Internet Engineering Task Force
Internet Draft
draft-gan-fast-reroute-00.txt
April 10, 2001
Expires: October, 2001

Der-Hwa Gan
Ping Pan
Arthi Ayyangar
Kireeti Kompella
Juniper Networks

A Method for MPLS LSP Fast-Reroute Using RSVP Detours

STATUS OF THIS MEMO

Abstract

In this document, we introduce a method to establish backup LSP tunnels in large-scale networks. Two additional RSVP objects are proposed. These objects together make possible for routers using RSVP to create detours that can route around downstream links and nodes. As a result, an LSP can quickly and automatically repair itself, while redirecting the user traffic to the pre-computed and pre-established detour routes in event of network link and node failures. Packet loss is therefore minimized.

1 Introduction

The ability to quickly re-route traffic around failed links and nodes in a label switched path (LSP) can be extremely important to users.

When a network failure takes place, user data need to be re-directed over a detour path. The usefulness of a re-route mechanism is determined by the speed that a detour (or backup) path is established, the time it takes to re-route traffic, and how easy it is to configure detour path within the network.

To achieve timely detour path setup, one cannot establish a path after a failure is detected. Thus using pre-computed and pre-established detour path is essential for data traffic where packet loss is undesirable.

In order to achieve shortest re-route time, the detour decision must be made as close to the failure point as possible, since it may take significant time to notify other nodes in the LSP of such failure.

Since it is almost impossible to predict where failure may occur along an LSP, an ideal detour mechanism is to protect the entire LSP by establishing detour paths throughout the LSP. In the extreme, a fully protected LSP might require (N - 1) detour paths, where N is the number of hops that the LSP traverses. To minimize the path computation overhead, it is desirable for the detour paths to merge back to the main LSP as soon as possible. To simplify detour configuration, we need to automate how detour paths are established inside the network.

This document describes a method for automatically setting up detour paths over a RSVP signaled LSP[1,2], that offers the highest possible quality service to users while minimizing network overhead wherever possible. Detours are computed and established in a distributed fashion, continuously adapting to latest topology without manual intervention.

The procedures described in this document only protect unidirectional LSPs. Protecting bidirectional LSPs is left for further study. Throughout the document, the terms MPLS LSP and RSVP Session are used interchangeably.

2 Operation Overview

To protect from potential downstream link or node failures, a detour may be setup between the current node and one of the downstream nodes. The current node can be any node along a LSP except the LSP's egress, for the obvious reason that the egress node has no downstream link or node failure to speak of. Any downstream node of a LSP, which is more than one hop away from the current node, can be the detour merge point. For penultimate node, only the immediate downstream link need to be protect, so egress node is the detour merge point.

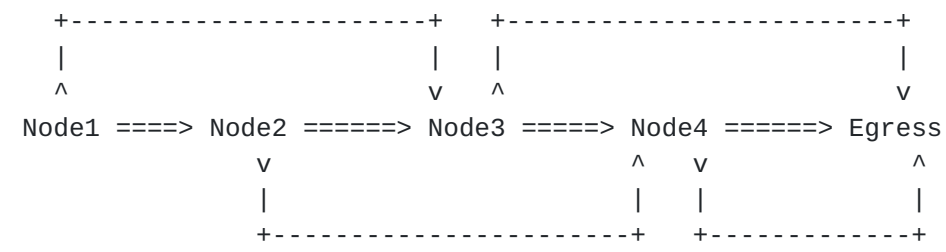


Figure 1: Example on where to initiate and terminate detour paths.

In Figure 1, detour (Node1, Node3) is created to protect against link failures between Node1-Node2 and Node2-Node3, as well as a node failure at Node2. This detour path is computed and initiated at Node1, and merges at Node3. In this case, Node2 and Node4 are not aware of the detour.

Likewise, detour (Node2, Node4) is created by Node2, to protect against link failures at Node2-Node3 and Node3-Node4, and node failure at Node3. Detour (Node4, Egress) is established to protect a local link between Node4 and Egress only.

A detour may traverse any number of transit nodes before merging at a downstream node. To be maximal flexible, there is no limitation on who the transit nodes are. The only hard limitation is that detour cannot traverse immediate downstream link and node.

We will describe the procedure on setting up detours in Section 4. But first, in the next section, we define two new RSVP objects that are used for the fast-reroute and detour operation.

3 RSVP Extension

Two new objects are defined to support LSP fast-reroute. The objects are the FAST_ROUTE object and the DETOUR object. Both objects can only be carried in RSVP Path messages. To support the proposed fast reroute functionality, an implementation MUST support both objects.

The objects are defined to be compatible with routers that do not recognize them (see Section 3.10 in [1]). For the routers that don't

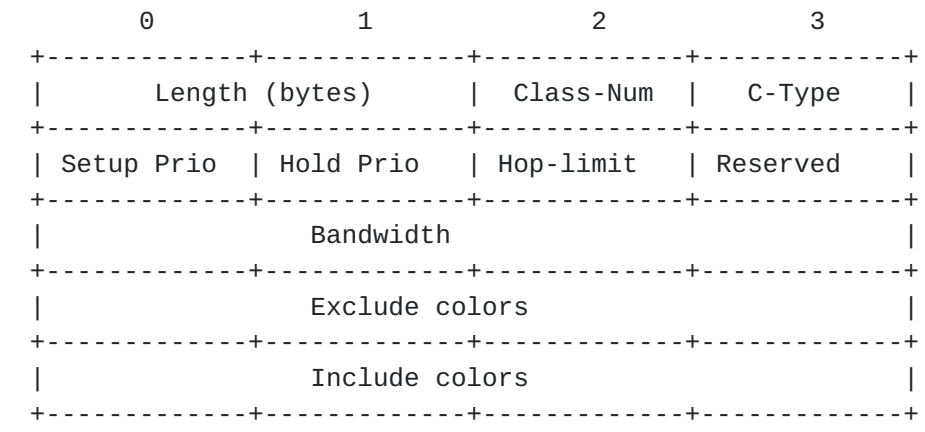
support the FAST_REROUTE objects, they MUST forward the objects downstream unchanged. For the routers that don't support the DETOUR objects, the routers MUST reject the message and send a PathErr to notify the sender.

This implies that, even if some nodes along a main LSP do not recognize or support the new objects, it is still possible to establish detour LSP's between the nodes that can support the new objects. At worst, the detour LSP's will not be established to protect the links between the non-supporting nodes. This feature can be very useful and important for network providers to deploy backup MPLS tunnels inside their networks.

3.1 FAST_REROUTE Objects

FAST_REROUTE Object

Class = 205 (use form 11bbbbbb for compatibility)
C-Type = 7



Setup Priority

The priority of the detour with respect to taking resources, in the range of 0 to 7. The value 0 is the highest priority. Setup Priority is used in deciding whether this session can preempt another session. See RSVP-TE draft for usage of priority.

Holding Priority

The priority of the detour with respect to holding resources, in the range of 0 to 7. The value 0 is the highest priority.

Holding Priority is used in deciding whether this session can be preempted by another session. See RSVP-TE for usage of priority.

Hop-limit

The maximum number of extra hops the detour is allowed to take, from current node (branching point) to merging node, with current node and merging node excluded in counting. For example, hop-limit of 0 means only direct links between current and merging nodes can be considered.

Reserved

This field is reserved. It MUST be set to zero on transmission and MUST be ignored on receipt.

Bandwidth

Bandwidth estimate (32-bit IEEE floating point integer) in bytes-per-second.

Exclude colors

A 32-bit vector representing a set of attribute filters associated with a detour any of which renders a link unacceptable.

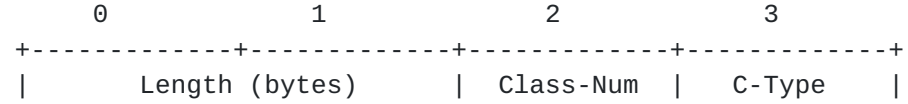
Include colors

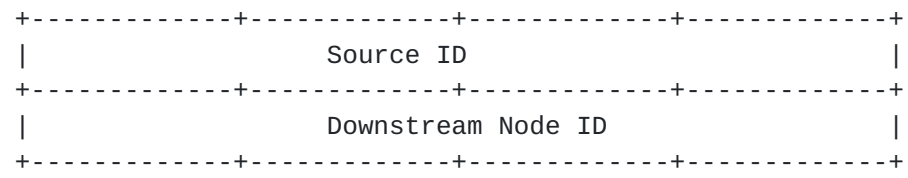
A 32-bit vector representing a set of attribute filters associated with a detour any of which renders a link acceptable (with respect to this test). A null set (all bits set to zero) automatically passes.

3.2 DETOUR Object

DETOUR Object

Class = 63 (to conform 0bbbbbbb format for compatibility)
C-Type = 7





Source ID

IPv4 address identifying the beginning point of detour. Any address on the branching node can be used.

Downstream Node ID

IP address identifying the downstream node that source is trying to avoid. Router ID of downstream node is preferred.

3.3 Message Formats

Both FAST_REROUTE and DETOUR objects MUST be carried in RSVP Path messages.

To request for a fast-reroute, the Path message MUST carry a FAST_REROUTE object. RSVP object ordering makes no logical difference, and an implementation should be ready to accept objects in any order. A recommended message format is the following:

```
<Path Message> ::=  <Common Header> [ <INTEGRITY> ]
                    <SESSION> <RSVP_HOP>
                    <TIME_VALUES>
                    [ <EXPLICIT_ROUTE> ]
                    <LABEL_REQUEST>
                    [ <SESSION_ATTRIBUTE> ]
                    <FAST_REROUTE>
                    [ <POLICY_DATA> ... ]
                    <sender descriptor>

<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                        [ <ADSPEC> ]
```

The presence of FAST_REROUTE object in a RSVP session indicates that the fast reroute service is requested. The object itself contains the information in aiding detour computations and setup.

After being processed at a node, the FAST_REROUTE MUST be stored by the node for later Path refreshes. A node, that recognizes FAST_REROUTE but cannot support it (possibly because temporary failure to compute a viable detour), should silently retry periodically. No PathErr should be sent. A RSVP node that does not recognize FAST_REROUTE MUST forward it unchanged. This has no impact on the main LSP. The worst result is that some transit nodes do not establish detours.

A node which accepts FAST_REROUTE MUST be ready to accept and correctly process DETOUR object for the same LSP.

To create a detour, a branching node MUST send out a Path message with DETOUR object in the following format. Once again, object placement and ordering make no logical difference.

```
   <Path Message> ::=  <Common Header> [ <INTEGRITY> ]
                       <SESSION> <RSVP_HOP>
                       <TIME_VALUES>
                       [ <EXPLICIT_ROUTE> ]
                       <LABEL_REQUEST>
                       [ <SESSION_ATTRIBUTE> ]
                       <DETOUR>
                       [ <POLICY_DATA> ... ]
                       <sender descriptor>

  <sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                          [ <ADSPEC> ]
```

At downstream nodes, the presence of DETOUR object indicates that this session is a detour. After being processed at a node, the DETOUR object MUST be stored for later Path refreshes.

A RSVP node that does not recognize DETOUR MUST send a PathErr with error code "Unknown object class" toward the sender. A sender here is the branching node that creates the DETOUR object, not ingress of the LSP. The sender should stop propagating PathErr further upstream. The PathErr causes the detour setup to fail, while the main LSP remains unaffected. The sender may choose to recompute a different detour, or

notify management that detour cannot be established.

It is illegal to have a Path message containing both FAST_REROUTE and DETOUR objects, since a session cannot be main and detour at the same time.

FAST_REROUTE object MUST only be present on main LSP, not on detours, whereas DETOUR object is only used for building a detour LSP. On downstream nodes, the presence of DETOUR object indicates that this session is a detour.

Applying the fast-reroute mechanism creates detour LSP's at intermediate nodes. Currently, the branching and merging procedures (described below) only support FF reservation style sessions. Supporting for shared reservation styles is left for future study.

RSVP is designed to cope gracefully with non-RSVP routers anywhere between senders and receivers. However, since LSP tunnels[2] doesn't work over non-RSVP cloud, and our extension only deals with LSP tunnels, thus the proposed RSVP extension does not work over non-RSVP cloud.

4 Setting up Detours

4.1 Detour Path Computation Algorithm

When a node receives FAST_REROUTE objects from the Path messages, it can trigger CSPF computation periodically to find out where to setup the detour paths. To protect an LSP, the node needs to first collect the following information:

1. A list of all downstream nodes that the LSP goes through. This information is readily available from the RECORD_ROUTE objects[2] during label setup.

2. The immediate outgoing link that the LSP goes through.

3. The downstream nodes that we want to protect against. Once again, this information is learnt from the RECORD_ROUTE objects.

4. The bandwidth requirement, hop-count limit, priority, and link coloring information for the detour. The FAST_REROUTE object can provide such information.

The node applies a typical CSPF algorithm to compute the route and the destination for a detour path. The path computation must satisfies the following constraints:

- The detour path originates from current node.
- It should not traverse to the immediate outgoing link.
- It should not traverse the downstream nodes that we try to protect against.
- It should satisfy all the requirements as specified in the FAST_REROUTE object.

If such computation succeeds, the node should trigger RSVP to establish a detour path immediately, and schedule a re-computation at a later time. The detour path should be as short as possible, and must merge back into the main LSP automatically at its destination. If for any reason, the node is unable to bring up a detour path, it must schedule a retry at a later time.

The node has the option to apply other constraints during the CSPF computation. For example, a simple method can be to terminate the computation as soon as a detour path is found. On the other hand, an implementation may wish to continue exhaustive search to discover an optimal path with lowest cost (or highest available bandwidth). The node also has the option to re-compute the detour path periodically even after the detour is up and running to ensure continuous adaptation to the latest network conditions.

The main advantage for running CSPF[3] here is to eliminate the needs for manual configuring detours, thus reduce the administrative overheads. However, it is important to be aware that the detour paths cannot cross the traffic engineering boundaries. A traffic engineering boundary is currently deliminated by an OSPF area, or an ISIS level.

4.2 Originating a Detour Request

At a detour branching node, a successful detour computation (described in Section 4.1) yields enough information to construct a LSP detour request. The information includes:

- A DETOUR object that specifies the detour's destination.
- A new EXPLICIT_ROUTE object toward the detour's destination.
- A new next-hop router's address to construct a new RSVP_HOP object.
- The out-going interface information to send the detour request. The out-going interface must be different from the

one used by the main LSP.

A Path message that is used to setup a detour path MUST consists of the new DETOUR, ERO and RSVP_HOP objects. In addition, the SENDER_TSPEC object contains the bandwidth information from the previously received FAST_REROUTE objects. However, the Path message MUST not have a FAST_REROUTE object.

The branching node MUST not mix the messages for the main and the detour LSP's. When it receives Resv, ResvTear and PathErr messages from the downstream detour destination, the messages MUST not be forwarded upstream. Similarly, when it receives ResvErr and ResvConf messages from upstream, the node MUST not propagate them onto the detour LSP.

In RSVP-TE operation, the session tear-down request is normally originated by the sender via PathTear messages. During error conditions, the network routers can send ResvTear messages to fix problems on the failing path. Thus, when the branching node receives a PathTear message from upstream, it MUST tear-down both the main and detour LSP's. The PathTear messages MUST propagate to both main and detour LSP's. On the other hand, the branching node may receive ResvTear messages from downstream for the main LSP. As long as a detour is up, the ResvTear messages MUST not sent further upstream to ingress.

4.3 Detour Merging Nodes

A detour merging node is where a detour LSP merges back onto a main LSP. A merging node may receive multiple Path messages from different interfaces, but with identical SESSION and SENDER_TEMPLATE objects. To maintain the LSP's, it is important for the non-egress node to identify the main LSP's from the detour LSP's (by the way, it is possible to have multiple detour LSP's to merge at a single node). Generally, a Path message for a detour LSP MUST contain a DETOUR object, whereas a Path that has a FAST_REROUTE object MUST represent a main LSP.

4.4 Loop detection

It is possible to have detours traversing through the same nodes as the main LSP. To make the proposed detour method working, the nodes MUST process the detour LSP's independently from the main LSP's during the LSP loop detection procedure as suggested in [2].

5 Security Considerations

This document does not introduce new security issues. The

specification adds new objects to RSVP. Therefore, the security considerations pertaining to the original RSVP protocol[1] remain relevant.

6 Intellectual Property Considerations

Juniper Networks, Inc. is seeking patent protection on technology described in this Internet-Draft. If technology in this Internet-Draft is adopted as a standard, Juniper Networks agrees to license, on reasonable and non-discriminatory terms, any patent rights it obtains covering such technology to the extent necessary to comply with the standard.

7 Acknowledgments

We acknowledge the helpful comments from Yakov Rekhter, Manoj Leelanivas, Nischal Sheth and Cliff DeGuzman.

8 Bibliography

[1] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation protocol (RSVP) -- version 1 functional specification," Request for Comments 2205, Internet Engineering Task Force, Sept. 1997.

[2] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP tunnels," Internet Draft, Internet Engineering Task Force, Feb. 2001. Work in progress.

[3] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," Request for Comments 2702, Internet Engineering Task Force, Sept. 1999.

9 Authors

Der-Hwa Gan (dhg@juniper.net)
Ping Pan (pingpan@juniper.net)
Arthi Ayyangar (arthi@juniper.net)
Kireeti Kompella (kireeti@juniper.net)
Juniper Networks, Inc.
1194 N. Mathilda Avenue
Sunnyvale, CA 94089

Table of Contents