Network Working Group
Request for Comments: 4208
Category: Standards Track

G. Swallow
Cisco Systems, Inc
J. Drake
Boeing
H. Ishimatsu
G1M Co., Ltd.
Y. Rekhter
Juniper Networks, Inc

October 2005

Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

Generalized Multiprotocol Label Switching (GMPLS) defines both routing and signaling protocols for the creation of Label Switched Paths (LSPs) in various switching technologies. These protocols can be used to support a number of deployment scenarios. This memo addresses the application of GMPLS to the overlay model.

Table of 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 overlay 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 directly in the signaling described in this document. Only the edge-nodes can signal to set up links across the core to other edge-nodes.

How a link between edge-nodes is requested and triggered is out of the scope of this document, as is precisely how that link is used by the Overlay Network. One possibility is that the edge-nodes will inform the other nodes of the overlay network of the existence of the link, possibly using a forwarding adjacency as described in [MPLS-HIER]. Note that this contrasts with a forwarding adjacency that is provided by the core network as a link between core-nodes.

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-TE procedures between an edge-node and a core-node in the overlay model. All signaling procedures are identical to the GMPLS extensions specified in [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; out-of-band and non-adjacent signaling capabilities may mean that signaling messages are delivered on a longer path. Except where noted, the term core-node refers to the node immediately adjacent to an edge-node across a particular data plane interface. The term core-nodes, however, refers to all nodes in the core.

Realization of a single or multiple instance of the UNI is implementation dependent at both the CN and EN so long as it meets the functional requirements for robustness, security, and privacy detailed in Section 7.

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].

Readers are assumed to be familiar with the terminology introduced in [RFC3031], [GMPLS-ARCH], and [RFC3471].

1.1. GMPLS User-Network Interface (GMPLS UNI)

One can apply the GMPLS Overlay model at the User-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 an EN and a 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 that is used by GMPLS to signal between the edge-nodes across the core network. 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 without address mapping or translation (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. This does not imply a mandatory 1:1 mapping between a set of LSPs and a given addressing space.

When multiple overlay networks are supported by a single core network, one or more address spaces may be used according to privacy requirements. This may be achieved without varying the core-node addresses since it is the <edge-node, core-node> tuple that constitutes address space membership.

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 that 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, as well as (of course) the addresses used in the overlay network island of which it is a part.

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 a PathErr message with an error code of "Unknown object class" toward the sender as described in [RFC3209]. This causes the path setup to fail.

Further, a core-node MAY accept EROs that 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 that is not of the permitted format, it SHOULD return a 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 an 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 (according to topology, currently available resources, or local policy), 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 this information in the ERO that it passes to its neighboring core-node. In the case that no other ERO is supplied, this explicit control information is provided as the only hop of the ERO and is encoded 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.

The same rules apply to the presence of the explicit control subobjects as the last hop in the ERO, if a fuller ERO is supplied by the ingress edge-node to its neighbor core-node; but in this case the L-bit MAY be clear.

This process is described in [RFC3473] and [EXPLICIT].

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. Core-nodes may send Notify messages to edge-nodes that have included the NOTIFY_REQUEST object.

A core-node MAY remove a NOTIFY_REQUEST object from a Path or Resv message received from an edge-node before forwarding it.

If no NOTIFY_REQUEST object is present in the Path or Resv received from an edge-node, the core-node adjacent to the edge-node may include a NOTIFY_REQUEST object and set its value to its own address.

In either of the above cases, the core-node SHOULD NOT send Notify messages to the edge-node.

When a core-node receives a NOTIFY_REQUEST object from an edge-node, it MAY update the Notify Node Address with its own address before forwarding it. In this case, when Notify messages are received, they MAY be selectively (based on local policy) forwarded to the edge-node.

6. Connection Deletion

6.1. Alarm-Free Connection Deletion

RSVP-TE 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 a Path or Resv message along the connection's path to inform all nodes en route to the intended deletion, prior to the actual deletion of the connection. The procedure is described in [RFC3473].

If an ingress core-node receives a PathTear without having first seen an ADMIN_STATUS object informing it that the connection is about to be deleted, it MAY pause the PathTear and first send a Path message with an ADMIN_STATUS object to inform all downstream LSRs that the connection is about to be deleted. When the Resv is received echoing the ADMIN_STATUS or using a timer as described in [RFC3473], the ingress core-node MUST forward the PathTear.

6.2. Connection Deletion with PathErr

[RFC3473] introduces the Path_State_Removed flag to a PathErr message to indicate that the sender has removed all state associated with the LSP and does not need to see a PathTear. A core-node next to an edge-node MAY map between teardown using ResvTear/PathTear and PathErr with Path_state_Removed.

A core-node next to an edge-node receiving a ResvTear from its downstream neighbor MAY respond with a PathTear and send a PathErr with Path_State_Removed further upstream.

Note, however, that a core-node next to an edge-node receiving a PathErr with Path_State_Removed from its downstream neighbor MUST NOT retain Path state and send a ResvTear further upstream because that would imply that Path state further downstream had also been retained.

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, it is RECOMMENDED that the following procedure based on [MPLS-HIER] is followed.

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 that 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 [MPLS-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

The trust model between the core and edge-nodes is different than the one described in [RFC3473], as the core is permitted to hide its topology from the edge-nodes, and the core is 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, Bala Rajagopalan, and Adrian Farrel for their comments and input. Thanks for thorough final reviews from Loa Andersson and Dimitri Papadimitriou.

Adrian Farrel edited the last two revisions of this document to incorporate comments from Working Group last call and from AD review.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

10.2. Informational References

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)", RFC 3477, January 2003.

[BUNDLE] Kompella, K., Rekhter, Y., and Berger, L., "Link Bundling in MPLS Traffic Engineering (TE)", RFC 4201, October 2005.

[EXPLICIT] Berger, L., "GMPLS Signaling Procedure for Egress Control", RFC 4003, February 2005.

[GMPLS-ARCH] Mannie, E., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[MPLS-HIER] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, October 2005.

[G.8080] ITU-T Rec. G.8080/Y.1304, "Architecture for the Automatically Switched Optical Network (ASON)," November 2001 (and Revision, January 2003). For information on the availability of this document, please see http://www.itu.int.

Authors' Addresses

George Swallow
Cisco Systems, Inc.
1414 Massachusetts Ave,
Boxborough, MA 01719

Phone: +1 978 936 1398
EMail: swallow@cisco.com

John Drake
Boeing Satellite Systems
2300 East Imperial Highway
El Segundo, CA 90245

Phone: +1 412 370-3108
EMail: John.E.Drake2@boeing.com

Hirokazu Ishimatsu
G1M Co., Ltd.
Nishinippori Start up Office 214,
5-37-5 Nishinippori, Arakawaku,
Tokyo 116-0013, Japan

Phone: +81 3 3891 8320
EMail: hirokazu.ishimatsu@g1m.jp

Yakov Rekhter
Juniper Networks, Inc.

EMail: yakov@juniper.net

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.