Network Working Group
Internet Draft
Expiration Date: December 1999

Liwen Wu, Pierrick Cheval
Dirk Ooms, Alex Mondrus
Alcatel

June 1999

MPLS Multicast Traffic Engineering
draft-wu-mpls-multicast-te-00.txt

Status of this Memo

Abstract

In a core ISP backbone network, when there are a lot of multicast traffic which belongs to huge number of multicast groups, providing differentiated services to different kind of multicast traffic through this core network becomes a very difficult task. Also, how to use network resources in an efficient and optimized way to support differentiated multicast service is very difficult.

This draft introduces one method, MPLS multicast traffic engineering, which can be used to manage multicast traffic in a core ISP backbone network.

This draft starts with introducing the MPLS multicast traffic engineering concept. Then it describes 2 ways of building a multicast traffic engineering tree: sender-initiated tree and receiver-initiated tree. Finally, it provides an appendix, which defines the extensions to CR-LDP to support MPLS multicast traffic engineering.

1. Introduction

If a core ISP backbone network wants to provide differentiated multicast services, the constructed multicast trees must not include highly congested nodes or links.

Similar to unicast traffic engineering[MPLS-TE], an MPLS multicast tree(or a point-to-multipoint tunnel) can be built in a network to carry multicast traffic. This tree can be administratively specified, or automatically computed by a suitable entity based on QoS and policy requirements, taking into consideration the prevailing network state.

Let's assume the following scenario. An IP network consists of four IP devices, device e1, e2, e3, e4 and some other nodes. A network administrator, for some reason, made a decision that e1 will become the root of the tree. The network administrator requests devices e2,e3, e4 to compute the reverse path(source-route) to e1. The reverse path (source-route) is calculated from the Constraint Routed(CR) topology database. Since the reverse path from e1 to e2,e3 and e4 are calculated based on the CR topology, there is much less chance that routes from e1 to e2,e3 and e4 will be congested. So, a MPLS signaling protocol using the calculated source-routing information sets up a MPLS point-to-multipoint tree from e1 to e2,e3 and e4.

When a multicast packet arrives at the root of an MPLS multicast tree, after the classification, an MPLS label is imposed to the packet. Then, at the subsequent hops, the LSR looks up the forwarding table with the incoming label, finds out all the downstream routers and corresponding outgoing labels, makes the replications, and forwards them to the downstream routers with the outgoing labels.

Since the traffic that flows along a label-switched tree is defined by the label applied at the ingress node(or root) of the MPLS tree, these trees can be regarded as tree tunnels. When an MPLS tree is used in this way we refer to it as an MPLS tree tunnel.

MPLS tree tunnels allow the implementation of a variety of policies related to network performance optimization. For example, MPLS tree tunnels can be automatically or manually routed away from network failures, congestion, and bottlenecks.

The mechanism in this draft constructs multicast trees immediately on L2. Thus the mapping of L3 trees onto L2, as described in [MPLS-MC], is not needed here.

The MPLS tree tunnel can be built as a result of a multicast SLA between a core network and its access network, or dynamic JOIN/PRUNE[PIM] from the access network. The actual mechanisms,

processes, and algorithms used to trigger and compute explicitly routed trees are beyond the scope of this specification.

The MPLS tree tunnel concept proposed in the draft applies to a core network model. It does not apply to the end-host workstation.

1.1 Dependencies on CR-IGP

In the case of automatic tree building, an IGP is needed. Both, OSPF [CR-OSPF] and IS-IS [CR-ISIS] can be used for this purpose. The LSA information is flooded throughout an AS, the point-to-multipoint tree is calculated based on the pruned CR topology.

In the case of manual tree building, it is assumed that a network administrator provisions the CR routing paths and it is assumed that the network operator knows his network topology. Based on this CR routing information the point-to-multipoint tree can be calculated. Therefore, the tree is always using the CR topology and the tree is not congested.

It is good to point out here that this document assumes that an IGP detects bottlenecks and the multicast tree is built on the pruned, excluding bottlenecks, topology.

We would like to stress the important role of IGPs for Multicast Traffic Engineering. IGPs should be able to convey specific multicast information. Since this document concentrates on Multicast Traffic engineering, new extensions for IGPs are needed, but they are out of the scope of this document.

Also, we would like to emphasize that the proposed technique is relatively simple and relies on existing routing and signalling protocols.

1.2 Interworking with other multicast routing protocols.

The method we describ in the draft only applies to a single administrative domain(AS).

We assume that the root and receivers (border routers) of the point-to-multipoint tree are running some kind of inter-domain multicast protocol, such as MBGP/BGMP or MBGP/MSDP. These inter-domain multicast protocols are used to pull the multicast traffic into the domain and also send them out to the downstream domains.

2. Sender-Initiated Multicast Traffic Engineering Tree

A sender-initiated multicast traffic engineering tree, which is one

of the applications mentioned in[EXPLICIT_TREE], is a tree built from root to all the receivers. This tree can be centrally calculated by an NMS station and sent over to the root, or it can be a tree calculated by the root of the tree.

The tree can be setup by extending CR-LDP. A new CR-LDP object, explicit tree object, can be defined to represent the whole multicast tree. The root of the tree sends a label request along with the explicit tree object. The subsequent LSR looks up its downstream routers in the explicit tree object of the label request msg. Then it sends the label request to these downstream routers. After the router receives the label mapping msgs from all of the downstream routers, it allocates a label, puts this point-to-multipoint MPLS forwarding entry into the forwarding table and sends a label mapping msg to its upstream router.

Given that this style of tree creation must carry all of the elements of the entire tree in the initial label request, and given that it is highly undesirable to fragment such requests, this style of tree building is primarily suited to trees with smaller numbers of receivers.

If a root driven tree creation is desired for large trees, a mechanism will be needed by which the tree can be established in several separate requests.

Setting up a sender-initiated tree which contains a large number of receivers and dynamic JOIN/PRUNE receivers is a subject for future study.

This tree can be torn down by the Label Release msgs sent from the root to all the receivers. When a node receives a Label Release msg, it takes the MPLS forwarding entry out of the forwarding table, and sends a Label Release msg to every downstream routers.

3. Receiver-Initiated Multicast Traffic Engineering Tree

A receiver-initiated multicast traffic engineering tree is a tree built from receivers to the root. The tree can be centrally calculated by an NMS station. The reverse path from receiver to root can also be calculated at the receiver of the tree. The reverse path of the root to each receiver is sent to the receiver if it is calculated by an NMS station. And each receiver sends a CR-LDP JOIN with the explicit reverse path and an MPLS label towards the root. At the subsequent upstream router, it merges all the CR-LDP JOIN msgs of the same tree, allocates a label, puts the point-to-multipoint label forwarding entry into the forwarding table, and sends a CR-LDP JOIN with the newly allocated MPLS label and explicit reverse path object

to the upstream router.

When a receiver wants to leave the group, it can send a Label Withdraw to its upstream router. When all the downstreams neighbors of a router leave the group, it should send a label withdraw to its upstream neighbor.

When a CR-LDP JOIN reaches a on-tree router, the router processes the CR-LDP JOIN, modifies the forwarding entry with the label assigned by the newly joined downstream router and finishes the JOIN procedure.

This method relies on CR IGP. Since this method is receiver-initiated, the point-to-multipoint tree is set up on demand.

If we compare it with the sender-initiated tree approach, the receiver-initiated method is more flexible in adding and removing LSP branches. In the case of the sender-initiated approach, the source should be able to know all receivers, so if there is requirement for building a more static tree , then the sender-initiated approach may be chosen.

4. Security Considerations

Security considerations will be addressed in a future revision of this document.

5. Acknowledgements

The authors would like to acknowledge the helpful comments and suggestions of the following people: Joel M.Halpern, Cheng-Yin Lee.

6. Authors's Address

Liwen Wu

Alcatel
44983 Knoll Square
Ashburn, VA. 20147
U.S.A
Phone: 703-724-2619
Email:liwen.wu@adn.alcatel.com

Pierrick Cheval

Alcatel
44983 Knoll Square
Ashburn, VA. 20147
U.S.A

Phone: 703-724-2080
Email:Pierrick.Cheval@adn.alcatel.com

Alex Mondrus

Alcatel
44983 Knoll Square
Ashburn, VA. 20147
U.S.A
Phone: 703-724-2749
Email:alex.mondrus@adn.alcatel.com

Dirk Ooms

Alcatel Research
Francis Wellesplein
B-2018, Antwerp
Belgium
Phone: 32-3-240-4732
Email:Dirk.Ooms@alcatel.bel

7. References

[MPLS-MC]: "Framework for IP Multicast in MPLS", D.Ooms, et.al., work in progress, Internet Draft, <draft-ooms-mpls-multicast-02.txt>

[MPLS-TE]: "Requirements for Traffic Engineering Over MPLS", Daniel O. Awduche, et.al., work in progress, Internet Draft <draft-ietf-mpls-traffic-eng-00.txt>

[EXPLCIT_TREE]: "Explicit Tree Routing", Heinrich Hummel,Swee Loke, work in progress, Internet Draft, <draft-hummel-mpls-explicit-tree-00.txt>

[CR-LDP]: "Constraint-Based LSP Set up Using LDP", Bilel Jamoussi, et.al., work in progress, Internet Draft, <draft-ietf-mpls-cr-ldp-01.txt>

[CR-OSPF]: "OSPF Extensions for Traffic Engineering", Derek M. Yeung, work in progress, Internet Draft, <draft-yeung-ospf-traffic-00.txt>

[CR-ISIS]: "IS-IS extensions for Traffic Engineering", Henk Smit, et.al., work in progress, Internet Draft, <draft-ietf-isis-traffic-00.txt>

[PIM]: "Protocol Independent Multicast-Sparse Mode (PIM-SM)", D. Farinacci, et.al., RFC2362.

Appendix A. Extensions for CR-LDP

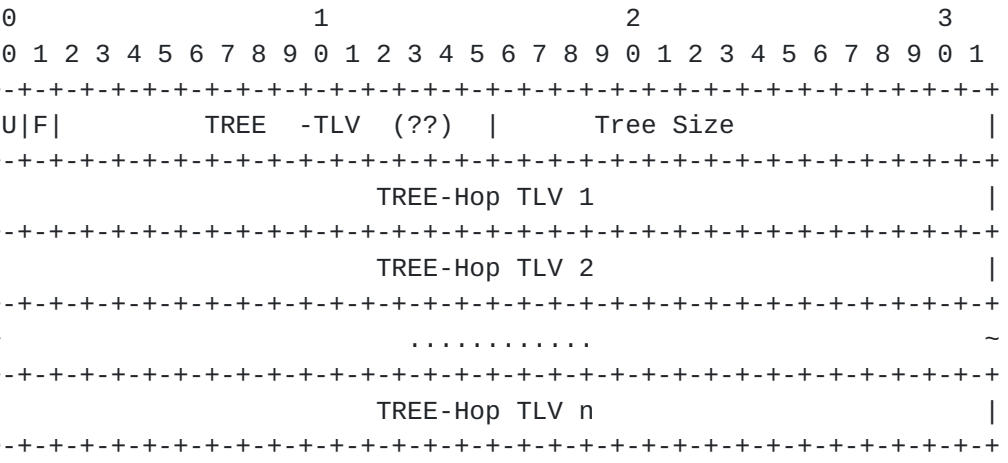
A.1 MPLS Multicast Tree ID

A MPLS multicast tree id is used to identify a network-wide unique multicast tree. The LSPID field can be used to represent the MPLS multicast tree id value.

The semantics of LSPID is specified in [CR-LDP].

A.2 Explicit Tree Object(TREE-TLV)

The TREE-TLV is an object that specifies the tree to be taken by the point-to-multipoint LSP being established. It is composed of one or more Explicit Tree Hop TLVs (TREE-Hop TLVs) defined in Section 4.2.1



U bit
  Unknown TLV bit. As defined in [LDP].

F bit
  Forward unknown TLV bit. As defined in [LDP].

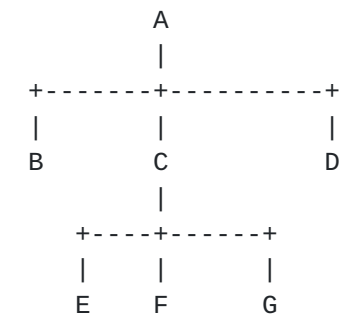
Type
  A two byte field carrying the value of the TREE-TLV type which is ??.

Tree Length
  Specifies the number of the TREE-HOP objects in the tree.

TREE-Hop TLVs
  One or more TREE-Hop TLVs defined in Section 4.2.

The TREE-HOP objects are ordered as "depth-first-order" in the msg.

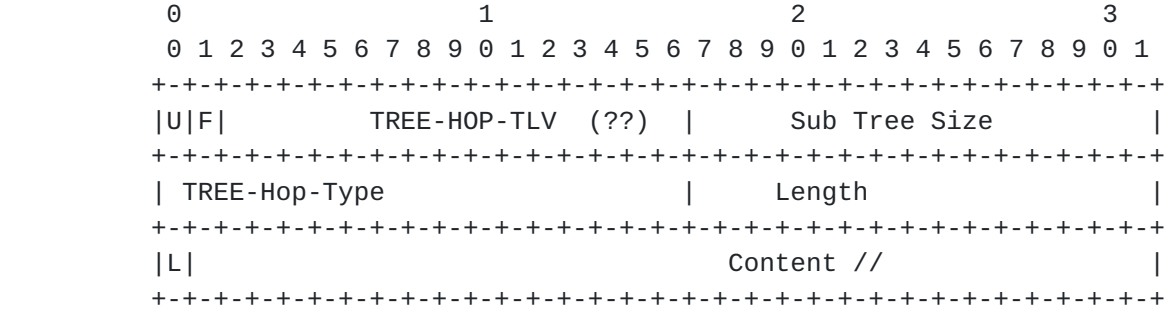
Here is one example:



The TREE-TLV are encoded as {A,B,C,E,F,G,D}

A.2.1 Tree-Hop TLV

The TREE-HOP TLV is a object that is used to represent a node that is the part of the tree.



U bit
  Unknown TLV bit. As defined in [LDP].

F bit
  Forward unknown TLV bit. As defined in [LDP].

Type
  A two byte field carrying the value of the TREE-HOP-TLV type which is ??.

Sub-Tree Size
  This field contains the number of TREE-HOP objects under this subtree.

Tree-Hop Type
  A fourteen-bit field indicating the type of contents of Tree-Hop. Currently defined values are:

Value	Type
0x801	IPv4 address
0x802	IPv6 address

Length
Specifies the length of the value field in bytes.

L bit
The L bit is an attribute of the Tree-Hop. The L bit is set if the Tree-Hop represents a loose hop in the explicit reverse route. If the bit is not set, the Tree-Hop represents a strict hop in the explicit reverse route.

The L bit in the Tree-Hop is a one-bit attribute. If the L bit is set, then the value of the attribute is "loose." Otherwise, the value of the attribute is "strict." For brevity, we say that if the value of the Tree-Hop attribute is loose then it is a "loose Tree-Hop." Otherwise, it's a "strict Tree-Hop.". Further, we say that the abstract node of a strict or loose Tree-Hop is a strict or a loose node, respectively. Loose and strict nodes are always interpreted relative to their prior abstract nodes.

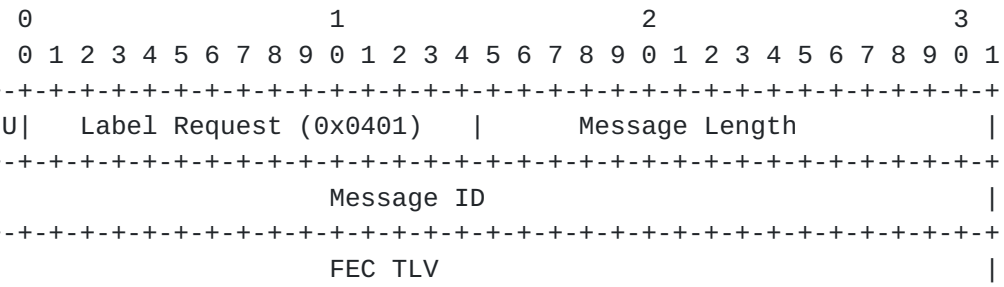
The path between a strict node and its prior node MUST include only network nodes from the strict node and its prior abstract node.

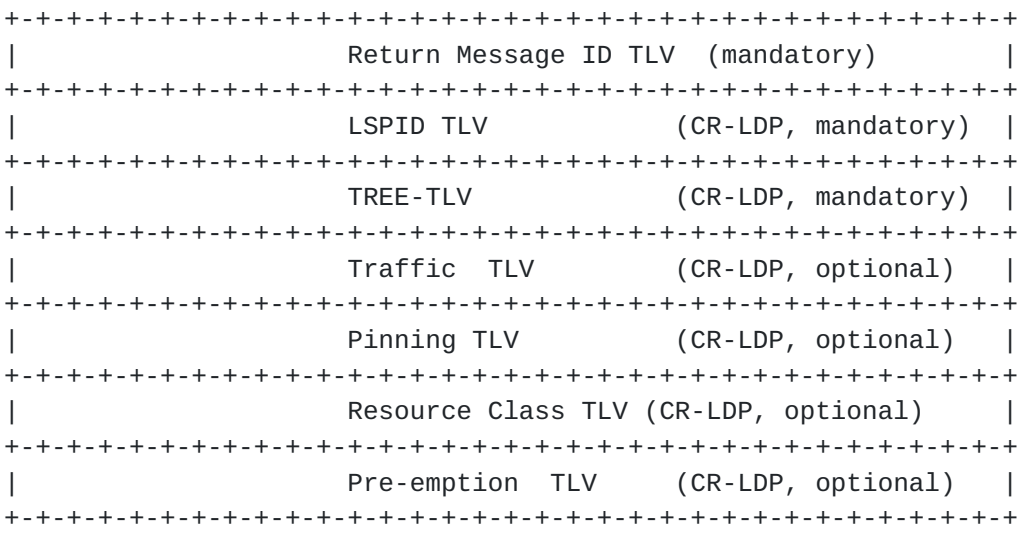
The path between a loose node and its prior node MAY include other network nodes which are not part of the strict node or its prior abstract node.

Contents
A variable length field containing the node or abstract node that is the consecutive nodes that make up the explicit routed LSP.

A.3 Label Request

The label request described in [CR-LDP] is changed to carry TREE-TLV instread of ER-TLV. So the format of label request is as follows:

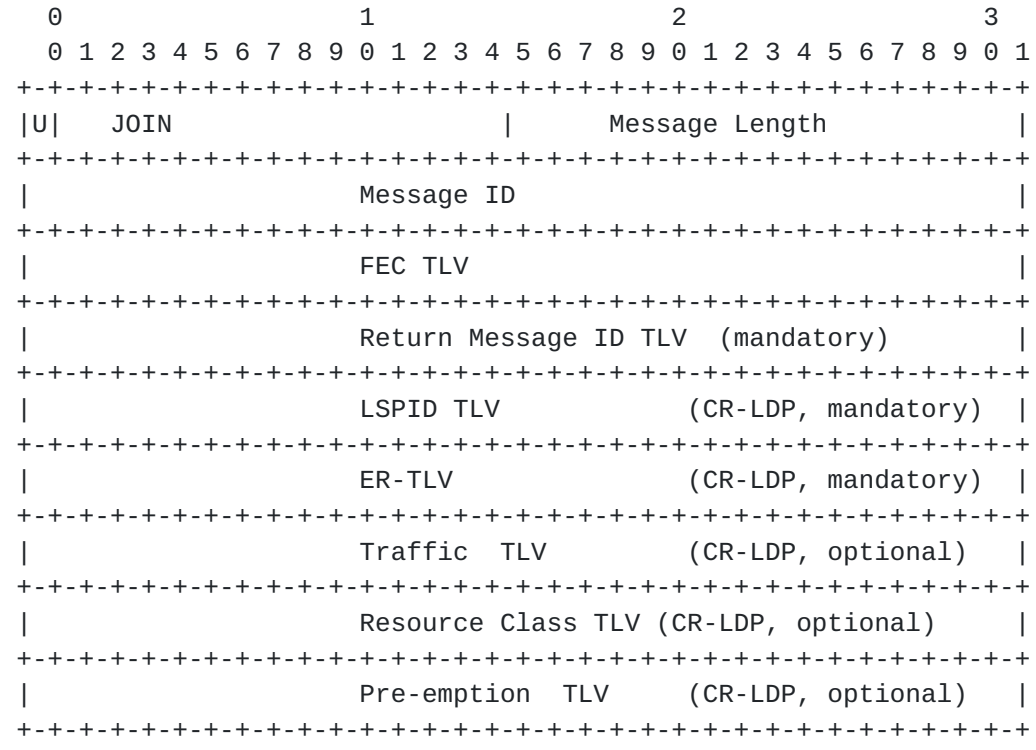




A.4 JOIN Msg

The JOIN msg is sent from downstream routers towards to root to build a tree.

The format of JOIN msg is as follows:



Each node decides the upstream root according the value specified in ER-TLV.