

MPLS-based Multicast Shared Trees

Ashraf Matrawy Wei Yi Ioannis Lambadaris Chung-Horng Lung
 Department of Systems and Computer Engineering
 Carleton University
 Ottawa, Ontario, Canada

Abstract—This paper presents a study of our proposed architecture for the setup of a MultiPoint-to-MultiPoint (MP2MP) Label Switched Path (LSP). This form of LSP is needed for establishing uni-directional multicast shared trees. Such trees are required for information distribution in applications such as video conferencing. The presented architecture is intended for multicast applications within a single autonomous domain and can be extended to cover inter-domain multicast sessions.

We propose the use of one (or more) control points in the network called Rendez-vous Points (RP) in a simple extension of the PIM-SM protocol to implement multicast in MPLS networks. This architecture has the advantage of using existing MPLS techniques and existing routing protocols and requires only the addition of more management capabilities at the RPs.

The experiments we carried out show that while retaining the advantages of using MPLS over traditional multicast routing, the performance of the new architecture is comparable to that of IP multicast in terms of the volume of control messages and label and memory consumption. Also the architecture scales well with the increase of the number of senders within a multicast group and with the increase of the number of multicast groups.

I. INTRODUCTION

MultiProtocol Label Switching (MPLS) is designed as a flexible traffic engineering alternative to traditional IP routing. IP routing protocols have some shortcomings [1], [2] due to the fact that IP routing algorithms in general search for the shortest path without taking overall network utilization into account. This might result in overloading some links (the shortest) while keeping others (longer paths) underutilized. In MPLS, packet headers are examined at the network's entry points (routers). Based on the packet's header and on traffic engineering settings, the packet is assigned a *label*. Once the packet passes the first router, intermediate routers will only examine the label and *switch* the packet based on this label. In addition to a more flexible traffic engineering and the faster switching at a sub-IP layer compared to routing at the IP layer, MPLS has other advantages. Please refer to [3] for a more detailed discussion of MPLS architecture.

Work on the MPLS architecture [4], [5], [6], [7] started with unicast forwarding. Since multicast communications are becoming an essential part of data networks architectures, some proposals [8], [9], [10], [11], [12] advocate providing multicast support in MPLS networks. To provide multicast communications in MPLS networks, new elements have to be added to the MPLS architecture and some modifications/extensions will be required for the existing unicast MPLS architecture [9], [13]. Some issues arise when an IP layer multicast tree (constructed using a multicast routing algorithm) is mapped directly to a layer 2 MPLS path. These issues include, the frequent

flood/prune requests of multicast users, label consumption, big state tables, and handling the co-existence of source and shared trees for the same multicast session [13].

In this paper, we present a simulation study of our proposed framework [14] to establish uni-directional multicast shared trees using MultiPoint-to-MultiPoint (MP2MP) Label Switched Path (LSP). We use Rendez-vous Points (RP) in a manner similar to PIM-SM shared trees [15]. This architecture may utilize more than one RP to implement RP failure recovery, to provide load balancing within the domain, and to enable the extension of this architecture to multiple domains by establishing LSPs between RPs in different domains. The architecture assigns labels in a way that scales very well with an increased number of senders and with an increased number of multicast groups. We provide examples of how to create and maintain the Information Bases (IB) required for this architecture. The simulation of setting up these trees in a real network topology shows that while gaining the benefits of using MPLS, setting up multicast shared trees using this architecture does not introduce performance overhead.

The rest of this paper is organized as follows. Section II presents an overview of multicast routing and discusses the need for different routing techniques to support different multicast applications. Section III defines the problems encountered in the area of multicast communications in MPLS networks. Section IV presents related work on MPLS and Point-to-MultiPoint (P2MP) multicast trees. Our architecture is discussed in Section V along with an example on how to set up MP2MP trees in Section VI. An evaluation of the scalability of this architecture is carried out using simulation experiments. The simulation results are presented in Section VII. Since the selection of the RP is crucial to this approach, we dedicate Section VIII to discuss issues related to the selection of the RP. Section IX concludes the paper with a discussion and outlines some of our future work.

II. MULTICAST ROUTING

This section briefly presents an overview of IP multicast routing algorithms [16]. Multicast routing started with a single flat routing space as opposed to the unicast hierarchical routing space. The first IP multicast on the Internet started in the early 1990s on the Multicast Backbone (MBone) [17]. MBone started with a few routers that are capable of recognizing forward multicast packets and that run multicast routing. Because most routers on the Internet were not able to do that, unicast tunnels were created between the MBone routers creating a virtual topology on top of the Internet. Routing

decisions were made using the Distance Vector Multicast Router Protocol (DVMRP) [18].

DVMRP uses the *flood-and-prune* technique. In this technique, messages from the source are sent to every router in the network. Routers will respond by sending a message back to the source on the interface the router finds the shortest back to the source. This method to construct a multicast tree is called a *Reverse Path Forwarding* tree. Leaf routers are responsible of keeping track of receivers that belong to a certain multicast session using the Internet Group Management Protocol (IGMP) [19]. When there are no receivers on a certain leaf sub-network, the corresponding router will send a *prune* message back to towards the source. The *flood-and-prune* techniques are suitable for networks that are densely populated where most routers are expected to be part of the multicast tree. They are usually referred to as Dense-mode techniques. Dense-mode protocols require the maintenance of a large amount of state information about all of the routers in the network regardless of the existence of receivers.

As the number of nodes in the Mbone grew, more routers became able to support multicast. Sparse-mode multicast routing was proposed for multicast sessions where the number of routers expected to join a session is less than the total number of routers and where they are expected to be located in more than one domain. In Sparse-mode, receivers are expected to explicitly join the multicast session by sending a join-request.

Protocol Independent Multicast (PIM) [15], with its two modes PIM-DM and PIM-SM, evolved to be a widely used multicast routing algorithm. PIM-DM constructs multicast trees in a way similar to DVMRP. PIM-SM, on the other hand, is a complex protocol that is used to construct shared multicast trees where more than one source exist on the same tree. It uses a control point called Rendez-vous Point (RP). Each source has to register and send its packets to the RP. Receivers also have to send their join-requests to the RP which forwards the sources packets to the receivers.

III. MULTICAST COMMUNICATIONS IN MPLS NETWORKS

As explained in Section II, the distribution of multicast traffic in IP networks is done using a distribution tree constructed by means of a multicast routing protocols. In order to distribute multicast traffic using MPLS networks, these multicast trees need to be mapped to layer 2 MPLS paths or Label Switched Paths (LSPs). This mapping procedure has to address some issues [13]. The most common of these issues are the following:

- Label consumption in the multicast tree case will be high. In unicast, several unicast destinations can be aggregated to one LSP. It has not been studied yet how this can be done for multicast.
- Multicast trees usually experience frequent changes due to the join/leave requests of the users. Mapping these changes to layer 2 LSPs in the case of multicast MPLS should be efficient.
- Using shared trees is more efficient in the case of multicast MPLS than using multiple source trees. The drawback of shared trees is the requirement of using MP2MP

LSPs. This requires the merging of some LSPs which may not be supported by some Layer 2 technologies such as ATM.

- Multicast shared trees can be bi-directional in some cases. While they are more efficient in terms of resource utilization, they produce more merging points in the tree compared to uni-directional trees.

Mapping the multicast tree to layer 2 and the creation of LSPs are triggered by different events. Three types of events can trigger the creation of a multicast LSP [9], [13]:

- LSP Request: either by a multicast routing protocol or by a resource reservation protocol.
- Topology changes: if the tree at layer 3 changes, the change should be mapped to layer 2.
- Traffic changes: in this case, LSPs are created only when traffic exists on this branch of the tree. This saves labels and is suitable for explicit-join multicast. This trigger is the one we adopt for our work.

In some cases, it can be beneficial to piggy-back label request on existing control messages instead of sending separate MPLS control messages. Two candidates for this are multicast routing messages [12] and RSVP-TE messages [20]. Using this mechanism in a certain network depends on a few factors such as which triggers are used in the network and whether multicast routing is available.

IV. RELATED WORK

This section provides a summary of the work that has been done so far in the area of MPLS multicast. Some of this work is done by IETF-affiliated groups to develop standards while other work is a research effort concerned with the performance of multicast architectures in MPLS networks. While RFC3353 [13] is so far the main work in this area by the IETF, it does not propose any specific architecture or protocol. Instead, it presents a framework for multicast deployment in an MPLS environment.

One group of solutions proposed to the IETF is based on piggy-backing labels on multicast routing messages. In particular, some proposals assume that the join messages of the PIM-SM multicast routing protocol will be used to send labels upstream to the multicast source [10], [12]. This method was adopted in [21], [22] to build a multicast tree for the PIM-SM (the source specific mode). In this work [21], [22], a join message from an endnode propagates upstream until it reaches a router that is actually part of the multicast tree. At that point, label assignments take place and a special database – that keeps track of mapping incoming labels to multiple outgoing labels – is created. A simulation environment to study this solution was developed in [22] which is an extension to the simulators [23], [24].

The other group of mechanisms does not rely on multicast routing. Instead they rely on extending RSVP-TE to support the establishment of P2MP LSPs [25], [26]. The work in [25] specifies mechanisms for both sender-initiated and leaf-initiated signaling. It argues in favour of the need to use RSVP-TE rather than conventional multicast routing protocols due to (1) the lack of inter-domain multicast routing in

some domains (2) the non-optimality of trees established by multicast routing (3) the fact that some multicast applications are not very dynamic, therefore pre-established sender-initiated LSPs maybe suitable for these applications. In [26], similar work is based on RSVP-TE sender-initiated LSP setup with emphasis on optimizing packet replication and minimizing the amount of maintained state information in the network core.

V. MPLS MULTI-POINT-TO-MULTI-POINT LSP SETUP

This section presents our proposed architecture for setting up the MP2MP LSP to support multicast applications that require the establishment of uni-directional shared trees. Our main design goal is to keep the setup process simple. Multicast in general introduces complexity into the network, therefore we tried to limit that complexity by using and extending existing protocols wherever that was possible. This architecture requires the existence of at least one designated control point, called Rendez-vous Point (RP) in the network domain where the tree exists. An RP will serve as a meeting point where senders send packets to be distributed using a Source Specific Multicast (SSM) tree rooted at that RP.

A. Assumptions

These assumptions should hold true in the network domain where this architecture is applied.

- We are considering the case of multi-source multi-receiver multicast. Other methods are more applicable to single-source multicast [21].
- This work is concerned with multicast within an MPLS domain. If a branching node is located at the border with a non-MPLS receiver. Both layer 2 and layer 3 forwarding mechanisms should exist at this node.
- Hop-by-hop routing is assumed. Explicit routing is not considered at this stage of our work.
- The multicast application is dynamic, frequent join/leave requests are expected.
- An FEC (and an associate LSP) will be associated with each multicast tree, i.e. with the tree D-class address.
- Labels will be distributed starting from leaf LSRs (Label Switching Routers), i.e., label distribution will be Unsolicited-Downstream.
- Label distribution will be piggybacked with the join request messages.
- Any end-node can act as a sender and a receiver at the same time. However, the example we present in Section VI does not show this case.

B. Information bases

In addition to the traditional MPLS mapping of incoming labels to output labels at each Label Switching Router (LSR), we propose the use of the following additional Information Bases (IB). A mapping of the multicast group address to the sources and their corresponding labels is the first IB we suggest. A similar entity, the LSG table (Label for Source and Group) is used in [22] for the single-source case. We modify this LSG to accommodate the multi-source case. In

TABLE I
INFORMATION BASE MAPPING GROUPS TO SOURCES

Group and Label	Source	Incoming interface
G_1, L_{G_1}	S_{11}	$I_{S_{11}}$
	\vdots	\vdots
	S_{1i}	$I_{S_{1i}}$
G_2, L_{G_2}	\vdots	\vdots
	S_{1n}	$I_{S_{1n}}$
	\vdots	\vdots
G_2, L_{G_2}	S_{21}	$I_{S_{21}}$
	\vdots	\vdots
	S_{2i}	$I_{S_{2i}}$
G_2, L_{G_2}	\vdots	\vdots
	S_{2m}	$I_{S_{2m}}$
	\vdots	\vdots

TABLE II
INFORMATION BASE MAPPING INCOMING LABELS TO OUTGOING LABELS

(Group label)	(Outgoing interface, Label)
L_{G_1}	$(O_{R_{11}}, L_{R_{11}})$
	\vdots
	$(O_{R_{1a}}, L_{R_{1a}})$
L_{G_2}	$(O_{R_{21}}, L_{R_{21}})$
	\vdots
	$(O_{R_{2b}}, L_{R_{2b}})$

our architecture, this IB resides at the RP. An example of that IB is shown in Table I where we have two multicast groups G_1 and G_2 . There are n sources subscribed to G_1 and m sources subscribed to G_2 .

To save labels, all senders who request to register with a certain group will be asked by the RP to use the same label when they sent their traffic to the RP. This will save labels and will save space in the next IB as we will explain shortly. In this example, we call the label L_{G_1} for group G_1 and L_{G_2} for group G_2 .

The second Information Base is the Label Information Base (LIB). It maps an incoming label to more than one label on different outgoing interface. An incoming label to the RP is unique per group, hence there will be as many entries per group in the LIB as the number of outgoing interfaces subscribed to this group. If we use a different label for every sender in a group, the LIB will grow linearly with the number of senders as we will have to repeat the outgoing interfaces with every sender's label. The LIB should exist on every branching node in the distribution tree rooted at the RP. However, the way this IB will be created at the RP is different from that of any other branching point due to the source merging process performed at the RP. In Table II, the mapping is shown between label L_{G_1} and a outgoing interfaces for group G_1 . Similarly, label L_{G_2} is mapped to b outgoing interfaces for group G_2 .

C. LSP establishment

The following are the details of the LSP establishment of this architecture.

- The selection of an RP is a network planning decision. In [27], some guidelines are provided on how to select

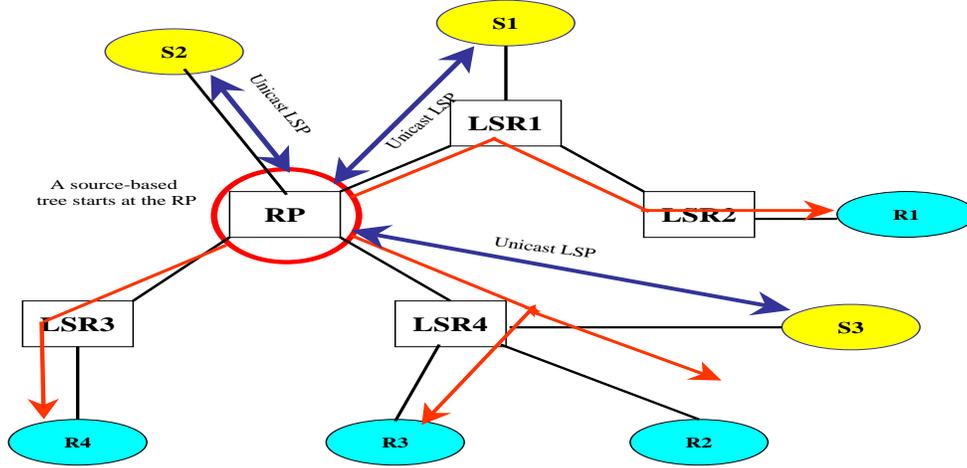


Fig. 1. Architecture for MP2MP LSP setup

RPs for PIM-SM.

- A backup RP should also be selected to support the recovery from a failure of the primary RP, to provide load-balancing capabilities, and to facilitate future extension of this architecture to support inter-domain multicast in MPLS networks. Both the primary and the backup RPs should work together to keep their Information Bases synchronized. The selection of a backup RP is beyond the scope of this paper. For the rest of this paper, when we refer to the RP, we mean the primary RP.
- Senders must register with the RP before they start to send traffic to the RP. Receivers, on the other hand, have to explicitly join the multicast session to start receiving the traffic from the RP.
- A sender registers with the RP to send its traffic to a group G . This registration will involve the creation of a unicast LSP and label binding between the sender and the RP. The RP will assign a single label, L_G , as the incoming label for this group. This label must be used by all senders that want to send traffic to this group. A record of that relates this sender, incoming interface to the RP, the group incoming label L_G , and group G will be created in the LSG table described in Table I. This label assignment saves the number of labels required and also saves the space required in the IB tables. If a different label is used per sender per group, the size of LIB table (Table II) would grow linearly with the number of senders.
- If a source tries to register with an RP in a group G where no receiver has registered, a *source-register-error* should be sent to the source from the RP after a duration of *register-time-out* seconds, releasing the reserved labels.
- Similarly, if a receiver tries to join a group where no sender is registered, a *join-request-error* should be sent to that receiver from the RP after a duration of *join-time-out* seconds, releasing the reserved labels.
- A source-based multicast tree rooted at the RP will be

built in the same way PIM-SM builds a multicast tree, i.e., based on explicit join requests from receivers.

- Requests to join group G from the end-receivers will travel upstream towards the RP with labels piggy-backed on them. A request travels upstream until it either reaches an LSR on the tree that receives packets from group G or reaches the RP.
 - 1) If the request reaches an LSR on the tree for group G , a branch from the tree starting at this LSR will be extended down towards the receiver. Label binding will be based on the label forwarded with the request message. An LSP will be setup from this LSR towards the receiver. An entry will be added to the LIB at this LSR. This is similar to the source-based tree creation proposed in [21].
 - 2) If the request reaches the RP, a record will be added to the LIB of the RP that maps the label for this group to the outgoing interface leading to this LSP (refer to Table II).
- If a receiver decides to leave the group, a *prune* message is sent towards the RP. The first LSR on the tree that receives this message, whether an LSR or the RP itself, will delete the record it has for the branch for this receiver and release the labels on the LSP downstream towards the receiver.

VI. EXAMPLE

Fig. 1 illustrates the setup of an MP2MP LSP for a single multicast group G . In the figure, five routers are represented by the boxes. Four of them are labeled LSR_1 through LSR_4 , the fifth is labeled RP . The end nodes are divided into three senders and four receivers. Three unicast LSPs are established between each of the senders S_1 , S_2 , and S_3 and the Rendezvous Point RP . A source-based tree is established with its root at router RP . Labels from the three sources are mapped to those on the outgoing branches of RP . Label mapping at

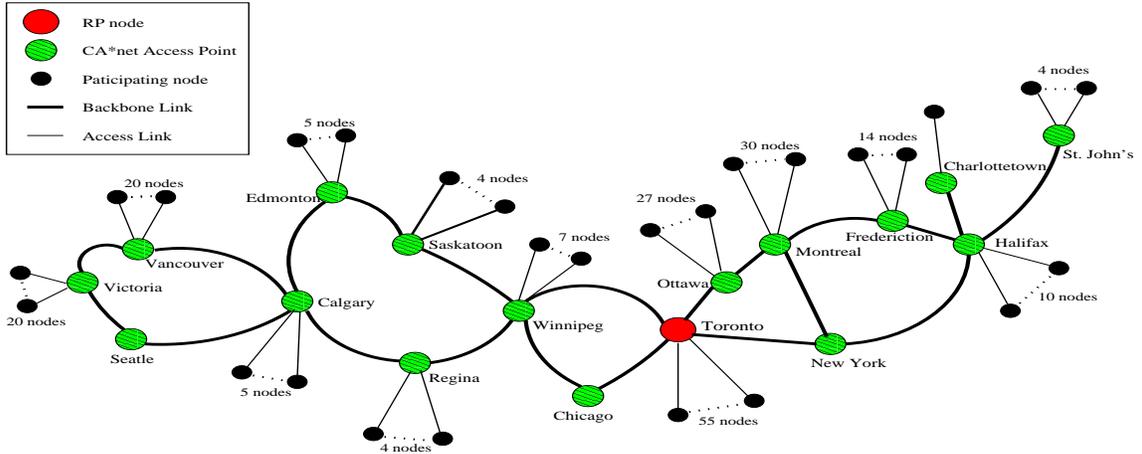


Fig. 2. The simulated topology

all LSR nodes will be done using unicast MPLS techniques except at RP .

Table III shows the mapping between the sources and group G at RP . The notation in the table is set so that I_{S_i} denotes the incoming interface at RP from S_i . Also note that all three sources will be asked by the RP to use L_G as their labels when sending their traffic to the RP . This type of table only exists at RP and a copy should be kept at the backup RP . Table IV shows the mapping of incoming labels to outgoing labels at RP . In the case other branching points exist in the tree, tables similar to Table IV should be created at these points.

TABLE III
IB MAPPING GROUPS TO SOURCES AT NODE RP

Group and Label	Source	Incoming interface
G, L_G	S_1	I_{S_1}
	S_2	I_{S_2}
	S_3	I_{S_3}

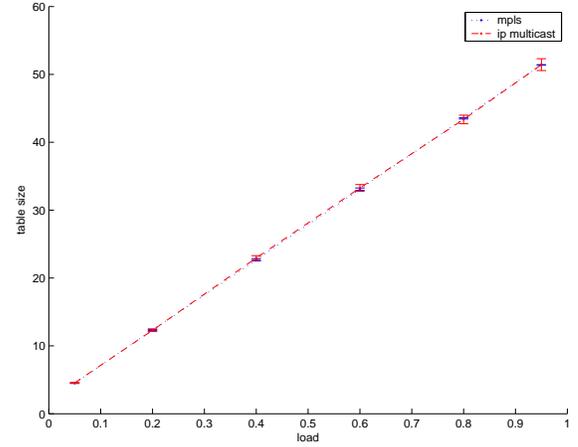
TABLE IV
IB MAPPING INCOMING LABELS TO OUTGOING LABELS AT NODE RP

Group label	(Outgoing interface, Label)
L_G	(O_{LSR_1}, L_{LSR_1})
	(O_{LSR_3}, L_{LSR_3})
	(O_{LSR_4}, L_{LSR_4})

VII. EVALUATION

In this section, we present the simulation results of the experiments we carried out to explore the performance of our proposed architecture. We extended the $ns-2$ simulator [23] by adding an implementation of our architecture to it.

The simulated topology is similar to CA*net [28] topology. As shown in the Fig. 2, the nodes with city names represent the access point in the CA*net topology. Each access point has a number of end nodes connected to it that may join the

Fig. 3. LIB table vs. Multicast routing table at the RP

multicast group. The total number of access points is 17 and the total number of end nodes is 207. We arbitrarily selected the access point named Toronto to be the RP of the multicast groups. Senders will have to register with this RP where the tree will be rooted to distribute the multicast traffic to all receivers.

At the beginning of the simulation, we choose a certain percentage of total 207 end nodes to act as the sender of the multicast group. While this percentage is fixed per simulation, the selection of which nodes to become senders is done randomly. The rest of the end nodes will act as receivers. When the simulation starts, each node will pick a random time to join the multicast group. This node will stay in the group for a period of time called On-time. After the expiration of this on-time, the node leaves the group and rejoins the group after a period of idle time called Off-time. Both the On-time and Off-time are exponentially distributed random variables with averages O and f . We control the group size of the multicast session (expressed as the percentage of all the nodes that joins the multicasession) by adjusting the average value of exponential random variables representing the On-time and

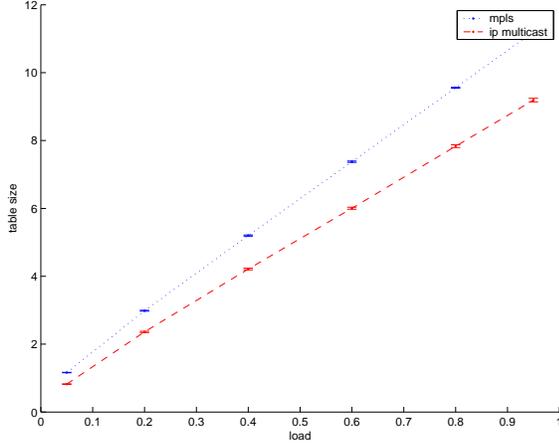


Fig. 4. LIB table vs. Multicast routing table at other access nodes

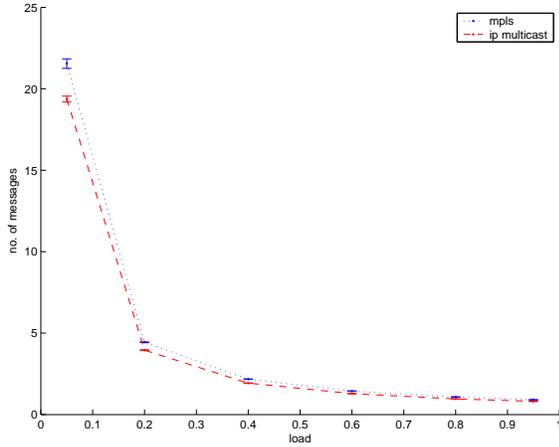


Fig. 5. Number of control messages per node per sec with different group sizes

the Off-time. In this case, the group size is expressed by

$$S = \frac{O}{O + f} * 100 \quad (1)$$

where S is the average percentage of end nodes that are in the session at any given time during the simulation. For example, if we set average value of On-time as 4 sec. and average value of Off-time as 1 sec, then the total time that each node stays in the multicast group is 80% of total simulation time. Therefore at a given time approximately 80% of all node will be in the group and we can achieve 80% group size (with respect to the total number of nodes). We also avoid synchronization in these experiments since every node is setting its own timers. For repeated each simulation 20 times with different random seeds and calculated its average value and a 95% confidence interval. The experiments we present mainly describe how the system compares to traditional multicast routing, how the increase of the number of senders as well as the increase of the number of groups affect the performance.

A. Comparison to traditional IP Multicast routing

In the first experiment, we first observed the sizes of information tables generated by our architecture. The sizes in our results are expressed in the number of *entries* in the table. We compared these values to their counterparts in the case of traditional IP Multicast routing using the same simulation scenario. In IP multicast routing, the multicast packets are encapsulated in the sender and sent directly to the RP as unicast packets. The RP will decapsulate the packet and send the packet to its destination according to its multicast routing table. We compared the size of this multicast routing table to the LIB table in our architecture both at the RP and at other nodes. The LSG table is of minimal size so we did not include it in the comparison. We also compared the number of control messages in the case of our MPLS architecture and the case IP Multicast.

We randomly picked 5% of all nodes to be the senders of the multicast group and the others become receivers. We run the simulations under receiver's participation of 5%, 20%, 40%, 60%, 80% and 95%, each for 300 sec. Fig. 3 shows the comparison between the LIB table size at the RP for our MPLS architecture and the IP Multicast routing table. Using the same scenario, we found that the difference in their values are minimal. The table size in both cases grows linearly with the increase of the group size. Fig. 4 shows the comparison of average LIB table size and the average multicast routing table size for the other nodes excluding the RP. The values in both cases are smaller than the case of the RP and the difference between MPLS and multicast routing is not significant as the figure shows.

We now look at the control messages. When a control message crosses a hop on its way to the destination, we increase the total number of messages by one. The reason is we consider the overhead of the nodes in receiving the message and forwarding it. Fig. 5 shows the average number of control message per second per active node. The difference is very small between the two cases. We found through the simulations, that the total number of messages required to establish the LSP does not vary much with the group size. This is why the average per node is high with the 5% case. Moreover, the total number of message is high with the smaller group sizes because the tree is smaller in this case. Therefore, it will take more hops for join and leave messages to reach the tree and, as mentioned, every hop counts as a new message.

B. Effect of increasing the number of senders

In the above experiment, we have 5% of the nodes acting as senders. In the following experiment, we only consider our MPLS architecture. We repeat the MPLS scenario with 1% senders, 10% senders, and with 20% senders, in addition to the 5% case. Fig. 6 shows the comparison of LIB table size in the RP. As more nodes act as sender, less will act as receiver, so LIB table size becomes slightly smaller. As we mentioned earlier in Section V-B, the RP asks the sender to assign the same incoming label to all the packets from different senders of the same group, so that the LIB table size does not grow linearly with the number of senders.

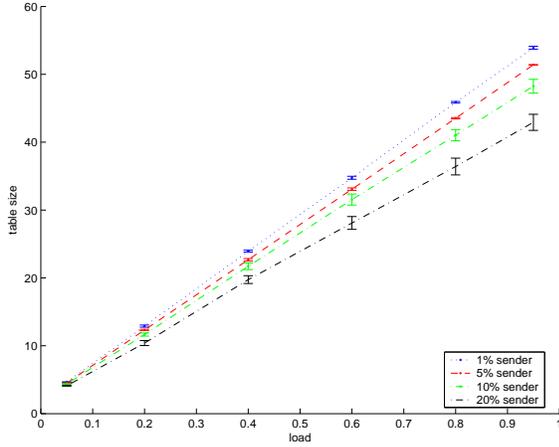


Fig. 6. Effect of increasing the number of senders on the LIB table size at the RP

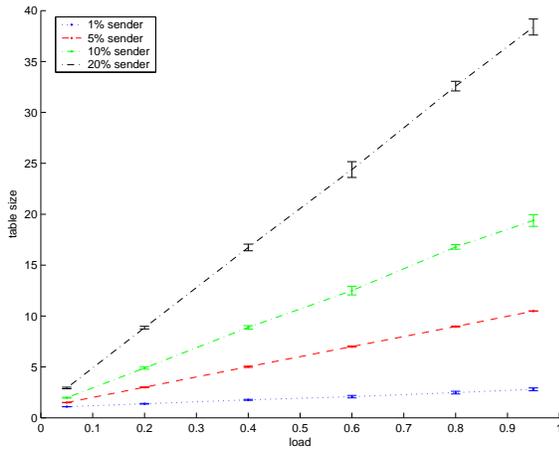


Fig. 7. Effect of increasing the number of senders on the LSG table size at the RP

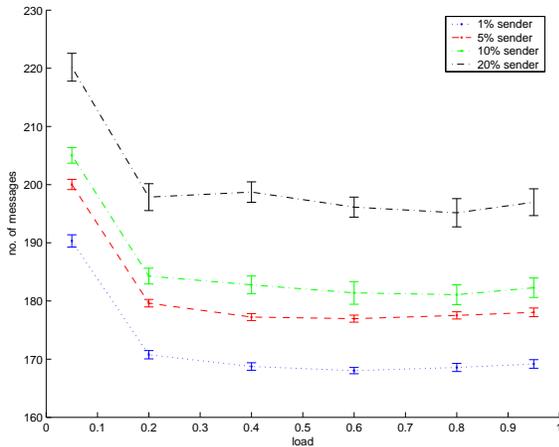


Fig. 8. Effect of increasing the number of senders on the number of control messages

Fig. 7 shows the LSG table size at the RP. It shows that the LSG table size is bigger when the number of senders increase. Fig. 8 shows the number of message per second. When the percentage of senders is higher, more messages are required. This is because when a sender decide to leave or join the group, it must propagate the request all the way along the path to the RP. On the other hand, a receiver needs only to propagate the request to one branch node. Thus a receiver needs less control messages to send a request.

C. Multiple multicast groups

In the above experiments, we only have one multicast group. Now we will look into multiple groups case in our architecture. The setting for each node to join each group is similar to the setting of one group case. Each node will join each group independently. The results for one group scenario and multiple groups scenario are compared. Fig. 9 to Fig. 11 show the results of the comparison. As shown in Fig. 9 and Fig. 10, the LIB and LSG table sizes in the RP are proportional to the number of groups. Fig. 11 shows that the multiple groups case require more control message than one group case. In the three figures, it is clear that the increase of number of groups is associated with a linear increase in the required resources, be it memory (for table sizes) or processing power for the increasing number of messages.

VIII. THE LOCATION OF THE RP

As mentioned in Section V-B, the selection of the RP is crucial to this approach. This is the case in general in schemes that depend on the selection of a central control point [27]. We observed in the simulations that the location of the RP will impact the requirements of resources at the RP. We repeated some of the simulations with other nodes acting as the RP for the group and the results differ significantly. We created two nodes at Toronto and dedicated one of them to be the RP. This resulted in much less resources requirement at the RP. The node degree at the RP (that is the number of outgoing branches from the RP) has an effect on the scalability of our architecture. Selecting one RP creates a scalability problem in case of increasing loads on this node. In case of RP collapse the operation of the entire network will be disrupted and that is of course is something to be addressed. That is why our next point of attention will be the selection of this RP and the selection of a backup RP and how to co-ordinate their operations.

IX. CONCLUSION

In this paper, we have presented an architecture for the creation of MP2MP MPLS LSPs to support multicast communications that requires uni-directional shared trees. This architecture does not require the use of multicast routing and uses most of MPLS unicast mechanisms. The simulation results show that the performance of this architecture is comparable to that of multicast routing in terms of resource requirement. Therefore, introducing it in operational networks will not require more resource while adding the benefits of using MPLS. The results also shows that it scales well with

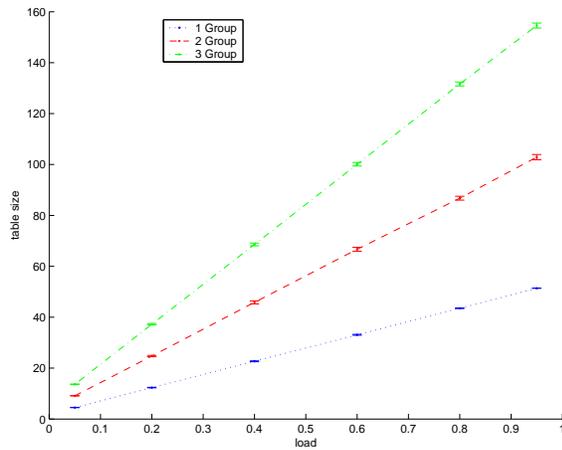


Fig. 9. Effect of increasing the number of groups on the LIB table size at the RP

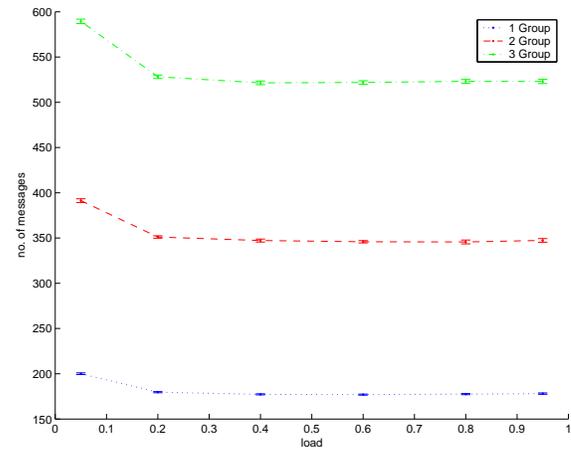


Fig. 11. Effect of increasing the number of groups on the number of control messages

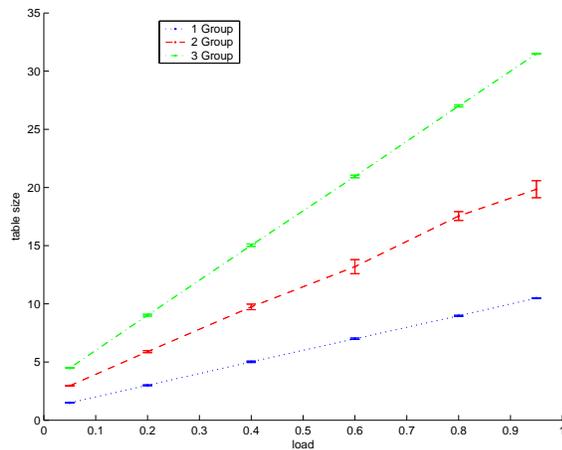


Fig. 10. Effect of increasing the number of groups on the LSG table size at the RP

increase the number of the senders as well as the number of groups.

Other issues that are in our future research plan in this area are efficient load-balancing techniques, protection and recovery, and inter-domain multicast in MPLS networks.

ACKNOWLEDGEMENT

The authors would like to thank Xiang Hu for helping with the processing the simulation results and for the generation of the graphs.

REFERENCES

- [1] D. Awduche *et al.*, "Requirements for Traffic Engineering Over MPLS," *RFC 2702*.
- [2] X. Xiao *et al.*, "Traffic Engineering with MPLS in the Internet," *IEEE Network*, March/April 2000.
- [3] E. Rosen *et al.*, "MultiProtocol Label Switching Architecture," *RFC 3031*.
- [4] "IETF MPLS Working Group," <http://www.ietf.org/html.charters/mpls-charter.html>.
- [5] "MPLS Research Center," <http://www.mplsrc.com>.
- [6] B. Davis and Y. Rekhter, *MPLS Technology and Applications*. Morgan Kaufmann, 2000.
- [7] V. Alwayn, *Advanced MPLS Design and Implementation*. Cisco Press, 2002.
- [8] A. Acharya and F. Griffoul, "IP Multicast Support in MPLS," *Proc. of IEEE ATM Workshop*, 1999.
- [9] D. Ooms and W. Livens, "IP Multicast in MPLS Networks," *Proc. of High Performance Switching and Routing*, 2000.
- [10] D. Ooms *et al.*, "MPLS for PIM-SM," *draft-ooms-mpls-pimsm-00.txt*, Work in progress.
- [11] D. Farinacci, "Partitioning Label Space among Multicast Routers on a Common Subnet," *draft-farinacci-multicast-label-part-00.txt*, Work in progress.
- [12] D. Farinacci *et al.*, "Using PIM to Distribute MPLS Labels for Multicast Routes," *draft-farinacci-mpls-multicast-02.txt*, Work in progress.
- [13] D. Ooms *et al.*, "Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment," *RFC 3353*.
- [14] A. Matrawy, C.-H. Lung, and I. Lambadaris, "A Framework for MPLS Path Setup in Uni-Directional Multicast Shared Trees," *In Proc. of SPIE Optics East, (in the Conference on Performance, QoS, and Control of Next-Generation Communication Networks)*, 2004.
- [15] S. Deering *et al.*, "PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Trans. on Networking*, April 1996.
- [16] K. C. Almeroth, "The Evolution of Multicast: From the Mbone to Inter-domain Multicast to Internet2 Deployment," *IEEE Network*, January/February 2000.
- [17] H. Eriksson, "MBONE: The Multicast Backbone," *Communications of The ACM*, August 1994.
- [18] D. Waitzman *et al.*, "Distance Vector Routing Multicast Protocol (DVMRP)," *RFC 1075*, November 1988.
- [19] W. Fenner, "Internet Group Management Protocol, Version 2," *RFC 1112*.
- [20] D. Awduche *et al.*, "RSVP-TE : Extensions to RSVP for LSP Tunnels," *RFC 3209*.
- [21] A. Boudani and B. Cousin, "A New Approach to Construct Multicast Trees in MPLS Networks," *Proc. of Seventh International Symposium on Computers and Communications*, 2002.
- [22] A. Boudani *et al.*, "Multicast Routing Simulator over MPLS Networks," *Proc. of the 36th Annual Simulation Symposium*, 2003.
- [23] S. McCanne and S. Floyd, "ns Network Simulator," <http://www.isi.edu/nsnam/ns/>.
- [24] G. Ahn and W. Chun, "Design and Implementation of mpls network simulator supporting ldp and cr-ldp," *Proc. of IEEE International Conference on Networks*, 2000.
- [25] S. Yasukawa *et al.*, "Extended RSVP-TE for Multicast LSP Tunnels," *draft-Yasukawa-mpls-rsvp-multicast-01.txt*, Work in progress.
- [26] R. Aggarwal *et al.*, "Establishing Point to Multipoint MPLS TE LSPs," *draft-raggarwa-mpls-p2mp-te-00.txt*, Work in progress.
- [27] B. Williamson, *Developing IP Multicast Networks*. Cisco Press, 1999, vol. 1.
- [28] "CANARIE Inc. - Canada's advanced Internet development organization," <http://www.canarie.ca/canet4/>.