

# Multicast Tree in MPLS Network

Ali Boudani<sup>\*,†</sup>, Bernard Cousin<sup>†</sup>

<sup>\*</sup>Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC)  
IUT Belfort-Montbéliard, BP 71427, pl. Tharradin, 25211 Montbéliard Cedex, France

<sup>†</sup>IRISA/INRIA Rennes, Campus Universitaire de Beaulieu, 35042 Rennes, France

Email: aboudani@pu-pm.univ-fcomte.fr, {aboudani,bcousin}@irisa.fr

**Abstract**—In this paper, we study multicast tree construction in MPLS network. We discuss the difficulty in combining multicast and MPLS in a network. We describe some MPLS proposals for the multicast traffic and we justify the need for defining a new protocol. Thereafter we propose MMT, the MPLS Multicast Tree protocol, which uses MPLS LSP (Label switched Path) between multicast tree branching nodes in order to reduce the multicast routing states in routers and to increase scalability. We present improvements to MMT protocol and we evaluate it in term of scalability and efficiency. Finally, we present simulation results to validate our evaluation and we conclude that the MMT protocol seems promising and well adapted to a possible implementation of multicast traffic engineering in the network.

## I. INTRODUCTION

Increasing the efficiency of Internet resources utilization is very important. Several evolving applications like WWW, video/audio on-demand services, and teleconferencing consume a large amount of network bandwidth. By reducing the number of packets transmitted across the network, the multicast service essentially increases the QoS given to users due to the additional available bandwidth in the network, which improves network performance. MPLS (Multi-Protocol Label Switching) [1] as a traffic engineering tool has emerged as an elegant solution to meet the bandwidth management and service requirements for next generation Internet Protocol (IP) based backbone networks. MPLS is an advanced forwarding scheme that extends routing with respect to packet forwarding and path controlling. Packets are classified easily at domain entry and rerouted faster in the case of link failures. Explicit routes are easily constructed and packets may follow these explicit routes instead of following the traditional shortest route. Once a packet is assigned to a FEC (Forwarding Equivalence Class), no further header analysis is done by subsequent routers in the same MPLS domain. An MPLS header, called label, is inserted for each packet within an MPLS domain. An LSR (Label Switching Router) will use the label as the index to look up in the forwarding table. The packet is processed as specified by the forwarding table entry. The incoming label is replaced by an outgoing label, and the packet is switched toward the next LSR. Before a packet leaves an MPLS domain, its MPLS header is removed. The paths between the ingress LSRs (at the domain entry) and egress LSRs (at the domain exit) are called label-switched paths (LSPs). MPLS uses some signaling protocol such as Resource Reservation Protocol (RSVP) or Label Distribution Protocol (LDP) to set up LSP.

Multicast and MPLS are two complementary technologies. Merging these two technologies, making multicast trees built on top of MPLS networks will enhance the network performance and present an efficient solution for multicast scalability and control overhead problems. Multicast attempts to reduce network bandwidth usage, while MPLS attempts to provide users with needed bandwidth in an appropriate switched-like manner.

The remainder of this paper is organized as follows. In Section II, we present MPLS proposals for multicast traffic and we justify the need for defining a new protocol. In Section III, we describe the MMT protocol (MPLS Multicast Tree) and its extension the MMT2 protocol, which use MPLS LSPs between the multicast tree branching node routers in order to reduce forwarding states and enhance scalability. In Section IV, we evaluate our proposals in terms of scalability and efficiency and we present some simulation results evaluation. Section V is a summary followed by a list of references.

## II. MERGING MPLS AND MULTICAST

MPLS can be deployed in a network to forward unicast traffic through explicit routes and multicast traffic by using explicit trees (an explicit tree is built through policies and explicit routes instead of topology). But multicast traffic has specific characteristics due to the nature of the multicast routing protocols [2]. Indeed, the multicast routing is based on multicast IP address and this is why it is very difficult to aggregate multicast traffic since receivers belonging to the same group can be located at multiple localizations.

A framework for IP multicast deployment in an MPLS environment is proposed in [2]. Issues arising when MPLS techniques are applied to IP multicast are overviewed. Following characteristics are considered: aggregation, flood and prune, co-existence of source and shared trees, uni/bi-directional shared trees, encapsulated multicast data, loop free ness and RPF (Reverse Path Forwarding) check. The pros and cons of existing IP multicast routing protocols in the context of MPLS are described and the relation to the different trigger methods and label distribution modes are discussed. The framework did not lead to the selection of one superior multicast routing protocol but it concluded that different IP multicast routing protocols could be deployed simultaneously in the Internet. It should be noted that the multicast tree structure requires P2M (point-to-multipoint) LSP or even MP2MP (multipoint-with-multipoint) LSP establishing. In the current architecture

of MPLS, only point-to-point LSP were studied. MPLS does not exclude other types of LSP, but no mechanism was standardized so far.

MPLS labels support the aggregation of trees but does not solve the problem completely. Indeed, algorithms should be designed to aggregate unicast flows with multicast flows and also aggregate multiple multicast flows together. Unfortunately, the current studies on multicast aggregation are limited to the aggregation of the routing states in each router rather than to the LSP aggregation. For further details, we recommend the reader the broad literature which exists in this subject, or to refer to our work ([3], chapter 4 of [4]). In this paper, we are concerned mainly in two MPLS multicast routing protocols : PIM-MPLS [5] and Aggregated multicast [6].

#### A. PIM-MPLS

Using PIM-SM *join* messages to distribute MPLS labels for multicast routes is proposed in [5] (called hereinafter PIM-MPLS). A piggy-backing methodology is suggested to assign and distribute labels for multicast traffic for sparse-mode trees. The PIM-SM *join* message should be expanded to carry an MPLS label allocated by the downstream LSR. Modifications to PIM-SM make this proposal not easily accepted by working groups dealing with multicast in the IETF. Moreover, MPLS is not used with all its efficiency as a traffic engineering tool since the multicast tree still constructed using the RPF tree checking without constraints.

#### B. AGGREGATED-MULTICAST

The key idea of aggregated multicast [6] is that, instead of constructing a tree for each individual multicast group in the CORE network, multiple multicast groups may share a single aggregated tree to reduce the number of multicast states and, correspondingly, tree maintenance overhead at the CORE network. In this proposal there is two requirements: (1) original group addresses of data packets must be preserved somewhere and can be recovered by exit nodes to determine how to further forward these packets; (2) some kind of identification for the aggregated tree which the group is using must be carried and transit nodes must forward packets based on that. To handle aggregated tree management and matching between multicast groups and aggregated trees, a centralized management entity called tree manager is introduced. In group to aggregated tree matching, complication arises when there is no perfect match or no existing tree covers a group (leaky matching). The disadvantage in leaky matching is that a certain amount of bandwidth is wasted to deliver data to nodes that are not involved for the group.

### III. THE MULTICAST MPLS TREE PROPOSAL

The MMT (MPLS Multicast Tree) protocol [3] constructs a multicast tree by considering only the branching node routers on this tree. By limiting the presence of multicast routing states to branching node routers, the MMT protocol converts multicast flows into multiple quasi-unicast flows. In MMT,

instead of constructing a tree for each individual multicast channel<sup>1</sup> in the CORE network, one can have several multicast channels sharing branches of their trees. The unicast LSP are used between the branching node routers of the multicast tree. By using this method, we reduce the information quantity to be memorized in routers and we ensure scalability.

#### A. MMT and other MPLS multicast proposals

In comparison with other MPLS multicast proposals, the MMT protocol has several advantages which are detailed as follows:

1- It uses a network information manager system, called hereinafter NIMS, to ensure the multicast traffic engineering in the network. It is conform with the IETF recommendations for the multicast MPLS. But the NIMS is a critical point of failure. A certain redundancy of the NIMS can ensure the survivability of the service. A certain distribution of the NIMS is possible. We do not treat it here: (1) it would unnecessarily complex our analysis;(2) ideally the distribution is independent of the multicast traffic engineering. A NIMS keeps all necessary information on LSP. All sources and all destinations of various multicast groups as well as the bandwidth associations are known. The NIMS is informed directly of any change of topology of the network (LSP or routers failure) and of any change of membership of a group destination. A tree is calculated using this NIMS and transmitted thereafter on the network.

2- It simplifies LSP setup: there is no need to create and maintain P2MP or MP2MP LSP. Instead, a tree can be broken up and its branches associated with P2P LSP. So P2P LSP are used for the transmission of multicast traffic.

3- It makes easier the aggregation of multicast flows: each branch of a multicast tree can be aggregated with other unicast traffic which shares the same ingress and egress LSR.

4- It is inter-operable with other multicast protocols: the protocol can be limited to only one domain (typically the CORE network). In other domains, traditional multicast routing protocols can be used. Once transmitted in MPLS domain, multicast packets will be forwarded on paths constructed by the MMT protocol mechanisms.

In the following Section, we present the role of the NIMS in charge to calculate the tree and to collect link state informations and group memberships besides running group to tree matching algorithm. Thereafter we present the MPLS tree construction as well as new LSP construction.

#### B. Multicast MPLS tree construction by the NIMS

In MMT, each domain contains a NIMS for each group, charged to collect *join* and *leave* messages from all group members in that domain. The NIMS is elected through a mechanism similar to the one used to elect the RP router in PIM-SM. The NIMS can be different within the same domain for each channel ( $S, G$ ). Thus, we can talk about load balancing, distribution of NIMS service and increased

<sup>1</sup>A channel is a group identified by the couple ( $S, G$ ) where  $S$  is the source address and  $G$  is the group address.

survivability of the system. After collecting all *join* messages, the NIMS computes the multicast tree for that group in the domain (It uses the Dijkstra's algorithm with constraints). The computation for a group means discovering all branching node routers for that group. The NIMS sends then *branch* messages to all branching node routers to inform them about their next hop branching node routers. On receiving this message, a branching node router creates a multicast forwarding state for the multicast channel. Once branching node routers and their next hops are identified, packets will be sent from a branching node router to another until reaching their destination.

Already established MPLS LSPs are used between multicast tree branching node routers in order to reduce forwarding states and enhance scalability. When a multicast packet arrives to the ingress router of an MPLS domain, the packet is analyzed according to its multicast IP header. The router determines who are the next hop branching node routers for that packet. Based on this information, multiple copies of the packets are generated and an MPLS label is pushed on the multicast packet according to next hop branching node router. When arriving to a next hop branching node router, the label is popped off and again the same process is repeated. This process should be repeated until the packet arrives to its destination (see Fig.1). When arriving to a LAN, the packet unlabeled can be delivered by conventional multicast protocols using IGMP messages.

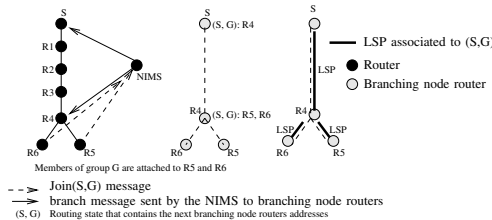


Fig. 1. Multicast MPLS tree construction.

In our approach we will use the same MPLS label for multicast traffic that follows the same path than unicast traffic. Other approaches use different labels for multicast and unicast traffic which mean the need of encoding techniques and additional overheads in routers.

*Edge Router Multicasting* is a proposal for the multicast in an MPLS network introduced in [7]. It is based on the same principles as the MMT protocol. However, ERM limits the branching node points of the multicast tree to EDGE routers of MPLS domains. Packets are sent on branches using established MPLS tunnels between the EDGE routers through the CORE routers. Consequently, as in MMT, the multicast LSP construction, the multicast flows association and the multicast traffic aggregation are transformed into simple unicast problems.

In ERM, contrary to MMT, the reservation of the bandwidth for multicast flows is not treated. Moreover, the link stress around the EDGE routers increases since the packet duplication is only allowed in the EDGE routers. The ERM characteristics make it not recommendable (as concluded in

similar approach of MPLS Multicast VPN [8]). A comparison between MMT and ERM can be found in [4].

### C. Improving MMT : the MMT2 protocol

In this section, we suppose that some routers in the network can not support mixed routing. We mean by mixed routing the coexistence of L2/L3 forwarding schemes in a router. For example, it is the case of router  $R_4$  in figure 1.

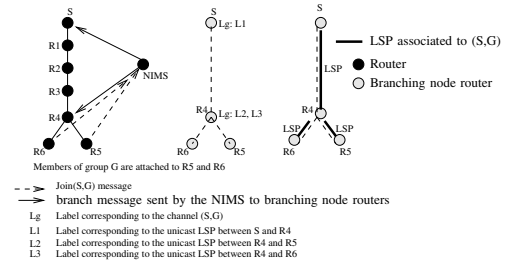


Fig. 2. Multicast MPLS tree construction with the MMT2 protocol.

We solve the mixed routing problem by using a double level of labels while preserving the MMT protocol principles of operation. The label of the lower level is a unique label representing a channel  $(S, G)$ . A label (belonging to a label interval reserved to the MMT2 protocol) is allotted to the channel  $(S, G)$  at the reception by the NIMS of the join messages for this channel. This label identifies the channel in the domain managed by the NIMS. This label could be different from one domain to another. The NIMS informs all branching node routers about this label as well as the labels corresponding to the next branching node routers for this channel. An extension of the *branch* message is necessary to carry the new information. The label corresponding to the channel  $(S, G)$  is added to the multicast packet at the domain entry, the LSR *ingress* of the domain adds also the labels of the higher level which corresponds to the next branching node routers for the channel. In intermediate routers, those who are not branching node routers, the packet is analyzed according to the entering label placed in top of the label stack, label which will be replaced by an outgoing label as in unicast MPLS. When the packet arrives to an intermediate branching node router, the label of the higher level is removed, the label identifying of the channel is treated and the new labels which corresponds to the next branching node routers are added (*cf.* figure 2). This operation is repeated until the arrival of the packet to the *egress* router. All the labels are thus popped and again the packet is sent towards the *ingress* routers of other domains or directly towards the destinations belonging to sub-networks of the *egress* routers.

### D. The MMT2 protocol and aggregated trees

Due to the limited number of labels [1], MMT2 calculates only the aggregated trees. We choose, like *Aggregated multicast*, that two channels will be associated to the same aggregated tree in a domain if the tree calculated for the first channel has exactly the same branches as the tree calculated

for the second channel in that domain. Thus, the NIMS can associate several channels to the same aggregated tree, in order to limit the use of labels in the domain and to reduce even the routing states to be stored at the branching node routers. In the next section, we evaluate the approach in term of scalability (multicast routing states reduction) and efficiency (the packet header processing time in routers and the cost of the multicast tree).

#### IV. EVALUATION OF MMT PROTOCOL

In this section, we compare MMT and its extension MMT2 (we only consider aggregated trees) with different multicast MPLS protocols, in particular PIM-MPLS [5] and *Aggregated multicast* [6]. In our simulations, PIM-MPLS refers to the simulator described in [3], [4] where PIM-SM source specific was chosen as the multicast routing protocol. We simulate the MMT protocol with NS [9] to validate the basic behavior of the approach and its efficiency to reduce the number of routing states, to decrease the packet header processing time and to lower the cost of the trees. Indeed, MMT uses on one hand the best paths tree and uses on other hand the MPLS fast label switching technique in routers. The best paths tree, calculated by the NIMS, coincides with the shortest paths tree in absence of any traffic engineering constraints.

##### A. Scalability

Since only branching node routers are considered in a multicast tree, it is obvious that our approach reduce the size of routing tables. An MPLS domain can be a transit domain for a channel where neither source nor destinations are present in the domain. A tree having one or more branching nodes in a domain is called BT (*Branched Tree*). A tree with only one path in the domain where no branching node appears in the tree is called OPT (*One Path Tree*). Table IV-A shows the average number of routing states in routers in both case : BT trees of transit with branching nodes and OPT trees of transit without branching nodes.

TABLE I  
THE AVERAGE NUMBER OF ROUTING STATES IN ROUTERS.

Tree / Protocol	BT	OPT
PIM-MPLS	$\bar{n}_T * T$	$\bar{n}_T * T$
<i>Aggregated multicast</i>	$n_{T_{agg}}^- * T_{agg}$	$n_{T_{agg}}^- * T_{agg}$
MMT	$n_{MMT} * T$	$2 * T$
MMT2	$n_{MMT-agg} * T_{agg}$	$2 * T_{agg}$

$T$  is the number of multicast trees present in the network,  $n_{MMT}$  is the average number of branching node routers on the trees by using the MMT protocol,  $n_{MMT-agg}$ <sup>2</sup> is the average number of branching node routers on the trees by using the protocol MMT2,  $\bar{n}_T$  is the average number of routers on the multicast trees by using a traditional multicast routing protocol,  $T_{agg}$  is the number of aggregated trees of *Aggregated multicast* and  $n_{T_{agg}}^-$  is the average number

<sup>2</sup>In the remainder of this evaluation, we consider that  $n_{MMT}$  and  $n_{MMT-agg}$  include the states present in the sources and the destinations.

of routers on the aggregated tree. These values satisfy the following relations:  $T \geq T_{agg}$ ;  $\bar{n}_T \geq n_{MMT}$ ,  $n_{MMT-agg}$ ;  $n_{T_{agg}}^- \geq n_{MMT-agg}$ ;  $\bar{n}_T, n_{T_{agg}}^- \geq 2$ .

It is obvious according to table IV-A that MMT presents better performances than PIM-MPLS. In the case of OPT trees, the number of routing states for MMT in the intermediate routers in the network is equal to 0 if we do not consider the routing states in the two EDGE routers source and destination in the network.

MMT has better performances compared to *aggregated multicast*. Indeed, less memory usage in the tables, thus less processing required scanning tables. In the case of BT trees, the number of routing states for the MMT protocol is not always lower than that *Aggregated multicast*. Indeed, the MMT protocol present of better performances on *Aggregated multicast* only when  $n_{MMT} * t < n_{T_{agg}}^- * T_{agg}$ . Let's take the following example: According to [6], the vBNS network is composed of 43 routers of which 16 are CORE routers. The 43 routers participate in distributing multicast traffic. In the example presented [6], a set of 2500 multicast channels are present in the network. These 2500 trees are aggregated in 1150 trees. Thus,  $n_{T_{agg}}^-$  must be larger than  $n_{MMT} * \frac{2500}{1150} \approx 2.2 * n_{MMT}$  to have MMT better than *Aggregated multicast*. As we presented in chapter 2 of [4], the number of branching node routers on a tree is very small (about 8% of the number of routers of the tree). We deduce that  $n_{MMT}$  at maximum can reach  $\approx 4$ . If the value of  $n_{T_{agg}}^-$  exceeds  $n_{MMT} * \frac{T}{T_{agg}} \approx 9$ , MMT presents then better performances. Thus, it is possible that MMT reduces more than *Aggregated multicast* the size of the multicast routing tables in the routers. Finally, according to table IV-A, MMT2 presents better performances compared to all the other protocols. To validate our evaluation, we consider 2 networks: MCI<sup>3</sup> (18 nodes in the CORE network) and Abilene (11 nodes in the CORE network) and we calculate the number of trees aggregated for 5000 trees. We consider that only one node is attached to each CORE node and this node may be either source either destination. The number of members for each group is between 2 and 10 for the Abilene network and 2 to 16 for MCI network.

Figures 3 and 4 show the average number of routing states in a router for Abilene and MCI networks. We notice that the MMT2 protocol has advantages over all the other protocols. We also notice that PIM-MPLS has the worst results. For MMT and *Aggregated multicast*, we notice that MMT has advantages over *Aggregated multicast* in MCI network but it is very bad with the Abilene network. Indeed, the Abilene network contains only 11 node: on one hand, if the number of members in a group is large, then all routers in the CORE are possible branching node routers. On the other hand, if the number of members in the groups is small, and since the network number of nodes is small the probability to have the same groups with same members is high. Then,  $\frac{T}{T_{agg}}$  ratio becomes large and thus the MMT protocol is not appropriate for this kind of topology. In all the cases, the MMT2 protocol

<sup>3</sup>Note that MCI developed the very known vBNS+ network.

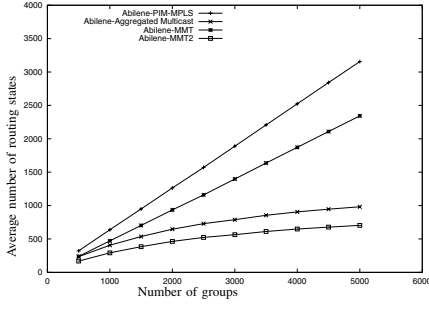


Fig. 3. Average number of routing states in a router for the Abilene network.

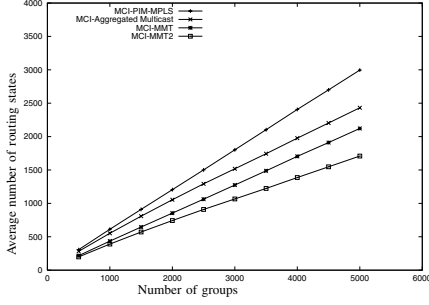


Fig. 4. Average number of routing states in a router for the MCI network.

reduces better than other protocols the size of the routing tables.

### B. Packet header processing in routers

The delay time for tree construction and the packet transmission delay are two important parameters to study. In the three protocols MMT, MMT2 and *Aggregated multicast*, a control entity receives the join messages, calculates the tree for each channel and starts association between labels and channels. Let us note that the calculation in a control entity of randomly deployed of million trees in the Abilene Network can take only a computing time of about 43 seconds. The algorithm to associate a channel to a tree was tested on Linux P4 2.4 Ghz. The tree construction delay time is thus the same for these three protocols but an advantage for MMT and MMT2 is that the LSP *unicast* are used for the multicast traffic. However, in the case of a traditional multicast routing protocol, each router between the ingress and the egress must contain a routing state for each multicast tree. Let us consider  $t_{Trait}(Ri)$ , called hereinafter  $Ri$  processing time, the time necessary to treat a packet in a router  $Ri$  and to thereafter retransmit it towards the outgoing interface. We will compare the total packet processing time  $t_{Traitg} = \sum t_{Trait}(Ri)$  for the protocols PIM-MPLS, *Aggregated multicast*, MMT and MMT2. The packet processing time,  $t_{Trait}$ , can be calculated by the following formula :  $t_{Traitg} = \sum t_{Trait}(Ri) = t_L + t_{FA} + t_R$ , where  $t_L$  is the packet transmission time on links between the source and the receiver on the multicast tree,  $t_{FA}$  is the latency of the packet in the queue of the routers, and  $t_R$  is the packet processing time in all the routers on the path from source to destination. Let us consider  $A = t_L + t_{FA}$ , a constant which does not change with the different protocols (a network with

symmetrical links).

1) *PIM-MPLS*: we note that  $t_R = \bar{n}_T * t_{MPLS\_multicast}$ , so :  $t_{Traitg} = A + \bar{n}_T * t_{MPLS\_multicast}$ , where  $\bar{n}_T$  is the average number of routers on a multicast tree and  $t_{MPLS\_multicast}$  is the average time to traverse the label table.  $t_{MPLS\_multicast}$  depends on the number of trees passing by a router.

2) *MMT*: in MMT, only branching node routers keep multicast routing states. In these routers, the MMT protocol seeks to find the corresponding *FEC* in the label table. All the other routers on the tree use the unicast MPLS routing tables to forward packets. So :  $t_{Traitg} = A + (\bar{n}_T - \bar{n}_{MMT}) * t_{MPLS\_unicast} + \bar{n}_{MMT} * (t_{MPLS\_multicast})$ .  $t_{MPLS\_unicast}$  is the average time to traverse the unicast table of labels. We have then :  $\delta_{MMT,PIM-MPLS} = \bar{n}_T * (t_{MPLS\_unicast} - t_{MPLS\_multicast}) + \bar{n}_{MMT} * (t_{MPLS\_multicast} - t_{MPLS\_unicast}) = (\bar{n}_T - \bar{n}_{MMT}) * (t_{MPLS\_unicast} - t_{MPLS\_multicast})$ . The value of  $\bar{n}_{MMT}$  is smaller in general than the value of  $\bar{n}_T$ . Moreover, with the increase in the number of channels, the value of  $t_{mpls\_multicast}$  grows too. The value of  $t_{mpls\_unicast}$  becomes smaller than the value of  $t_{mpls\_multicast}$  and  $\delta_{mmt,pim-mpls}$  takes negative values. We conclude that the MMT protocol has advantages over the PIM-MPLS protocol and all other protocols using the same type of construction of MPLS multicast trees. Note that according to [10], a Juniper T640 router can treat a package in  $10^{-9}s$  and the saving of time resulted from packets processing can be translated into a higher flow and a capacity to forward a higher number of packets.

3) *Aggregated multicast*: as in the case of PIM-MPLS we note that :  $t_R = \bar{n}_{T\_aggr} * t_{MPLS\_aggr}$ , Then :  $t_{Traitg} = A + \bar{n}_{T\_aggr} * t_{MPLS\_aggr}$ , where  $t_{MPLS\_aggr}$  is the average time to traverse the table of labels present in the routers after use of the protocol *Aggregated multicast*. We obtain as follows:  $\delta_{MMT,Aggregated\ multicast} = (\bar{n}_T - \bar{n}_{MMT}) * t_{MPLS\_unicast} + \bar{n}_{MMT} * t_{MPLS\_multicast} - \bar{n}_{T\_aggr} * t_{MPLS\_aggr}$ . It is not easy to make approximations with this formula. Indeed, searching in a routing table is not linear : it can be sometimes logarithmic curve with the use of the various techniques of searching. The  $t_{mpls\_aggr}$  value also depends on the reduction ratio of the multicast trees. Moreover, using the same reasoning with the protocol MMT2, we obtains as follows:  $\delta_{MMT2,Aggregated\ multicast} = (\bar{n}_{T\_aggr} - \bar{n}_{MMT-aggr}) * t_{MPLS\_unicast} + \bar{n}_{MMT-aggr} * t_{MPLS\_aggr} - \bar{n}_{T\_aggr} * t_{MPLS\_aggr} = (\bar{n}_{T\_aggr} - \bar{n}_{MMT-aggr}) * (t_{MPLS\_unicast} - t_{MPLS\_aggr})$ .

Since  $(\bar{n}_{T\_aggr} - \bar{n}_{MMT-aggr})$  is always  $\geq 0$  and  $(t_{mpls\_unicast} - t_{MPLS\_aggr})$  is often  $\leq 0$  (with the growth of a number of channels *multicast* in the network), we conclude that  $\delta_{mmt2,aggregated\ multicast}$  often takes negative values, which enables us to conclude that protocol MMT2 has advantages on the protocol *Aggregated multicast* in term of packet processing time.

Let us notice that it is very difficult to simulate the exact values of the total processing time of a multicast packet. Indeed, this processing depends on the size of the routing table and the technique of searching used to find an entry

in this table. Thus, we can be satisfied by making a simple comparison between PIM-SM and PIM-MPLS. Indeed, MPLS can reduce about 25%<sup>4</sup> the packet processing time and since the two protocols PIM-SM and PIM-MPLS have exactly the same number of routing states, the comparison becomes easy by supposing that the packet processing time of a unicast and multicast packet is the same<sup>5</sup>.

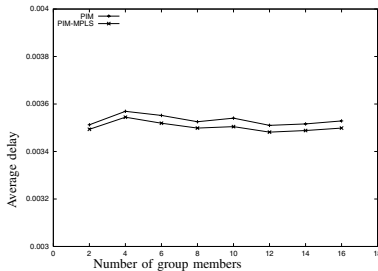


Fig. 5. The global packet average delay for the protocols PIM-SM and PIM-MPLS in the MCI network.

Figure 5 shows the average end-to-end delays of packets when the number of receivers varies from 2 to 16 for the two protocols PIM-SM and PIM-MPLS in the MCI network. We carried out 100 simulations for each value from 2 to 16. The value presented in the graphs is the average value for the 100 simulations. We deduce that using MPLS reduce the end-to-end delay time for the various destinations.

### C. Tree cost

The MMT, MMT2 protocols use the best paths tree. In the absence of constraints, the best paths tree is the shortest paths tree. *Aggregated multicast* uses the shortest paths tree while PIM-MPLS uses the reverse shortest paths tree. Note that the shortest paths tree is identical to the reverse shortest paths tree if it is considered that the network is symmetrical (it is not the case as we already saw in chapter 2 of [4]).

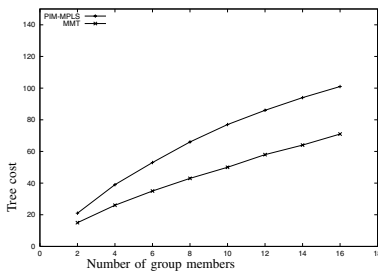


Fig. 6. The cost of the trees constructed with MMT and PIM-MPLS for the MCI network.

We consider the MCI topology and we use the following scenario: a new node is attached to each CORE node in the MCI network. A specific node is fixed as a source and a variable number of receivers is selected randomly among

<sup>4</sup>Information from JUNIPER website.

<sup>5</sup>We take the value of  $10^{-9}$ s for Juniper T640. taking an another value does not change the quality of the results.

nodes who are not CORE nodes. To the link  $\langle n1, n2 \rangle$  which connects the nodes  $n1$  and  $n2$  are associated two costs,  $n1-n2$  and  $n2-n1$  randomly chosen in the interval  $[1, 10]$ . For each number of receivers, we carried out 500 simulations by algorithm. Having identified the cost of the tree as being the average cost of the paths from the source towards all the receivers, the figure 6 shows the average cost of the tree built by the protocols MMT (the protocols MMT2 and *Aggregated multicast* offer identical results to protocol MMT) and PIM-MPLS for the MCI network. We notice that the tree cost is lower with protocol MMT than with PIM-MPLS protocol, which uses the reverse shortest paths tree.

## V. CONCLUSION

In this paper, we proposed the MMT protocol, which uses MPLS LSP between the branching node routers of a multicast tree in order to reduce routing states in intermediate routers and to increase scalability. Our approach is efficient compared to other multicast protocols and multicast MPLS proposals (PIM-MPLS, *Aggregated multicast*). Indeed, on one hand we use the best paths tree (which coincides with the shortest paths tree in absence of any traffic engineering constraints) to forward packets and on the other hand we use the fast label switching technique of MPLS in the routers. We presented the MMT2 protocol : an extension of the MMT protocol which solve the problem of the mixed routing of the "Network" layer and the "Data Link" layer in CORE routers. We evaluated MMT and MMT2 in term of scalability and efficiency. We noticed a reduction in size of the multicast routing tables compared to the other multicast MPLS approaches. We also noticed a faster packet processing time due to the use of the label switching technique of MPLS in routers. We validated the weak cost carried out by MMT compared to protocols using the reverse shortest paths tree. We conclude finally that the MMT protocol seems promising and adapted to a possible implementation of the multicast traffic engineering in the Internet.

## REFERENCES

- [1] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," IETF RFC 3031, January 2001.
- [2] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, and F. Ansari, "Overview of IP Multicast in a Multi-Protocol Label Switching MPLS Environment," IETF RFC 3353, August 2002.
- [3] A. Boudani and B. Cousin, "Using MPLS for Multicast Traffic Engineering," IRISA, Technical report 1548, July 2003.
- [4] A. Boudani, "Routage multicast : gestion des petits groupes et ingénierie de trafic," Ph.D. dissertation, Université de Rennes I, 2004.
- [5] D. Farinacci, Y. Rekhter, and E. Rosen, "Using PIM to distribute MPLS labels for multicast routes," IETF Internet draft, November 2000.
- [6] J. Cui, J. Kim, D. Maggiorini, K. Boussetta, and M. Gerla, "Aggregated Multicast: A Comparative Study," *Special Issue of Cluster Computing: The Journal of Networks, Software and Applications*, 2003.
- [7] B. Yang and P. Mohapatra, "Edge Router Multicasting With MPLS Traffic Engineering," in *IEEE International Conference on Networks (ICON'02)*, 2002.
- [8] E. Rosen, Y. Cai, D. Tappan, Y. Rekhter, and D. Farinacci, "Multicast in MPLS/BGP VPNs," IETF Internet draft, October 2003.
- [9] K. Fall and K. Varadhan, "The NS Manual," UC Berkeley, LBL/USC/ISI, and Xerox PARC, January 2001.
- [10] Juniper Networks T640 Performance Test Report, Technical Report TR0112, March 2003.