specifying whether the application is asking for a minimum expected value of bandwidth or a maximum. For buffer requirements parameter, we specify next 5 bits out of the 16 bits to be specifying the buffer value. The last 5 bits out if the 16 bits are to be used for specifying the delay value.

The application specifies the desired QoS and the Flow Label field in the IPv6 header is filled based on the QoS asked by the application. The application has the flexibility of specifying which format it wants to use for getting the desired QoS.

IV. The QOLSR extension protocol and routing table calculation

QOLSR is an enhancement of the OLSR routing protocol to support multiple-metric routing criteria. Each node calculates the delay and bandwidth information for each of its neighbors. No additional control traffic is generated (only *hello* and *TC* messages). As in the standard OLSR, link state information is generated only by nodes selected as MPRs. This information is then used by route calculation. QOLSR requires only partial link state to be flooded in order to provide optimal paths in terms of bandwidth and delay. The QOLSR does not require any changes to the format of IP packets. Thus any existing IP stack can be used as is: the protocol only interacts with routing table management. A detailed description can be found in [5]–[7].

Let $G = (V, E)$ be the network with $|V|$ nodes and $|E|$ arcs and $p = (i, j, k, ..., q, r)$ a directed path. For the Hard Real time applications we use a shortest path algorithm. For delay or jitter metric, each arc (i, j) in the path p is assigned a real number del_{ij}. When the arc (i, j) is inexistent or j is not a MPR of i (Referring to the OLSR routing mechanism), then $\mathrm{del}_{ij} = \infty$. Let $\mathrm{del}(p) = \mathrm{del}_{ij} + \mathrm{del}_{jk} + ... + \mathrm{del}_{qr}$. The routing problem is to find a path $p*$ between i and r so that $\mathrm{del}(p*)$ is the minimum. In such a case, we use the well-known Dijkstra routing algorithm. For bandwidth metric, each arc (i, j) in the path is assigned a real number Bw_{ij}. When the arc (i, j) is inexistent or j is not a MPR of i, $\mathrm{Bw}_{ij} = 0$. Let $\mathrm{Bw}(p) = \min\{\mathrm{Bw}_{ij}, \mathrm{Bw}_{jk}, ..., \mathrm{Bw}_{qr}\}$. The routing problem is to find a path $p*$ between i and r that maximizes $\mathrm{Bw}(p*)$. In order to implement such a metric, we use a variant-Dijkstra algorithm.

For the Soft Real time applications, We consider the Delay and Bandwidth Constrained Least Hop path problem (DBCLH). Given two constants, the minimum bandwidth $\Delta_{bandwidth}$ and the maximum delay Δ_{delay}. The Delay and Bandwidth Constrained Least Hop path problem (DBCLH) is to find a path p from i to r minimal for a hop-count, satisfying $del(p) \leqslant \Delta_{delay}$ and $\mathrm{Bw}(p) \geqslant \Delta_{bandwidth}$. The formal description is:

$\min\{\mathrm{hop}(p) : p \in P(s,t)$ and $\mathrm{del}(p) \leqslant \Delta_{delay}$ and $\mathrm{Bw}(p) \geqslant \Delta_{bandwidth}\}$ where $P(s,t)$ is the set of paths from the source node s to the destination node t, and $\mathrm{hop}(p)$ is the hop-count. It is clear that two additive and one concave metrics problem is *NP-Complete* [12]. We have proposed an efficient heuristic based on the Lagrange Relaxation algorithm to resolve this problem [13]. If there is no path that satisfies the constraints, we use the Best Effort method. Each node in the Ad hoc network makes its decision (next hop) using information in incoming packet and its topology table. To avoid the recomputing of next hop for any received packet, each incoming packet must mapped into some class; all packets in the same class get the same treatment. Choice of a class may be based upon the content of existing packet header(s).

V. Performance Evaluation

The simulation model introduced in [5], [14] is very close to a real Ad-Hoc network operations. At each time, we can detect the position of mobiles by our mobility model. Each node is represented by a subqueue and placed in the region by randomly selecting its x and y co-ordinates. The number of nodes can reach 100000 nodes. With our method, the simulation model is very optimized that enables to reduce the CPU time and consequently to increase the time of simulation. The random mobility model proposed is a continuous-time stochastic process. Each node's movement consists of a sequence of random length intervals, during which a node moves in a constant direction at a constant speed. A detailed description can be found in [5], [14].

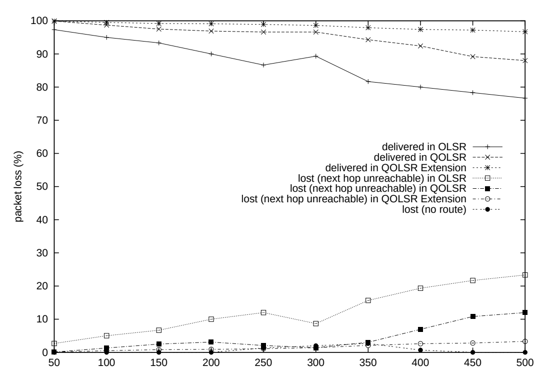

Fig. 3. Data load transmitted in varying node speeds

Figure 3 shows the results of our simulation in which the data packets sent and successfully delivered are plotted against the increasing speed. The speed is increased from $50 meters/minute$ ($3 Km/hr$) up to $500 meters/minute$ ($30 Km/hr$). In this simulation, 50 nodes constitute the network in a region of $1000^2 m^2$, and all the 50 nodes are packet-generating sources. We also keep the movement probability as 0.3, i.e., only 20% of nodes are mobile and the rest are stationary. Each mobile node selects its speed and direction which remains valid for next 60 seconds. We can see that when the mobility (or speed) increases, the number of packets delivered to the destinations decreases. This can be explained by the fact that when a node moves, it goes out of the neighborhood of a node which may be sending it the data packets. There are about 99.97% of packets delivered for QOLSR's extension at a mobility of $50 meters/minute$ (99.92% for QOLSR and 97.3% for OLSR). At a mobility of $500 meters/minute$, 96.2% of packets delivered for QOLSR's extension (88% for QOLSR and76.6% for OLSR). QOLSR's extension has the highest packets delivered because the routes are chosen with respect of the QoS requirements. The data packets are lost because the next-hop node is unreachable. A node keeps an entry about its neighbor in its neighbor table for about 6 seconds. If a neighbor moves which is the next-hop node in a route, the node continues to forward it the data packets considering it as a neighbor. Also, the next-hop is unreachable if there are interferences. Few of packets are also lost because of unavailability of route and it is the same for OLSR with or without QoS. This happens when a node movement causes the node to be disconnected from the network temporarily, until it re-joins the network again.

VI. Conclusions

We have proposed an extension of the QOLSR protocol to provide integrated services. We have used the IPv6 flow label to express the QoS requirements. For the Hard Real time applications we use a shortest path algorithm. For the Soft Real time applications we use our proposed algorithm based and Lagrange Relaxation. The QOLSR's extension produces better performance comparing with the OLSR and QOLSR protocols. To avoid the route decision for any received packet to its destination node, each incoming packet must mapped into some class; all packets in the same class get the same treatment. Simulation results show the performance our QOLSR's extension comparing to the standard QOLSR.

References

[1] "http://www.ietf.org/html.charters/manet-charter.html."

[2] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol," *RFC 3626*, October 2003.

[3] C. Perkins, E. M. Royer and S. R. Das, "Ad Hoc On-Demand Distance Vector (AODV) routing," *RFC 3561*, July 2003.

[4] D. Johnson and Al, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR)," *In IETF Internet Draft, draft-ietf-manet-dsr-09.txt*, April 2003.

[5] A. Munaretto, H. Badis, K. Al Agha and G. Pujolle, "A Link-state QoS Routing Protocol for Ad Hoc Networks," *In the proceedings of IEEE MWCN2002, Stockholm, Sweden*, September 2002.

[6] H. Badis, A. Munaretto, K. Al Agha and G. Pujolle, "QoS for Ad hoc Networking Based on Multiple Metrics: Bandwidth and Delay," *In the proceedings of IEEE MWCN2003, Singapore*, October 2003.

[7] H. Badis, K. Al Agha, A. Munaretto and G. Pujolle, "Optimal Path Selection in a Link State QoS Routing," *In the proceedings of IEEE VTC'04 spring, Italy*, May 2004.

[8] D. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," *RFC 1633*, June 1994.

[9] S. Blake, D. Blake, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for Differentiated Services," *RFC 2475*, Decembre 1998.

[10] C. Partridge, "Using the Flow Label Field in IPv6," *RFC 1809*, June 1995.

[11] R. Banerjee and M. Mahaveer, "A Modefied Specification for use of the IPv6 Flow Label for providing an efficient QoS using a hybrid approach," *In Internet Draft, draft-banerjee-flowlabel-ipv6-qos-03.txt*, April 2002.

[12] F. Kuipers, P. Van Mieghem, T. Korkmaz and M. Krunz, "An Overview of Constraint-Based Path Selection Algorithms for QoS Routing," *IEEE Communications Magazine*, vol. Vol.40, No.12, December 2002.

[13] H. Badis and K. Al Agha, "A Distributed Algorithm for Multiple-Metric Link State QoS Routing Problem," *In the proceedings of IEEE MWCN2003, Singapore*, October 2003.

[14] K. Al Agha and H. Badis, " Scalable model for the simulation of OLSR and Fast-OLSR protocols ," *Med-hoc-Net*, June 2003.