

An Architecture for Multicast Routing Protocol Support in MPLS

Siavash Samadian-Barzoki Mozafar Bag-Mohammadi Mohammad Nikoopour Nasser Yazdani

Router Lab.,
Univ. of Tehran
s.samadian@ece.ut.ac.ir

Router Lab.,
Univ. of Tehran
mozafarb@ece.ut.ac.ir

Router Lab.,
Univ. of Tehran
mnikpur@modares.ac.ir

Router Lab.,
Univ. of Tehran
yazdani@ut.ac.ir

Abstract

MPLS (Multiprotocol Label Switching) technology is the solution to the performance requirements of the next generation IP core networks. Unfortunately, the existing architecture for MPLS ignores multicast support. Besides, the existing solutions cover a limited scope in supporting multicast services in MPLS. In this paper, we propose a general architecture to support all IP multicast routing protocols in an MPLS environment. Current multicast routing proposals for MPLS use a separate label for each branch of the multicast tree. In contrast, our architecture called GAM (General Architecture for Multicasting in MPLS) uses a unique Tree Label (*TL*) to identify a multicast tree. The architecture requires smaller MFT (Multicast Forwarding Table) sizes at each LSR as a result.

Keywords

Multicast; MPLS; Multicast in MPLS; Multiprotocol Label Switching; Multicast Routing; Architecture.

1. Introduction

The rapid growth of Internet and the extra traffic volume injected makes the packet forwarding process more challenging. MPLS is suggested to overcome the shortcomings of IP networks which perform complex layer 3 packet forwarding based on the longest prefix match. In an MPLS domain, all time-consuming tasks are pushed to the edge of the network where LERs (Label Edge Routers) are located. Ingress LERs categorize packets into different FECs (Forwarding Equivalent Classes) and assign a short fixed label to each class. Then, inside the MPLS domain, LSRs (Label Switch Routers) use these labels to switch the packets applying the label swapping scheme. It seems MPLS will be the dominant technology in the future backbone networks. However, the current architecture does not support multicast traffic services [6].

On the other hand, several evolving applications such as audio/video conferencing exist that can benefit from the multicast deployment. Using this facility, data can be sent from a source to several destinations avoiding unnecessary bandwidth consumption. Many multicast routing protocols such as PIM-SM (Protocol Independent Multicast –

Sparse Mode) [7], CM (Centralized Multicast) [8] and DVMRP (Distance Vector Multicast Routing Protocol) [9] exist in IP networks that use different tree construction methods. Several difficulties arise when applying these methods in an MPLS environment [10].

Current multicast routing proposals in MPLS use a separate label for each branch of the multicast tree. Furthermore, the labels are selected from a label space common between multicast and unicast traffics. As a result, the label assignment process to the multicast traffic is not a trivial task currently. Besides, for each multicast tree branch, an output label must be stored in MFT of an LSR which consumes invaluable memory and label.

In contrast, our architecture called GAM uses a unique Tree Label (*TL*) to identify a multicast tree, as in our previous works [1][2][3]. Therefore, we store only one entry for each multicast tree in MFT of an LSR that considerably reduces the MFT size. It is worth noting that the need to label swapping is eliminated using this scheme. Multicast senders use *TL* to send data packets to the multicast receivers. GAM consists of three different stages for the multicast data delivery in MPLS:

1. Differentiating between unicast, broadcast and multicast data packets
2. Binding label to the multicast tree
3. Building the multicast distribution tree using *TL*

The rest of this paper is organized as follows. Section 2 contains our solutions to differentiate between unicast, broadcast and multicast data packets. Then, the two label binding methods in GAM are explained in section 3. Section 4 presents the multicast tree construction methods in GAM. Next, an overview of the related work comes in section 5. Finally, we conclude the paper in section 6. We use the terms LAN and subnetwork interchangeably in this paper.

2. Differentiating Data Packets

We propose two solutions to distinguish multicast and broadcast packets from unicast ones which are described in the following subsections.

2.1. Using Layer 2 Support

As the first solution, we suggest defining a specific value in layer 2 headers to indicate whether the

label space in the MPLS header is unicast, multicast or broadcast. This value already exists in PPP (*Protocol* field type 0281 hex for MPLS unicast and type 0283 hex for MPLS multicast), Ethernet and IEEE 802.3 (ethertype value 8847 hex for MPLS unicast and value 8848 hex for MPLS multicast) [4][5]. We propose to define specific values for MPLS unicast, multicast and broadcast in other layer 2 technologies as well where not available.

Figure 1 shows the three different MPLS packet's format using this solution.

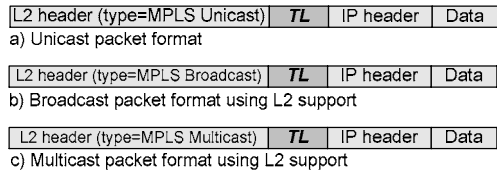


Figure 1: Packet formats in GAM

2.2. Using 2 Level Label Stack

If the first solution was not possible, then we use a two level label stack for each multicast or broadcast data packet in MPLS. Therefore, we define two globally specific reserved label values among MPLS labels (*Z* and *Y*) to sit at the top of packet's label stack, as illustrated in Fig.2. In this figure, labels *Z* and *Y* are used to identify multicast and broadcast data packets, respectively.

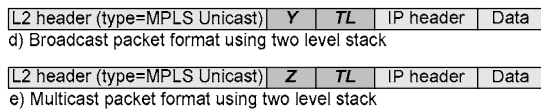


Figure 2: Packet formats in GAM

The second solution, of course, consumes more network bandwidth with 4 bytes extra label. Although we strongly recommend the first solution, the rest of the paper is based on the second solution to achieve an independent method.

3. Binding label to the multicast tree

In our architecture, a central node in the MPLS domain named TLA (Tree Label Assigner) is responsible for the multicast tree label binding. TLA assigns a unique label in the multicast label space named *TL* to each multicast distribution tree in a distributed or centralized manner. In the centralized mode, before data distribution, one of the LSRs responsible for multicast tree construction requests a label from TLA. This LSR is called MTD (Multicast Tree Director) in GAM and it can be the source of multicast data in a source specific tree [9], the central node in CM [8] or the RP (Rendezvous Point) in a shared tree [7]. When the tree construction method is distributed, TLA

broadcasts the *TL* assignment in response, using a special control message. Otherwise, the *TL* assignment is unicasted to the requesting node directly. Section 4 explains the tree construction methods in more details. At the end of multicast session, MTD sends a message asking TLA to release the label and TLA broadcasts/unicasts an appropriate message in return. The centralized label binding mode is illustrated in figure 3. In this figure, MTD₁ and MTD₂ are independently requesting a tree label. TLA assigns TL₁ and TL₂ correspondingly and broadcasts them.

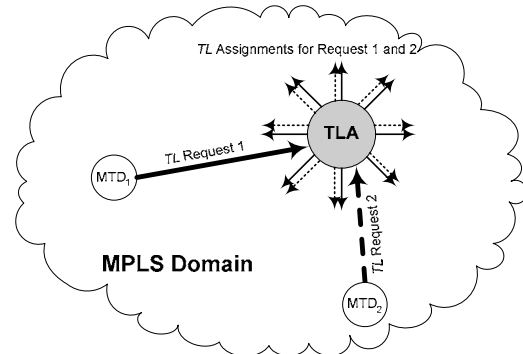


Figure 3: Centralized tree label binding mode

TLA in distributed mode, partitions the whole label space among subnetworks proportional to their size. Then, the authority of label assignment from the specified label range is given to one of the sub-network LSRs. We name this LSR as LLA (Local Label Assigner), which is selected as the querier of IGMP (Internet Group Management Protocol) [11].

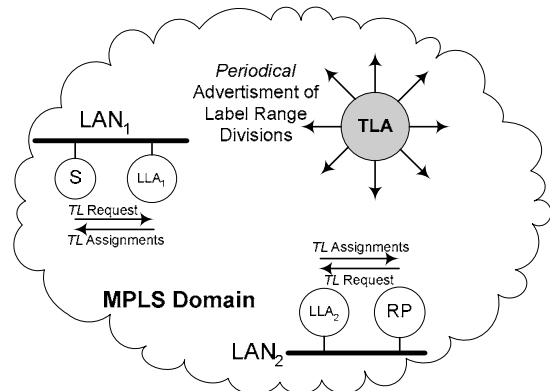


Figure 4: Distributed tree label binding mode

The distributed mode of the tree label binding is depicted in Fig.4. In this figure, to deal with the addition or deletion of subnetworks, the prefix address associated with each LAN and the corresponding assigned label range is updated periodically. Before *S* starts a multicast session in LAN₁, it requests a *TL* from LLA₁. This LLA assigns a *TL* to this session from its allocated label range and sends it in response.

4. Multicast Tree Construction

A multicast distribution tree may be constructed in two different methods:

1. **Distributed:** The distribution tree is constructed by processing the protocol messages such as join and leave messages in a distributed manner. Example protocols are PIM-SM [7], CBT [12] and DVMRP [9]. There are two kinds of distributed tree construction methods that follow.
 - a) **Flood & Prune:** To construct the tree in dense-mode protocols such as DVMRP, when a sender first starts sending, traffic is flooded out through the network. Then, the routers which have no receivers/routers downward interested in data reception, send prune messages back towards the source to stop unnecessary traffic flowing.
 - b) **Explicit Join:** Sparse-mode protocols such as PIM-SM construct the tree using explicit messages from the members of the multicast session.
2. **Centralized:** In this method, for each distribution tree, there is a central node which constructs it based on the network topology and the protocol messages such as join and leave messages collected from receivers. Example protocols are CM (Centralized Multicast) [8] and CSM (Conference Steiner Multicast) [13].

These three methods in combination with the two described methods of tree label binding create six different scenarios to be considered. The following subsections describe these scenarios.

4.1. Flood & Prune tree construction and centralized label binding

In this situation, the source requests a *TL* from TLA prior to flooding phase. The broadcasted *TL* assignment installs the appropriate entry in MFT of each LSR. Upon receiving the *TL* assignment, the source can flood its multicast data packets using the appropriate packet format from section 2. Therefore, the label switch tree is constructed directly with no change in IP multicast protocol messages. We have designed a dense-mode multicast routing protocol for this scenario in [1] and [2].

4.2. Flood & Prune tree construction and distributed label binding

When the label binding is distributed, the source requests a *TL* from its corresponding LLA locally. In addition, the *TL* assignment is sent locally to the

source. Therefore, the LSRs in MPLS domain, do not know about this assignment till they receive the first labeled multicast data packet in the flooding phase. By this time, the LSR can insert the appropriate entry in its MFT and construct the label switch tree. Thus, the protocol messages remain unchanged in this scenario as well. We have introduced a new protocol for this case in [3].

4.3. Explicit Join tree construction and centralized label binding

In this case, Join message of the first receiver triggers RP (in a shared tree) or source (in a source specific tree) to request a *TL* for the multicast tree construction. Therefore, the first receiver constructs its multicast tree branches in layer 3. The first labeled multicast data packet will map these layer 3 branches to the corresponding label switch path. The label switch tree is directly constructed for other receivers since the LSRs can use the broadcasted *TL* to process protocol messages. Hence, the label switch tree is constructed without changing the regular IP multicast protocol messages.

4.4. Explicit Join tree construction and distributed label binding

The tree is constructed in the IP layer using normal protocol messages since the LSRs do not know the *TL* used by the source or RP due to the local label selection mechanism. Upon data arrival, the LSR uses the label contained in the packet (*TL*) to map the IP layer tree to the MPLS label switch tree. Note that in this case, the layer 3 protocol messages and operations remain unchanged.

4.5. Centralized tree construction and centralized label binding

Join message of the first receiver triggers the central node, which is responsible for the tree construction, to request *TL* from TLA. TLA responds the corresponding *TL* assignment to the central node using unicast delivery. This differs from section 4.3 where the TLA response is broadcasted in the network. The central node uses *TL* to construct the MPLS label switch tree for all receivers directly.

4.6. Centralized tree construction and distributed label binding

The solution in subsection 4.4 applies to this situation as well. The tree is centrally constructed in IP layer by the central node and is transformed to the label switch tree with data arrival. Therefore, the existing IP protocol messages and operations need no change in favor of label switching.

4.7. Taxonomy

In this subsection, we summarize the characteristics of the six previous scenarios in table 1. In this table, scenarios 1 through 6 represent the cases in subsections 4.1 through 4.6 respectively. In scenarios 4 and 6, the tree is constructed in layer 3 using normal protocol messages and it is converted

to a label switch tree upon data arrival. Furthermore, the label requester node for each multicast distribution tree is RP in the shared tree and source in the source specific tree except for scenario 5 where the central node is responsible. Note that flood & prune protocols build source specific trees only.

Table 1: Taxonomy of different multicast routing scenarios in GAM

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
IP multicast routing protocol change	No	No	No	No	Yes	No
Label switch tree construction	Using the broadcasted <i>TL</i> assignment	Upon data arrival	Upon data arrival & using protocol messages directly	Upon data arrival	Using protocol messages directly	Upon data arrival
<i>TL</i> assignments delivery	Broadcasted by TLA	Locally assigned by LLA	Broadcasted by TLA	Locally assigned by LLA	Unicast by TLA	Locally assigned by LLA
Label requester	Source	Source	Shared tree: RP Source specific tree: Source	Shared tree: RP Source specific tree: Source	Central node	Shared tree: RP Source specific tree: Source

5. Related Work

The first operational prototype for label switching IP multicast consists of a Unix workstation and an ATM switch [14]. This LSR is a switch/router that is capable of forwarding multicast data using PIM-SM in IP layer and p2mp connections in ATM. The established tree in layer 3 is mapped to a p2mp tree in layer 2 in their LSR. Although they have chosen PIM-SM as multicast routing protocol in their implementation, the approach works also for PIM-DM and DVMRP.

Ooms et. al. in [15] and RFC3353 [10] present detailed framework for multicast support in MPLS. They explain many multicast related problems in MPLS and suggest solutions to some of them.

Protocols such as PIM-SM [7] and CBT [12] have explicit Join messages which could carry the label mappings. This approach is called piggy-backing method and described in [16]. Protocol messages must be changed properly in favor of MPLS. Implementation of their approach in case of dense-mode protocols like PIM-DM and DVMRP is inefficient since these protocols use no explicit messages for piggy-backing labels on them. The pros and cons of piggy-backing labels on multicast routing messages are described in [15][10].

Reference [17] suggests that labels be assigned on a per-flow (source, group) basis in a traffic-driven fashion. A traffic-driven label distribution method is introduced in [18] and a dense-mode multicast routing protocol is proposed there. In these proposals, label binding and distribution is done at each LSR which introduces extra delay in the tree construction. In addition, GAM consumes fewer labels when the label pool is common between interfaces in an LSR.

To make multicast traffic suitable for aggregation, the approach in [19] converts p2mp (point-to-

multipoint) LSP setup to multiple p2p (point-to-point) LSP problems. The protocol assumes multicast members are present only at edge routers. When the groups are dense, this method results in an inefficient usage of the network resources. The scheme also prevents end-to-end label switching of data and disturbs the unicast traffic due to layer 3 operations needed at LERs.

A new method for sparse mode multicast support is proposed in [20]. The proposed approach uses a centralized LSR named NIMS (Network Information Manager System) to calculate the multicast tree based on Join and Prune messages received from each group member.

Reference [21] proposes a simple and inefficient method to implement PIM-SM in ATM based MPLS networks.

Work in [22] addresses the required extensions to MPLS signaling protocols, RSVP-TE (Resource Reservation Protocol with Traffic Engineering extensions) and LDP (Label Distribution Protocol), to support MPLS network multicasting functionalities.

We suggested the first MPLS broadcast scheme using a central node called BLAC (Broadcast Label Assignment Center) and extended it to support dense-mode group communication in MPLS [1][2]. This proposal was a special case of subsection 4.1.

To provide scalable QoS multicast support, [23] proposes a new architecture, called AQoSM (Aggregated QoS Multicast). AQoSM can support QoS multicast scalably in DiffServ supported MPLS networks since it aggregates the groups on few trees. This aggregated approach results in some extra traffic in the network since an aggregated tree may be leaky for some groups. The reason is that the set of the group members and the tree leaves are not always identical.

6. Conclusion

In this paper, we introduce a novel architecture called GAM which provides the base for supporting multicast routing protocols in MPLS. This architecture can support both central and distributed multicast tree construction methods and protocols. The key feature in this architecture is that we use a unique label to identify each multicast distribution tree. This label may be assigned to the tree in a centralized or distributed mechanism. The LSRs in GAM no longer need label swapping for multicast data delivery in an MPLS domain. Using only one label for every branch of a multicast tree, GAM can achieve smaller MFTs.

References

1. S.Samadian-Barzoki, M.Bag Mohammadi, N.Yazdani, "A Mechanism for MPLS Broadcast and Its Extension for Multicast Dense-Mode Support in MPLS", Proc. Of ICOIN 2003, Jeju island, Korea, Feb. 2003
2. S.Samadian-Barzoki, M.Bag Mohammadi, N.Yazdani, "A New Protocol for Baroadcast in MPLS and Its Multicast Dense-Mode Extension", CSICC 2003, Mashhad, Iran, Feb. 2003 (in Persian)
3. M.Bag-Mohammadi, S.Samadian-Barzoki, M.Nikoopour, N.Yazdani, "Dense-Mode Multicast Support in MPLS: A New Approach", work in progress.
4. E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta, "MPLS Label Stack Encoding", RFC 3032, Jan. 2001
5. "Protocol Numbers and Assignment Services", <http://www.iana.org/numbers.html>
6. E.Rosen, A.Viswanathan, R.Callon, "Multiprotocol Label Switching Architecture", RFC 3031, Jan.2001
7. S.Deering, D.Estrin, D.Faranacci, V.Jacobson, C.G.Liu, L.Wei, "The PIM Architecture for Wide-Area Multicast Routing", IEEE/ACM Transactions on Networking, Vol.4, No.2, April 1996
8. S. Keshav, S. Paul, "Centralized Multicast", 7th International Conference on Network Protocols, ICNP 1999, Oct. 1999
9. D.Waitzman, C.Partridge, S.Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, Nov.1988
10. D.Ooms, B.Sales, W.Livens, A.Acharya, F.Griffoul, F.Ansari, "Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment", RFC 3353, Aug.2002
11. B. Cain, et. al., "Internet Group Management Protocol, Version 3", RFC 3376, Oct. 2002
12. A.Ballardie, "Core Based Trees (CBT Version 2) Multicast Routing - Protocol Specification", RFC 2189, Sep.1997
13. S. Aggarwal, S. Paul, D. Massey, D. Caldararu, "A flexible multicast routing protocol for group communication", Computer Networks, 2000
14. Dumortier, P., et al., "IP Multicast Shortcut over ATM: A Winner Combination", IEEE Globecom'98
15. D.Ooms, W.Livens, "IP Multicast in MPLS Networks", Proceedings of the IEEE Conference on High Performance Switching and Routing, 2000
16. D.Farinacci, Y.Rekhter, E.C.Rosen, T.Qian, "Using PIM to Distribute MPLS Labels for Multicast Routes", IETF Draft, draft-farinacci-mpls-multicast-03.txt, Nov. 2000
17. A.Acharya, F.Griffoul, F.Ansari, "IP Multicast Support in MPLS", IEEE Proceedings on ATM Workshop, 1999
18. Z. Zhang, K. Long, W. Wang, S. Cheng, "The new mechanism for MPLS supporting IP multicast", The 2000 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS 2000)
19. B. Yang and P. Mohapatra, "Edge Router Multicasting with MPLS Traffic Engineering", IEEE International Conference on Networks (ICON 2002), Aug. 2002
20. A. Boudani, B. Cousin, "A New Approach to Construct Multicast Trees in MPLS Networks", Proceedings of the Seventh International Symposium on Computers and Communications (ISCC 2002), pp. 913 - 919, July 2002
21. J. Cho, M. Y. Chung, "A Simple Method for Implementing PIM to ATM Based MPLS Networks", Proceeding of Ninth IEEE International Conference on Networks, p.p. 362-365, Oct. 2001
22. Jong-Moon Chung; Subieta Benito, M.A.; Grace Yoona Cho; Rasiah, P.; Chhabra, H.; "MPLS Multicasting Through Enhanced LDP and RSVP-TE Control", The 45th Midwest Symposium on Circuits and Systems (MWSCAS-2002), Volume: 3, p.p. 93-96, 2002
23. Jun-Hong Cui, Jinkyu Kim, Aiguo Fei, Michalis Faloutsos, Mario Gerla, "Scalable QoS Multicast Provisioning in Diff-Serv-Supported MPLS Networks", In Proceedings of IEEE Globecom2002, Taiwan, Nov. 2002