A New Approach to Construct Multicast Trees in MPLS Networks

Ali Boudani and Bernard Cousin

IRISA/INRIA Rennes,
Campus Universitaire de Beaulieu, Avenue du Général Leclerc
35042 Rennes, France
Tel: +33 2 9984 2537, Fax: +33 2 9984 2529
{Aboudani, Bcousin}@Irisa.fr

Abstract. In this paper[1], we present a new approach to construct multicast trees in MPLS networks. This approach utilizes MPLS LSPs between multicast tree branching node routers in order to reduce forwarding states and enhance scalability. In our approach only routers that are acting as multicast tree branching node for a group need to keep forwarding state for that group. All other non-branching node routers simply forward data packets over traffic engineered unicast routes using MPLS LSPs. We can deduce that our approach can be largely deployed because it uses for multicast traffic the same unicast MPLS forwarding scheme. In this paper, we briefly discuss MPLS, the multicast scalability problem, merging the two technologies, related works and different techniques for forwarding state reduction. We evaluate the approach and present some related issues to conclude finally that it is feasible and promising.

1 Introduction

Several evolving applications like WWW, video/audio on-demand services, and teleconferencing consume a large amount of network bandwidth. Multicasting is a useful operation for supporting such applications. Using the multicast services, data can be sent from a source to several destinations by sharing the link bandwidth. But multicast suffers from the scalability problem. Indeed, a multicast router should keep forwarding state for every multicast tree passing through it. The number of forwarding states grows with the number of groups. Besides, MPLS [1] has emerged as an elegant solution to meet the bandwidth-management and service requirements for next generation Internet protocol (IP) based backbone networks.

We think that multicast and MPLS are two complementary technologies and merging these two technologies where multicast trees are constructed in MPLS networks will enhance performance and present an efficient solution for multicast scalability and control overhead problems.

In this section, we will briefly present MPLS, multicast and then the multicast scalability problem.

[1] This work has been supported by the franco-lebanese program CEDRE.

1.1 Multi-Protocol Label Switching

Multi-protocol label switching (MPLS) is a versatile solution to address the problems faced by present day networks (speed, scalability, quality-of-service (QoS) management, and traffic engineering). MPLS is an advanced forwarding scheme that extend routing with respect to packet forwarding and path controlling. An MPLS domain is a contiguous set of routers which operate MPLS routing and forwarding and which are also in one routing or administrative domain [1]. An MPLS capable router is called LSR (label switching router).

At the ingress LSR of an MPLS domain, IP packets are classified and routed based on a combination of the information carried in the IP header of these packets and local routing information maintained by the LSR. An MPLS header, called label, is then inserted for each packet. Within an MPLS domain, an LSR will use the label as the index to look up the forwarding table of the LSR. The packet is processed as specified by the forwarding table entry. The incoming label is replaced by the outgoing label, and the packet is switched to the next LSR. Before a packet leaves an MPLS domain, its MPLS header is removed. This whole process is shown in Fig.1 [2]. The paths between the ingress LSRs and egress LSRs are called label-switched paths (LSPs). MPLS uses some signaling protocol such as Resource Reservation Protocol (RSVP) [3] or Label Distribution Protocol (LDP) [4] to set up LSPs.

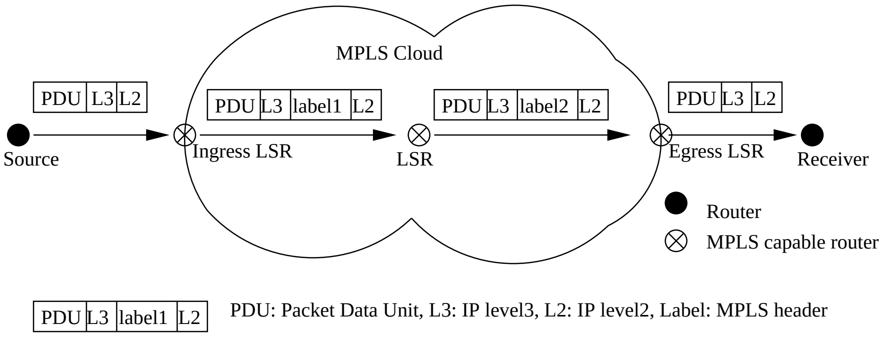

Fig. 1. MPLS forwarding scheme

1.2 Multicast

In the other hand, multicast has become increasingly important with the emergence of network-based applications such as IP telephony, video conferencing, distributed interactive simulation (DIS) and software upgrading.

Using the multicast services, a single transmission is needed for sending a packet to n destinations, while n independent transmissions would be required using the unicast services. A multicast routing protocol should be simple to implement, scalable, robust, use minimal network overhead, consume minimal memory resources, and inter-operate with other multicast routing protocols [5].

Multicast suffers from scalability problem with the number of concurrently active multicast groups because it requires a router to keep forwarding state for every multicast tree passing through it and the number of forwarding states grows with the number of groups. Scalability can be evaluated not only in terms of the overhead growth in the presence of a large number of groups but also by the number of participants per group and by groups for which the set of participants changes often over time. Overhead can be measured in terms of memory resources (in routers) as they relate to routing states maintained per group and can be measured also by bandwidth resources in terms of control or signaling messages per group and also by processing power.

1.3 Solutions for the Multicast Scalability Problem

Recently, significant research effort has focused on the multicast scalability problem. Some schemes attempt to reduce forwarding state by tunneling [6] or by forwarding state aggregation [7]. Both these works attempt to aggregate routing state after this has been allocated to groups. Other architectures aim to eliminate forwarding states at routers either completely by explicitly encoding the list of destinations in the data packets, instead of using a multicast address [8] or partially by using branching node routers in the multicast tree [9, 10].

The Xcast proposal [8], used for small groups, eliminates the need for forwarding states. The source encodes the list of destinations in the Xcast header, and then sends the packet to a router. Each router along the way parses the header, partitions the destinations based on each destinations next hop, and forwards a packet with an appropriate Xcast header to each of the next hops. In SEM [9], we proposed that the source uses unicast encoding for multicast packets and sends them to its next hop branching node routers. Each branching node router acts as a source and packets travel from a branching node router to another. A special mechanism was introduced to inform each branching node router about its next hop branching node routers for a group.

REUNITE [10], uses recursive unicast trees to implement multicast service. REUNITE does not use class D IP addresses. Instead, both group identification and data forwarding are based on unicast IP addresses. Only branching node routers for a group need to keep multicast forwarding state. All other non-branching node routers simply forward data packets by unicast routing.

We think that using the branching node routers to forward multicast data packets in unicast mode is very efficient in order to reduce forwarding state and thus enhance scalability. The main problem is how to ensure that a branching node router has a complete knowledge about its next hop branching node routers. And another issue is how can we reduce the effect of encapsulated multicast packets in unicast packets.

We think that a network information manager system (NIMS) in each domain can be used to resolve the first problem while the multi-protocol label switching (MPLS)[1] presents an efficient solution for the second problem.

The remainder of this paper is organized as follow. In Sect.2 we present some related work. In Sect.3 the concept of multicast tree construction in an MPLS network is described and some implementation related issues are discussed. Section 4 contains the approach analysis and an evaluation for its forwarding state and messaging overhead. Section 5 has a discussion for some related issues. Section 6 is a summary followed by a list of references.

2 Related Work

A framework for MPLS multicast traffic engineering proposed by Ooms et al [11] gives an overview about the application of MPLS techniques to IP multicast. Another study about MPLS and multicast proposed by Farinacci et al. [12] explain how to use PIM to distribute MPLS labels for multicast routes. A piggy-backing methodology is suggested to assign and distribute labels for multicast traffic for sparse-mode trees. Other expired Internet drafts studied the same subject [14]. The latest interesting proposal was aggregated multicast [13]. The key idea of aggregated mulicast is that, instead of constructing a tree for each individual multicast session in the core network, one can have multiple multicast sessions share a single aggregated tree to reduce multicast state and, correspondingly, tree maintenance overhead at network core. In this proposal there was 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. Complication arises when there is no perfect match or no existing tree covers a group. The disadvantage is that certain bandwidth is wasted to deliver data to nodes that are not involved for the group. Thus, bandwidth can be crucial factors for provisioning QoS in multicast networks and even for best effort Internet.

Instead of using IP encapsulation as in SEM, which of course, adds complexity and processing overhead, a potentially much better possibility is to use MPLS [1]. 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 aggregated tree covers a group. There was a disadvantage in leaky matching because certain bandwidth should be wasted to deliver data to nodes that are not involved for the group. In our proposal we intend to resolve also this problem of leaky matching.

Using MPLS with multicast has many benefits not only for reducing multicast forwarding states but also for traffic engineering and QoS issues. In this paper, we only focus on the scalability problem. We propose a novel approach that uses

MPLS LSPs between multicast tree branching node routers in order to reduce forwarding states and enhance scalability.

3 Multicast Tree Construction in an MPLS Network

The key idea of our approach is that, instead of having multicast forwarding states in all routers in a constructed tree for each individual multicast session in the core network (backbone), one can have only multicast forwarding states in branching node routers of the tree. By using the branching node routers, multicast forwarding states are reduced, correspondingly, the tree maintenance overhead at network core. This approach is targeted basically for intra-domain multicast, but it is not limited for that only since it is based on MPLS, and can be used also for inter-domain multicast.

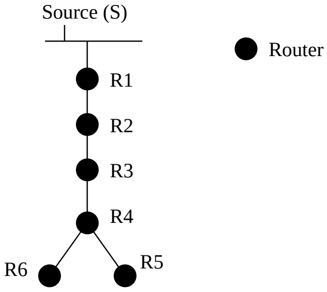

Fig. 2. Multicast tree (one source, one group)

3.1 Observation

In conventional multicast, routers on the multicast tree located between the source and a group member should maintain multicast states for the group.

Let's consider the network illustrated in Fig.2 (single multicast session with only one source and one group). Suppose that there are group members at router R5, then routers R1, R2 R3, R4 should all have multicast state for the group G even if they don't have directly connected members. Let's take now the same network with multiple multicast sessions (one single source and two groups). We suppose that there are members for group G1 at R5 and members for group G2 at R6. Routers R1, R2, R3, R4 should maintain multicast states for groups G1 and G2 even if they don't have directly connected group members. When the number of group increases in time the number of multicast states increase also.

Fig.3 represents a multicast topology (constructed tree) resulting from a traceroute experiments [10]. In the entire multicast tree there are 8 branching

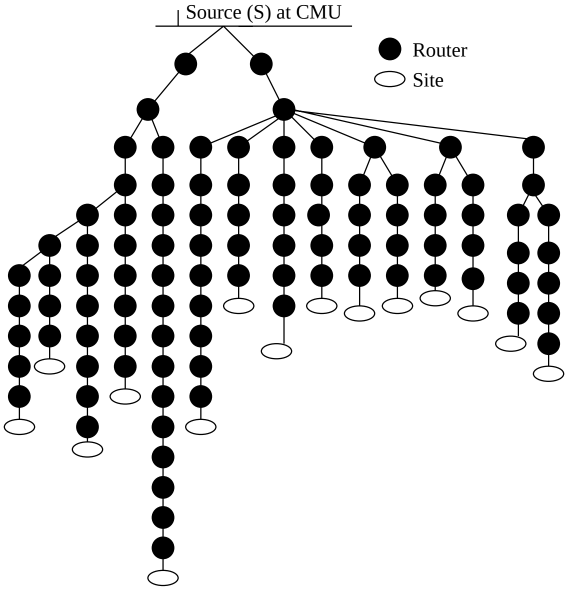

Fig. 3. Traceroutre experiments result from one source at CMU to 15 other sites

node routers out of 97 routers. We also obtained results from a set of traceroute experiments[2] from the IRISA (university of Rennes 1) to only 5 sites in France and constructed the resulting tree as shown in Fig.4. In the entire multicast tree there are 4 branching node routers out of 30 routers. One can conclude that maintaining multicast forwarding states in non-branching node routers is a waste of resources and that only branching node routers for multicast groups should maintain multicast forwarding states. In conventional multicast a router can discover that it is a branching node router for a group but it doesn't have any idea about its next hop branching node routers for that group. A second observation is that even when two multicast trees share multiple links, multicast forwarding states on routers for the shared links will not be aware of that and there is no possibility for aggregation. A third observation is that many of the LAN and WAN technologies have native support for multicast. Sending individual unicast messages to each of the receivers in a multicast-capable subnet as ethernet is very inefficient. Multicast packets should always contain the IP multicast header.

In REUNITE there was a completely unicast approach for delivering multicast packets. However,a possible proposed solution is to map a REUNITE group onto a local IP multicast group in such a network. In our proposal, Sources will allocate multicast addresses and receivers will join the multicast group by

[2] Thanks to the DESS-ISA students at the IFSIC institute for providing us with these data

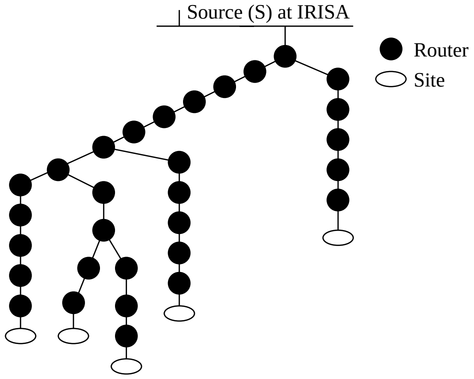

Fig. 4. Traceroutre experiments result from one source at IRISA to 5 other frensh sites

sending and (S,G) join message to the NIMS. When arriving to a LAN, the packet unlabeled can be delivered by conventional multicast protocols according to IGMP [16] informations.

In SEM, we have introduced a tunneling process which can be implemented to conventional multicast too but messages between branching node routers in order to construct tunnels can be considered as expensive overheads. We mean by overheads the header processing time that take a router to determine the routing, the size of multicast routing table overhead and the control messages overhead. All these overheads are related. Our goal is to minimize the multicast overheads in each router.

Our proposal will take all these observations into account. In our proposal, only branching node routers will contain the multicast routing table. All other routers do not need the multicast routing states.

3.2 Proposal: Multicast Tree Construction in an MPLS Network

In order to inform a branching node router about its next hop branching node routers for a group, each domain should contain a network information manager system (NIMS) for each group, charged to collect join and leave messages from all group members in that domain. After collecting all join messages, the NIMS should compute the multicast tree for that group in the domain. The computation for a group means discovering all branching node routers and the next hop branching node routers for each group. We should note that scalable multicast protocols like PIM-SM (shared tree) and CBT use a core router similar to the one used in our approach.

In our approach, we are suggesting that the NIMS should collect join messages from all group members and have a complete overview about the multicast network (Fig.5). 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 session. Once branching node routers and their next hops are identified, packets will be sent from a branching node router to another until achieving their destinations. Tunneling between branching node routers was studied in [6], [9], and [10] but was judged very expensive and very complicated since multicast packets should be encapsulated as unicast packets and then sent over the tunnel.

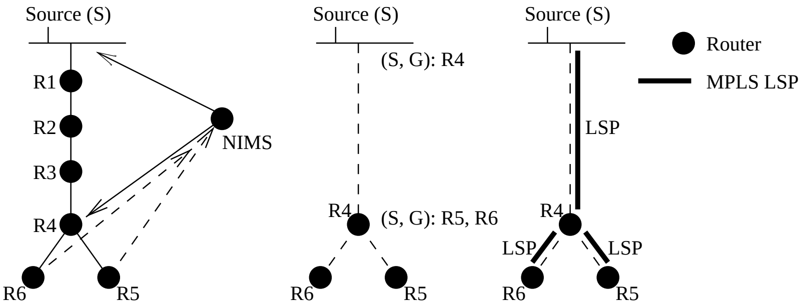

Fig. 5. The MPLS multicast tree construction

In our approach, 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 should determine who are the next hop branching node routers for that packet. Based on this information, multiple copies of the packets are generated and a MPLS label is pushed in to the multicast packet according to next hop branching node router. When arriving to a next hop branching node router, the label is pulled up and again the same process is repeated. This process should be repeated until the packet arrives to its destination (Fig.5). When arriving to a LAN, the packet unlabeled can be delivered by conventional multicast protocols according to IGMP [16] informations. When a branching node router receive the message from the NIMS it should calculate the label corresponding to the next hop router and it should add the couple (label, interface) to the (S, G) entry in the branching node router. In this way we can ensure that there is no extra overhead.

Note that labeled multicast packets will be examined at each branching router and that is why a multicast packet on non-branching router will be treated as an MPLS labeled unicast packet. A labeled multicast packet will not be different from a labeled unicast packet. So no extra charges are needed for multicast packets since the MPLS tables already exist in routers.

3.3 Comparison with Other Multicast MPLS Approaches

In comparison with other Multicast MPLS approaches presented in sect.2, the main difference is that only those routers acting as multicast tree branching node for a group need to keep forwarding state for that group. Other routers between two branching nodes don't have multicast states. Unicast is used between two branching node routers. This way the total number of multicast forwarding states may be significantly reduced. By using MPLS LSPs between branching routers, memory resource is reduced and no need for encapsulation techniques. The MPLS forwarding process conserve the router resources also.

Another difference is using NIMS to calculate the multicast tree. NIMS calculates the shortest paths from the source to all receivers. Other approaches utilize the resverse shortest paths and by consequence traffic may not follow the shortest path from the source to the receivers due to asymmetric links[17].

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

4 Evaluation

Our approach will be evaluated in terms of scalability (state information requirement and control messages overhead) and efficiency (tree cost and data processing). Multicast scalability was identified in sect.1.2. The state information requirement can be measured using the average multicast forwarding table size. State information analysis includes also the MPLS overhead. The control messages overhead can be measured using the total number of control packets sent to all the links that are needed to maintain the protocol states.

4.1 Average Multicast Forwarding Table Size

The state in a router is the forwarding state which consist of a set of (S, G) entries since packets will be sent always on shortest-path tree computed by the NIMS. A few number of BRANCH messages will be sent from the NIMS to the branching routers, thus there is no remarkable overhead.

We think that our approach has an advantage over conventional multicast protocols like PIM-SM and CBT since we don't force multicast packets to be sent all the way to the Rendez-Vous point and next to receivers. By contrast packets follow always the constraints shortest path. Besides there is no switching between shared tree and source specific tree. NIMS could be unique for each group like the case in CBT and PIM-SM.

The NIMS stores the topology of the domain and the multicast tree corresponding to each multicast group. We therefore envisage heavily loaded NIMSs to be implemented in the form of a computing cluster connected by a fault-tolerant and load-balancing middleware infrastructure. Such clusters can be scaled nearly arbitrarily, as is exemplified in real-life by web server clusters.A single NIMS may prove to be a bottleneck since all join and leave messages should be sent to it. We can imagine that there is multiple NIMSs that can communicate with each other to update informations about topology. Finally, in OSPF networks, any router can act as a NIMS since routers can easily collect topology informations.

Using the same reasoning in [6], we consider the parameter α of a distribution tree T to be the average number of multicast forwarding table entries per router for a tree:

$$\alpha(T) = \frac{Ne}{NT} \tag{1}$$

where *Ne* is sum of the total number of multicast forwarding table entries, i.e., the total number of (S, G) entries, on all the routers for distribution tree T, and *NT* is the number of routers on the tree.

When no MPLS tunnels are established, each router on a source specific distribution tree has one (S, G) forwarding table entry for the distribution tree, in which case $Ne = NT$ and the value of the α parameter reaches its maximum 1.0 for source specific trees. The minimum α value for any particular tree is defined by the following equation:

$$\alpha_{min}(T) = \frac{Nb + Nl + Nr}{NT} \tag{2}$$

where *Nb* is the number of branching points on tree T, *Nl* is the number of leaf nodes on the tree, Nr is the number of root node of the tree which always 1, and *NT* is the total number of nodes on tree T. The α parameter of a tree reaches its minimum when all uni-multicast routers on the tree are bypassed by dynamic tunnels. In conclusion, for source specific trees, the following condition holds:

$$0 < \frac{Nb + Nl + Nr}{NT} < \alpha < 1 \tag{3}$$

In Paxson's work, used in [6], routes between 37 sites located all over the world are recorded using the traceroute utility. We used the same analysis and deduced also that the minimum α parameter values are constantly smaller than 20% when using tunnels between branching node routers which implies that for global scope sparse multicast groups, over 80% reductions in forwarding table size can be achieved using our approach.

4.2 MPLS and Control Messages Overheads

In our approach, MPLS as unicast traffic engineering tool will be used also as multicast traffic engineering tool. Multicast will take all the benefits of MPLS, that already been used for unicast, and will introduce no extra overheads. A multicast packet will be treated exactly as a unicast packet in an MPLS context.

The control messages overhead can be measured in terms of average number of control messages sent per link or the total percentage of bandwidth spent on control traffic. The NIMS sends BRANCH messages to inform each branching node router about its next hop branching node routers. These BRANCH messages will be sent at each variation in the multicast topology.

In conventional multicast protocols like PIM-SSM, PIM-SM or CBT, each distribution tree needs to be refreshed periodically and that to rapidly detect a router failure (when a router that belong to the tree goes down). In our approach, when a router goes down, the unicast tables will detect that and thus no extra information is needed.

4.3 Efficiency

Efficiency can be measured in terms of tree cost and data processing. Our approach allows only the shortest path trees, which are the most efficient for data forwarding. Besides, we are using MPLS processing at routers, so our approach can be more efficient in data movement than any other scheme.

5 Discussion

5.1 MPLS Related Issues

In our approach, multicast packets will follow the same path as unicast packets. One can say that multicast packets should follow paths that differ from those that unicast packets follow, but that seems to be a heavy solution. In that case, encoding technique described in [15] will be used and that will introduce extra forwarding states in routers.

In our approach there is no extra LSP construction overhead needed. These LSPs are already constructed and used by unicast traffic. Besides, there are no modifications to LDP. Labels used for multicast packets are the same used for unicast packets and that is the major difference with all other MPLS multicast approaches. We have just to be sure that, when a packet arrive at branching node routers, the label is pulled up and the new labels (corresponding to each next hop branching node router) are pushed in.

5.2 Multicast Load Balancing

In our approach, multicast packets will be sent from a branching node router to another, but we are not obliged to follow the same path constructed by conventional multicast protocols. By using different LSPs, load balancing is ensured.

5.3 Multicast aggregation

Multicast address aggregation is important since multicast groups may share some links in their multicast trees. In conventional multicast, it is not possible to aggregate multicast IP addresses. Receivers can be located anywhere in the Internet, there is no other alternative than having one entry by multicast IP address in the multicast routing table. Since in our approach, we are using MPLS, aggregation of multicast IP addresses can be transformed to a simple aggregation of labels. There is no wasted bandwidth to realize aggregation so the aggregation overhead is zero.

5.4 Inter-Domain Multicasting

Every domain has its own NIMS for a group and all join ad leave messages from group members in that domain will be sent to that NIMS. To receive packets from sources belonging to other domains, the NIMS, after calculating which border router will transmit the packets, will sent a message to create a multicast forwarding state for the group in that border router. Due to the creation of this forwarding state, the border router will contact border routers in other domains with a normal (S, G) join message. These border routers will then contact NIMS routers in their domains. Other domains don't need to implement our approach. Once a multicast packet arrives at any border router that belong to a domain implementing the approach, a label is pushed in and sent to all receivers.

6 Conclusion and Future Works

In this paper, we presented a new approach which utilizes MPLS LSPs between multicast tree branching node routers in order to reduce forwarding states and enhance scalability. In our approach only routers that are acting as multicast tree branching node routers for a group need to keep forwarding state for that group. All other non-branching node routers simply forward data packets over traffic engineered unicast routes using MPLS LSPs. We briefly presented MPLS, multicast and then the multicast scalability problem. We presented also some related work for forwarding state reduction. We described the concept of multicast tree construction in an MPLS network and some implementation related issues are discussed. We discussed also the advantages of our approach, and we concluded after evaluating it that it is feasible and promising. We deduced also that our approach could be largely deployed because it uses for multicast traffic the same unicast MPLS forwarding scheme. Our future work will be more simulation comparison between our approach, other MPLS multicast approaches and the conventional multicast protocols.

References

1. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol label switching architecture. IETF RFC3031 (2001)
2. Xiao, X., Hannan, A., Bailey, B., Ni, L.: Traffic engineering with MPLS in The Internet. IEEE Network, March (2000)
3. Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., Swallow, G.: RSVP-TE: Extensions to RSVP for LSP tunnels. IETF Internet draft (2001)

4. Andersson, L., Doolan, P., Feldman, N., Fredette, A., Thomas, B.: LDP Specification. IETF RFC3036 (2001)
5. Ramalho, M.: Intra- and Inter-domain multicast routing protocols: A survey and taxonomy. IEEE Communications surveys and tutorials, First quarter (2000)
6. Tian, J., Neufeld, G.: Forwarding State Reduction For Sparse Mode Multicast Communication. Proceedings of IEEE INFOCOM (1998)
7. Radoslavov, P., Estrin, D., Govindan, R.: Exploiting The Bandwidth-Memory Tradeoff in Multicast State Aggregation. Technical report, USC Dept. of CS Technical Report 99-697 (1999)
8. Boivie, R., Feldman, N., Imai, Y., Livens, W., Ooms, D., Paridaens, O.: Explicit multicast (Xcast) basic specification. IETF Internet draft (2000)
9. Boudani, A., Cousin, B.: Simple explicit multicast(SEM). IETF Internet draft (2001)
10. Stoica, I., Eugene, T., Zhang, H.: "REUNITE: A recursive unicast approach to multicast". http://www.Ieee-infocom.org/2000/papers/613.ps (2000)
11. Ooms D., Livens W., Acharya A., Griffoul, F., Ansari, F.: Framework for IP multicast in MPLS. IETF Internet draft (2000)
12. Farinacci, D., Rekhter, Y., Rosen, E.: Using PIM to distribute MPLS labels for multicast routes. IETF Internet draft (2000)
13. Fei, A., Cui, J., Gerla, M., Faloutsos: Aggregated Multicast: an approach to reduce multicast state, Proceedings of third international workshop on networked group communications(NGC2001) UCL. London (2001)
14. http://ardnoc41.canet2.net/mpls/drafts/index.html.
15. Rosen, E., Tappan, D., Fedorkow, Rekhter, Y. Farinacci, D., Li, T., Conta, A.: MPLS label stack encoding. IETF RFC3032 (2001)
16. Cain, B., Deering, S., Fenner, B., Kouvelas, I., Thyagarajan, A.: Internet group management protocol, version 3. IETF Internet draft (2002)
17. Paxson, V. : End-to-End Routing Behavior in the Internet. SIGCOMM. ACM (1996)