Network Working Group G. Swallow
Internet Draft Cisco Systems, Inc
Category: Standards Track
Expiration Date: April 2004

J. Drake
Calient Networks, Inc

H. Ishimatsu
Japan Telecom

Y. Rekhter
Juniper Networks, Inc

October 2003

GMPLS UNI: RSVP Support for the Overlay Model

draft-ietf-ccamp-gmpls-overlay-02.txt

Status of this Memo

Copyright Notice

Abstract

Generalized Multiprotocol Label Switching (GMPLS) defines both routing and signaling protocols for the creation of Label Switched

Paths (LSPs) in various transport technologies. These protocols can be used to support a number of deployment scenarios. This memo addresses the application of GMPLS to the overlay model.

Contents

1. Introduction

Generalized Multiprotocol Label Switching (GMPLS) defines both routing and signaling protocols for the creation of Label Switched Paths (LSPs) in various transport technologies. These protocols can be used to support a number of deployment scenarios. In a peer model, edge-nodes support both a routing and a signaling protocol. The protocol interactions between an edge-node and a core node are the same as between two core-nodes. In the overlay model, the core-nodes act more as a closed system. The edge nodes do not participate in the routing protocol instance that runs among the core nodes; in particular, the edge nodes are unaware of the topology of the core nodes. There may, however, be a routing protocol interaction between a core node and an edge node for the exchange of reachability information to other edge nodes.

```
     Overlay                                                  Overlay
     Network       +----------------------------------+       Network
   +---------+     |                                  |     +---------+
   |  +----+ |     |  +-----+    +-----+    +-----+   |     | +----+  |
   |  |    | |     |  |     |    |     |    |     |   |     | |    |  |
   | -+ EN +-+-----+--+ CN  +----+ CN  +----+  CN +---+-----+-+ EN +- |
   |  |    | |  +--+--|     |    |     |    |     |   |     | |    |  |
   |  +----+ |  |  |  +--+--+    +--+--+    +--+--+   |     | +----+  |
   |         |  |  |     |          |          |      |     |         |
   +---------+  |  |     |          |          |      |     +---------+
                |  |     |          |          |      |
   +---------+  |  |     |          |          |      |     +---------+
   |         |  |  |  +--+--+       |       +--+--+   |     |         |
   |  +----+ |  |  |  |     |       +-------+     |   |     | +----+  |
   |  |    +-+--+  |  | CN  +---------------+ CN  |   |     | |    |  |
   | -+ EN +-+-----+--+     |               |     +---+-----+-+ EN +- |
   |  |    | |     |  +-----+               +-----+   |     | |    |  |
   |  +----+ |     |                                  |     | +----+  |
   |         |     +----------------------------------+     |         |
   +---------+                Core Network                  +---------+
     Overlay                                                  Overlay
     Network                                                  Network

                                             Legend:   EN  -  Edge Node
                                                       CN  -  Core Node
```

Figure 1: Overlay Reference Model

Figure 1 shows a reference network. The core network is represented by the large box in the center. It contains five core-nodes marked 'CN'. The four boxes around the edge marked "Overlay Network"

represent four islands of a single network overlaid network. Only the nodes of this network with TE links into the core network are shown. These nodes are called edge-nodes; the terminology is in respect to the core network, not the overlay network. Note that each box marked "Overlay Network" could contain many other nodes. Such nodes are not shown; they do not participate in the signaling described in this document. Only he edge-nodes can signal to set up links across the core to other edge-nodes. Once a link is established, presumably the edge-node will inform the other nodes of the overlay network of its existence. How this is accomplished is beyond the scope of this document. However, one possible means is using a forwarding adjacency as described in [MPLS-HIER].

In the overlay model there may be restrictions on what may be signaled between an edge-node and a core-node. This memo addresses the application of GMPLS to the overlay model. Specifically it addresses RSVP procedures between an edge-node and a core-node in the overlay model. All RSVP procedures are assumed to be identical to [RFC3473] except as noted in this document.

This document primarily addresses interactions between an edge-node and it's adjacent (at the data plane) core-node. Except where noted, the term core-node refers to the node which is immediately adjacent to an edge-node across a particular data plane interface. The term core-nodes, however, refers to all nodes in the core.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.1. GMPLS User-to-Network Interface

One can apply the GMPLS Overlay model at the User-to-Network Interface (UNI) reference point defined in the Automatically Switched Optical Network (ASON) [G.8080]. Consider the case where the 'Core Network' in Figure 1 is a Service Provider network, and the Edge Nodes are 'user' devices. The interface between EN and CN is the UNI reference point, and to support the ASON model, one must define signaling across the UNI.

The extensions described in this memo provide mechanisms for UNI signaling that are compatible with GMPLS signaling [RFC3471, RFC3473]. Moreover, these mechanisms for UNI signaling are in line with the RSVP model, namely, there is a single end-to-end RSVP session for the user connection. The first and last hops constitute the UNI, and the RSVP session carries the user parameters end-to-end. This obviates the need to map (or carry) user parameters to (in) the

format expected by the network-to-network interface (NNI) used within the Service Provider network. This in turn means that the UNI and NNI can be independent of one another, which is a requirement of the ASON architecture. However, in the case that the UNI and NNI are both GMPLS RSVP-based, the methodology specified in this memo allows for a single RSVP session to instantiate both UNI and NNI signaling, if so desired, and if allowed by Service Provider policy.

2. Addressing

Addresses for edge-nodes in the overlay model are drawn from the same address space as the edge-nodes use to address their adjacent core-nodes. This may be the same address space as used by the core-nodes to communicate among themselves or it may be a VPN space supported by the core-nodes as an overlay.

To be more specific, an edge-node and its attached core-node must share the same address space which is used by GMPLS. A set of <edge-node, core-node> tuples share the same address space if the edge-nodes in the set could establish LSPs (through the core-nodes) among themselves (note that edge-nodes in the set may be a subset of all the edge-nodes). The address space used by the core-nodes to communicate among themselves may, but need not be shared with the address space used by any of the <edge-node, core-node> tuples.

An edge-node is identified by either a single IP address representing its Node-ID, or by one or more numbered TE links that connect the edge-node to the core-nodes. Core-nodes are assumed to be ignorant of any other addresses associated with an edge-node (i.e. addresses which are not used in signaling connections through the GMPLS core).

An edge-node need only know its own address, an address of the adjacent core-node, and know (or be able to resolve) the address of any other edge-node to which it wishes to connect.

A core-node need only know (and track) the addresses on interfaces between that core-node and its attached edge-nodes, as well as the Node IDs of those edge-nodes. In addition, a core-node needs to know the interface addresses and Node IDs of other edge-nodes to which an attached edge-node is permitted to connect.

When forming a SENDER_TEMPLATE the ingress edge-node includes either its Node-ID or the address of one of its numbered TE links. In the latter case the connection will only be made over this interface.

When forming a SESSION_OBJECT, the ingress edge-node includes either

the Node-ID of the egress edge-device or the address of one of the egress' numbered TE links. In the latter case the connection will only be made over this interface. The Extended_Tunnel_ID of the SESSION Object is set to either zero or to an address of the ingress edge-device.

Links may be either numbered or unnumbered. Further, links may be bundled or unbundled. See [GMPLS-ARCH], [RFC3471], [BUNDLE], and [RFC3477].

3. ERO Processing

An edge-node MAY include an ERO. A core-node MAY reject a Path message that contains an ERO. Such behavior is controlled by (hopefully consistent) configuration. If a core-node rejects a Path message due to the presence of an ERO it SHOULD return an PathErr message with an error code of "Unknown object class" toward the sender. This causes the path setup to fail.

Further a core-node MAY accept EROs which only include the ingress edge-node, the ingress core-node, the egress core-node and the egress edge-node. This is to support explicit label control on the edge-node interface, see below. If a core-node rejects a Path message due to the presence of an ERO not of the permitted format it SHOULD return an PathErr message with an error code of Bad Explicit Route Object as defined in [RFC3209].

3.1. Path Message without ERO

When a core-node receives a Path message from an edge-node that contains no ERO, it MUST calculate a route to the destination and include that route in a ERO, before forwarding the PATH message. One exception would be if the egress edge-node were also adjacent to this core-node. If no route can be found, the core-node SHOULD return a PathErr message with an error code and value of 24,5 - "No route available toward destination".

3.2. Path Message with ERO

When a core-node receives a Path message from an edge-node that contains an ERO, it SHOULD verify the route against its topology database before forwarding the PATH message. If the route is not viable, then a PathErr message with an error code and value of 24,5 - "No route available toward destination" should be returned.

3.3. Explicit Label Control

In order to support explicit label control and full identification of the egress link an ingress edge-node may include an ERO that consists of only the last hop. This is signaled by setting the first subobject of the ERO to the node-ID of the egress core-node with the L-bit set. Following this subobject are all other subobjects necessary to identify the link and labels as they would normally appear.

4. RRO Processing

An edge-node MAY include an RRO. A core-node MAY remove the RRO from the Path message before forwarding it. Further the core-node may remove the RRO from a Resv message before forwarding it to the edge-node. Such behavior is controlled by (hopefully consistent) configuration.

Further a core-node MAY edit the RRO in a Resv message such that it includes only the subobjects from the egress core-node through the egress edge-node. This is to allow the ingress node to be aware of the selected link and labels on at the far end of the connection.

5. Notification

An edge-node MAY include a NOTIFY_REQUEST object in both the Path and Resv messages it generates. A core-node MAY remove the NOTIFY_REQUEST object from the Path or Resv message before forwarding it. Core-nodes may send Notification messages to edge-nodes which have included the NOTIFY_REQUEST object.

Further if no NOTIFY_REQUEST object is present in the Path or Resv message (either because it was not included or because it was removed) then the core-node adjacent to the edge-node may include a NOTIFY_REQUEST object to set its value to its own address.

6. Connection Deletion

RSVP currently deletes connections using either a single pass PathTear message, or a ResvTear and PathTear message combination. Upon receipt of the PathTear message, a node deletes the connection state and forwards the message. In optical networks, however, it is possible that the deletion of a connection (e.g., removal of the cross-connect) in a node may cause the connection to be perceived as failed in downstream nodes (e.g., loss of frame, loss of light,

etc.). This may in turn lead to management alarms and perhaps the triggering of restoration/protection for the connection.

To address this issue, the graceful connection deletion procedure SHOULD be followed. Under this procedure, an ADMIN_STATUS object MUST be sent in Path or Resv message along the connection's path to inform all nodes enroute of the intended deletion, prior to the actual deletion of the connection. The procedure is described in [RFC3473].

7. VPN Connections

As stated in the addressing section above, the extensions in this document are designed to be compatible with the support of VPNs. Since the core network may be some technology other than GMPLS, no mandatory means of mapping core connections to access connections is specified. However, when GMPLS is used for the core network, we RECOMMEND the following procedure which is based on [LSP-HIER].

The VPN connection is modeled as being three hops. One for each access link and one hop across the core network.

The VPN connection is established using a two step procedure. When a Path message is received at a core node on an interface which is part of a VPN, the Path message is held until a core connection is established.

The connection across the core is setup as a separate signaling exchange between the core nodes, using the address space of the core nodes. While this exchange is in progress, the original Path message is held at the ingress core node. Once the exchange for the core connection is complete, this connection is used in the VPN connection as if it were a single link. This is signaled by including an IF_ID RSVP_HOP object (defined in [RFC3473]) using the procedures defined in [LSP-HIER].

The original Path message is then forwarded within the VPN addressing realm to the core node attached to the destination edge node. Many ways of accomplishing this are available, including IP and GRE tunnels and BGP/MPLS VPNs. Specifying a particular means is beyond the scope of this document.

8. Security Considerations

This draft imposes no new security risks over [RFC3473]. In fact it represents a lower trust model between the core and edge-nodes as the core is permitted to hide its topology from the edge-nodes. The core is also permitted to restrict the actions of edge-nodes by filtering out specific RSVP objects.

9. Acknowledgments

The authors would like to thank Kireeti Kompella, Jonathan Lang, Dimitri Papadimitriou, Dimitrios Pendarakis and Bala Rajagopalan for their comments and input.

10. References

10.1. Normative References

[RFC3471] Berger, L., "Generalized MPLS Signaling Functional Description", RFC 3471, January 2003.

[RFC3473] Berger, L., "Generalized MPLS Signaling RSVP-TE Extensions", RFC 3473, January 2003.

[RFC3209] Awduche, et al, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

10.2. Informative References

[GMPLS-ARCH] Mannie, E, et al., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", Internet Draft, draft-ietf-ccamp-gmpls-architecture-04.txt, Feb., 2003.

[MPLS-HIER] Kompella, K., and Rekhter, Y., "LSP Hierarchy with MPLS TE", Internet Draft, draft-ietf-mpls-lsp-hierarchy-08.txt, Sep., 2001.

[BUNDLE] Kompella, K., Rekhter, Y., and Berger, L., "Link Bundling in MPLS Traffic Engineering", Internet Draft, draft-ietf-mpls-bundle-04.txt, July, 2002.

[RFC3477] Kompella, K., and Rekhter, Y., "Signaling Unnumbered Links in RSVP-TE", RFC 3477, January 2003.

[G.8080] ITU-T Rec. G.8080/Y.1304, "Architecture for the Automatically Switched Optical Network (ASON)," November 2001 (and Revision, January 2003).

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119.

11. Authors' Addresses

John Drake
Calient Networks
5853 Rue Ferrari
San Jose, CA 95138
Phone: +1 408 972 3720
Email: jdrake@calient.net

Hirokazu Ishimatsu
Japan Telecom Co., Ltd.
2-9-1 Hatchobori, Chuo-ku, Tokyo, 104-0032, Japan
Tel: +81 3 5540 8493
Email: hirokazu.ishimatsu@japan-telecom.co.jp

Yakov Rekhter
Juniper Networks, Inc.
Email: yakov@juniper.net

George Swallow
Cisco Systems, Inc.
1414 Massachusetts Ave,
Boxborough, MA 01719
Phone: +1 978 936 1398
Email: swallow@cisco.com

12. Full Copyright Statement