Internet Engineering Task Force (IETF)
Request for Comments: 8424
Category: Experimental
ISSN: 2070-1721

H. Chen, Ed.
Huawei Technologies
R. Torvi, Ed.
Juniper Networks

August 2018

# Extensions to RSVP-TE for Label Switched Path (LSP) Ingress Fast Reroute (FRR) Protection

## Abstract

This document describes extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for locally protecting the ingress node of a Point-to-Point (P2P) or Point-to-Multipoint (P2MP) Traffic Engineered (TE) Label Switched Path (LSP). It extends the Fast Reroute (FRR) protection for transit nodes of an LSP to the ingress node of the LSP. The procedures described in this document are experimental.

## Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8424.

Table of Contents

## 1. Introduction

For an MPLS Traffic Engineered (TE) Label Switched Path (LSP), protecting the failures of its transit nodes using Fast Reroute (FRR) is covered in [RFC4090] for Point-to-Point (P2P) LSPs and [RFC4875] for Point-to-Multipoint (P2MP) LSPs. However, protecting the failure of its ingress node using FRR is not covered in either [RFC4090] or [RFC4875]. The MPLS Transport Profile (MPLS-TP) Linear Protection described in [RFC6378] can provide a protection against the failure of any transit node of an LSP between the ingress node and the egress node of the LSP, but it cannot protect against the failure of the ingress node.

To protect against the failure of the (primary) ingress node of a primary end-to-end P2MP (or P2P) TE LSP, a typical existing solution is to set up a secondary backup end-to-end P2MP (or P2P) TE LSP. The backup LSP is from a backup ingress node to backup egress nodes (or node). The backup ingress node is different from the primary ingress node. The backup egress nodes (or node) are (or is) different from the primary egress nodes (or node) of the primary LSP. For a P2MP TE LSP, on each of the primary (and backup) egress nodes, a P2P LSP is created from the egress node to its primary (backup) ingress node and configured with Bidirectional Forwarding Detection (BFD). This is used to detect the failure of the primary (backup) ingress node for the receiver to switch to the backup (or primary) egress node to receive the traffic after the primary (or backup) ingress node fails when both the primary LSP and the secondary LSP carry the traffic. In addition, FRR may be used to provide protections against the failures of the transit nodes and the links of the primary and secondary end-to-end TE LSPs.

There are a number of issues in this solution:

- It consumes lots of network resources. Double states need to be maintained in the network since two end-to-end TE LSPs are created. Double link bandwidth is reserved and used when both the primary and the secondary end-to-end TE LSPs carry the traffic at the same time.

- More operations are needed, which include the configuration of two end-to-end TE LSPs and BFDs from each of the egress nodes to its corresponding ingress node.

- The detection of the failure of the ingress node may not be reliable. Any failure on the path of the BFD from an egress node to an ingress node may cause the BFD to indicate the failure of the ingress node.

- The speed of protection against the failure of the ingress node may be slow.

This specification defines a simple extension to RSVP-TE for local protection (FRR) of the ingress node of a P2MP or P2P LSP to resolve these issues. Ingress local protection and ingress FRR protection will be used interchangeably.

Note that this document is an Experimental RFC. Two different approaches are proposed to transfer the information for ingress protection. They both use the same new INGRESS_PROTECTION object, which is sent in both PATH and RESV messages between a primary ingress and a backup ingress. One approach is the Relay-Message Method (refer to Sections 6.1.1 and 6.2.1), the other is the Proxy-Ingress Method (refer to Sections 6.1.2 and 6.2.2). Each of them has advantages and disadvantages. It is hard to decide which one is used as a standard approach now. It is expected that the experiment on the ingress local protection with these two approaches will provide quantities to help choose one. The quantities include the numbers on control traffic, states, codes, and operations. After one approach is selected, the document will be revised to reflect that selection and any other items learned from the experiment. The revised document is expected to be submitted for publication on the standards track.

## 1.1. Ingress Local Protection Example

Figure 1 shows an example of using a backup P2MP LSP to locally protect the ingress of a primary P2MP LSP, which is from ingress Ia to three egresses: L1, L2, and L3. The backup LSP is from backup ingress Ib to the next hops of ingress Ia: R2 and R4.

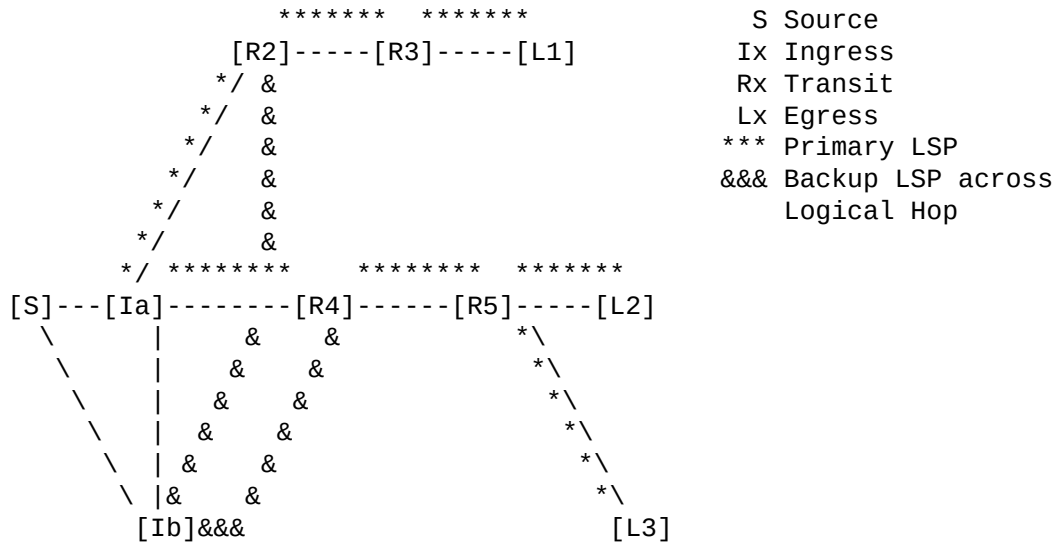



Figure 1: Ingress Local Protection

In normal operations, source S sends the traffic to primary ingress Ia. Ia imports the traffic into the primary LSP.

When source S detects the failure of Ia, it switches the traffic to backup ingress Ib, which imports the traffic from S into the backup LSP to Ia's next hops, R2 and R4, where the traffic is merged into the primary LSP and then sent to egresses L1, L2, and L3.

Note that the backup ingress is one logical hop away from the ingress. A logical hop is a direct link or a tunnel (such as a GRE tunnel) over which RSVP-TE messages may be exchanged.

## 1.2. Ingress Local Protection Overview

There are four parts in ingress local protection:

- setting up the necessary backup LSP forwarding state based on the information received for ingress local protection;
- detecting the primary ingress failure and providing the fast repair (as discussed in Sections 3 and 4);
- maintaining the RSVP-TE control-plane state until a global repair is done; and,
- performing the global repair (see Section 6.4.2).

The primary ingress of a primary LSP sends the backup ingress the information for ingress protection in a PATH message with a new INGRESS_PROTECTION object. The backup ingress sets up the backup LSP(s) and forwarding state after receiving the necessary information for ingress protection. Then, it sends the primary ingress the status of ingress protection in a RESV message with a new INGRESS_PROTECTION object.

When the primary ingress fails, the backup ingress sends or refreshes the next hops of the primary ingress the PATH messages without any INGRESS_PROTECTION object after verifying the failure. Thus, the RSVP-TE control-plane state of the primary LSP is maintained.

## 2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Ingress Failure Detection

Exactly how to detect the failure of the ingress is out of scope. However, it is necessary to discuss different modes for detecting the failure because they determine what is the required behavior for the source and backup ingress.

### 3.1. Source Detects Failure

Source Detects Failure, or Source-Detect for short, means that the source is responsible for "fast detecting" the failure of the primary ingress of an LSP. Fast detecting the failure means detecting the failure in a few or tens of milliseconds. The backup ingress is ready to import the traffic from the source into the backup LSP(s) after the backup LSP(s) is up.

In normal operations, the source sends the traffic to the primary ingress. When the source detects the failure of the primary ingress, it switches the traffic to the backup ingress, which delivers the traffic to the next hops of the primary ingress through the backup LSP(s), where the traffic is merged into the primary LSP.

For an LSP, after the primary ingress fails, the backup ingress MUST use a method to verify the failure of the primary ingress before the PATH message for the LSP expires at the next hop of the primary ingress. After verifying the failure, the backup ingress sends/

refreshes the PATH message to the next hop through the backup LSP as needed. The method may verify the failure of the primary ingress slowly, such as in seconds.

After the primary ingress fails, it will not be reachable after routing convergence. Thus, checking whether the primary ingress (address) is reachable is a possible method.

When the previously failed primary ingress of a primary LSP becomes available again and the primary LSP has recovered from its primary ingress, the source may switch the traffic to the primary ingress from the backup ingress. An operator may control the traffic switch through using a command on the source node after seeing that the primary LSP has recovered.

## 3.2. Backup and Source Detect Failure

Backup and Source Detect Failure, or Backup-Source-Detect for short, means that both the backup ingress and the source are concurrently responsible for fast detecting the failure of the primary ingress.

Note that one of the differences between Source-Detect and Backup-Source-Detect is the following: in the former, the backup ingress verifies the failure of the primary ingress slowly, such as in seconds; in the latter, the backup ingress detects the failure fast, such as in a few or tens of milliseconds.

In normal operations, the source sends the traffic to the primary ingress. It switches the traffic to the backup ingress when it detects the failure of the primary ingress.

The backup ingress does not import any traffic from the source into the backup LSP in normal operations. When it detects the failure of the primary ingress, it imports the traffic from the source into the backup LSP to the next hops of the primary ingress, where the traffic is merged into the primary LSP.

The Source-Detect is preferred. It is simpler than the Backup-Source-Detect, which needs both the source and the backup ingress to detect the ingress failure quickly.

## 4. Backup Forwarding State

Before the primary ingress fails, the backup ingress is responsible for creating the necessary backup LSPs. These LSPs might be multiple bypass P2P LSPs that avoid the ingress. Alternately, the backup ingress could choose to use a single backup P2MP LSP as a bypass or detour to protect the primary ingress of a primary P2MP LSP.

The backup ingress may be "off path" or "on path" of an LSP. If a backup ingress is not any node of the LSP, it is off path. If a backup ingress is a next hop of the primary ingress of the LSP, it is on path. When a backup ingress for protecting the primary ingress is configured, the backup ingress MUST not be on the LSP except for if it is the next hop of the primary ingress. If it is on path, the primary forwarding state associated with the primary LSP SHOULD be clearly separated from the backup LSP(s) state.

### 4.1. Forwarding State for Backup LSP

A forwarding entry for a backup LSP is created on the backup ingress after the LSP is set up. Depending on the failure-detection mode (e.g., Source-Detect), it may be used to forward received traffic or simply be inactive (e.g., Backup-Source-Detect) until required. In either case, when the primary ingress fails, this entry is used to import the traffic into the backup LSP to the next hops of the primary ingress, where the traffic is merged into the primary LSP.

The forwarding entry for a backup LSP is a local implementation issue. In one device, it may have an inactive flag. This inactive forwarding entry is not used to forward any traffic normally. When the primary ingress fails, it is changed to active; thus, the traffic from the source is imported into the backup LSP.

## 5. Protocol Extensions

A new object, INGRESS_PROTECTION, is defined for signaling ingress local protection. The primary ingress of a primary LSP sends the backup ingress this object in a PATH message. In this case, the object contains the information needed to set up ingress protection. The information includes:

- the Backup Ingress IP Address, which indicates the backup ingress;
- the TRAFFIC_DESCRIPTOR, which describes the traffic that the primary LSP transports (this traffic is imported into the backup LSP(s) on the backup ingress when the primary ingress fails);

- the Labels and Routes, which indicate the first hops of the primary LSP, each of which is paired with its label; and,

- the Desire options on ingress protection, such as a P2MP option, which indicates a desire to use a backup P2MP LSP to protect the primary ingress of a primary P2MP LSP.

The backup ingress sends the primary ingress this object in a RESV message. In this case, the object contains the information about the status on the ingress protection.

## 5.1. INGRESS_PROTECTION Object

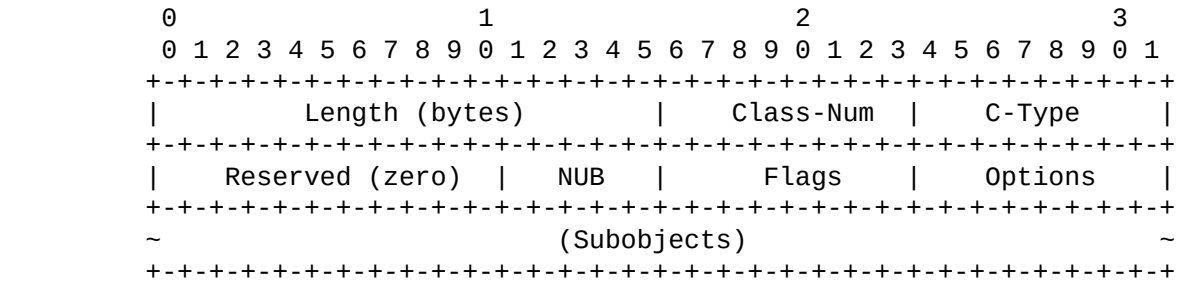
### 5.1.1. Class Number and Class Type

The Class Number for the INGRESS_PROTECTION object MUST be of the form 0bbbbbbb to enable implementations that do not recognize the object to reject the entire message and return an "Unknown Object Class" error [RFC2205]. It is suggested that a Class Number value from the Private Use range (124-127) [RFC3936] specified for the 0bbbbbbb octet be chosen for this experiment. It is also suggested that a Class Type value of 1 be used for this object in this experiment.

The INGRESS_PROTECTION object with the FAST_REROUTE object in a PATH message is used to control the backup for protecting the primary ingress of a primary LSP. The primary ingress MUST insert this object into the PATH message to be sent to the backup ingress for protecting the primary ingress.

### 5.1.2. Object Format

The INGRESS_PROTECTION object has the following format:




Flags
- 0x01 Ingress local protection available
- 0x02 Ingress local protection in use
- 0x04 Bandwidth protection

Options
- 0x01 Revert to Ingress
- 0x02 P2MP Backup

For protecting the ingress of a P2MP LSP, if the backup ingress doesn't have a backup LSP to each of the next hops of the primary ingress, it SHOULD clear "Ingress local protection available" and set the Number of Unprotected Branches (NUB) to the number of the next hops to which there is no backup LSP.

The flags are used to communicate status information from the backup ingress to the primary ingress.

- Ingress local protection available: The backup ingress MUST set this flag after backup LSPs are up and ready for locally protecting the primary ingress. The backup ingress sends this to the primary ingress to indicate that the primary ingress is locally protected.

- Ingress local protection in use: The backup ingress MUST set this flag when it detects a failure in the primary ingress and actively redirects the traffic into the backup LSPs. The backup ingress records this flag and does not send any RESV messages with this flag to the primary ingress since the primary ingress is down.

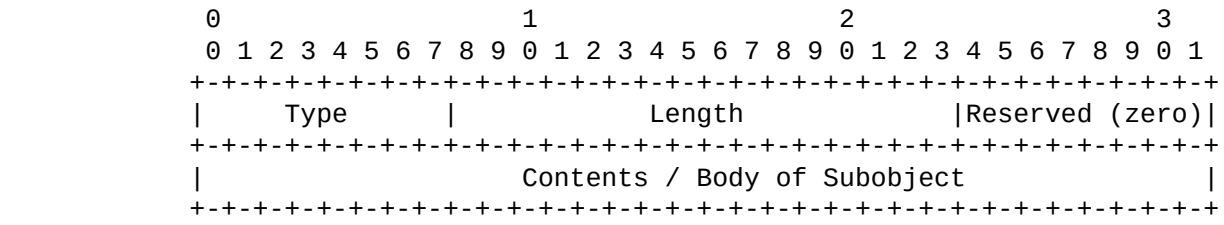
- Bandwidth protection: The backup ingress MUST set this flag if the backup LSPs guarantee to provide the desired bandwidth for the protected LSP against the primary ingress failure.

The options are used by the primary ingress to specify the desired behavior to the backup ingress.

- Revert to Ingress: The primary ingress sets this option, which indicates that the traffic for the primary LSP, if successfully resignaled, will be switched back to the primary ingress from the backup ingress when the primary ingress is restored.

- P2MP Backup: This option is set to ask for the backup ingress to use backup P2MP LSP to protect the primary ingress.

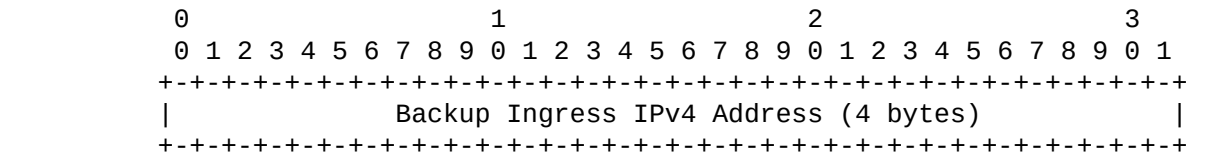
The INGRESS_PROTECTION object may contain some subobjects of following format:




where Type is the type of a subobject and Length is the total size of the subobject in bytes, including Type, Length, and Contents fields.

### 5.1.3. Subobject: Backup Ingress IPv4 Address

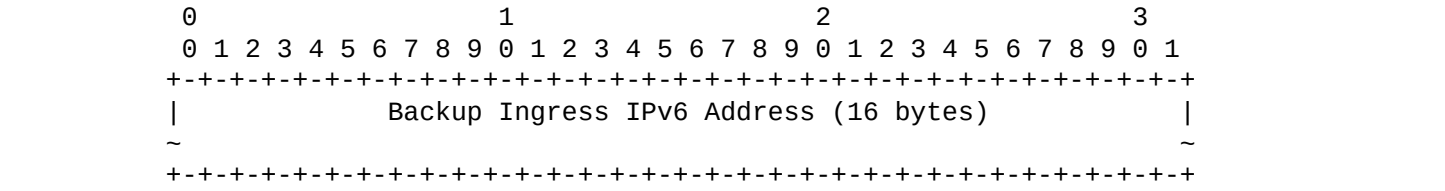
When the primary ingress of a protected LSP sends a PATH message with an INGRESS_PROTECTION object to the backup ingress, the object MUST have a Backup Ingress IPv4 Address subobject containing an IPv4 address belonging to the backup ingress if IPv4 is used. The Type of the subobject is 1, and the body of the subobject is given below:




Backup Ingress IPv4 Address: An IPv4 host address of backup ingress

### 5.1.4. Subobject: Backup Ingress IPv6 Address

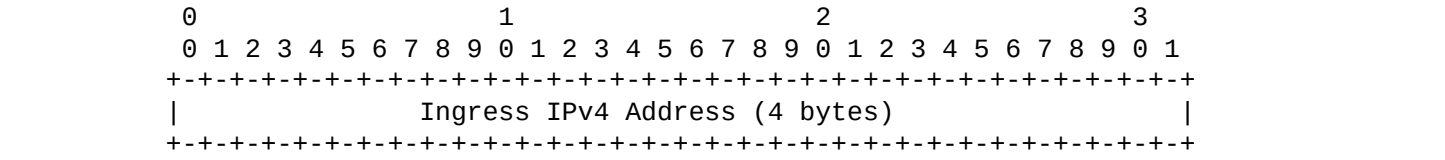
When the primary ingress of a protected LSP sends a PATH message with an INGRESS_PROTECTION object to the backup ingress, the object MUST have a Backup Ingress IPv6 Address subobject containing an IPv6 address belonging to the backup ingress if IPv6 is used. The Type of the subobject is 2, the body of the subobject is given below:




Backup Ingress IPv6 Address: An IPv6 host address of backup ingress

### 5.1.5. Subobject: Ingress IPv4 Address

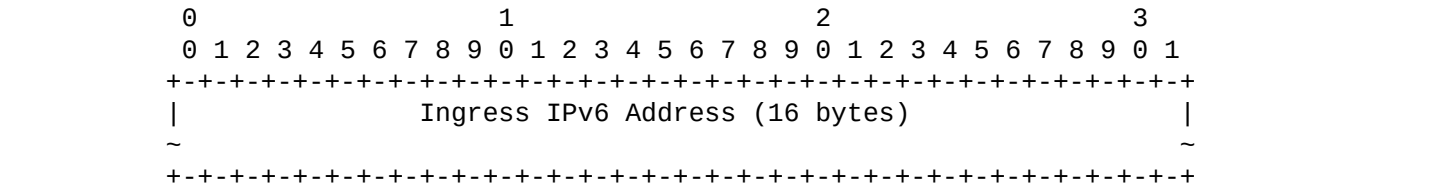
The INGRESS_PROTECTION object may have an Ingress IPv4 Address subobject containing an IPv4 address belonging to the primary ingress if IPv4 is used. The Type of the subobject is 3. The subobject has the following body:




Ingress IPv4 Address: An IPv4 host address of ingress

### 5.1.6. Subobject: Ingress IPv6 Address

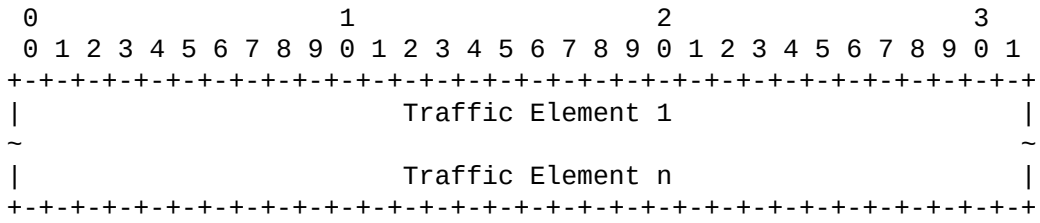
The INGRESS_PROTECTION object may have an Ingress IPv6 Address subobject containing an IPv6 address belonging to the primary ingress if IPv6 is used. The Type of the subobject is 4. The subobject has the following body:




Ingress IPv6 Address: An IPv6 host address of ingress

### 5.1.7. Subobject: TRAFFIC_DESCRIPTOR

The INGRESS_PROTECTION object may have a TRAFFIC_DESCRIPTOR subobject describing the traffic to be mapped to the backup LSP on the backup ingress for locally protecting the primary ingress. The subobject types for Interface, IPv4 Prefix, IPv6 Prefix, and Application Identifier are 5, 6, 7, and 8, respectively. The subobject has the following body:




The TRAFFIC_DESCRIPTOR subobject may contain multiple Traffic Elements of the same type as follows:

- Interface Traffic: Each of the Traffic Elements is a 32-bit index of an interface from which the traffic is imported into the backup LSP.

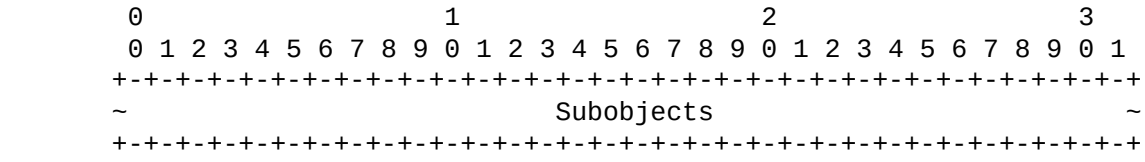
- IPv4 Prefix Traffic: Each of the Traffic Elements is an IPv4 prefix that contains an 8-bit prefix length followed by an IPv4 address prefix (whose length, in bits, is specified by the prefix length) that is padded to a byte boundary.

- IPv6 Prefix Traffic Each of the Traffic Elements is an IPv6 prefix, containing an 8-bit prefix length followed by an IPv6 address prefix (whose length, in bits, is specified by the prefix length) that is padded to a byte boundary.

- Application Traffic: Each of the Traffic Elements is a 32-bit identifier of an application from which the traffic is imported into the backup LSP.

### 5.1.8. Subobject: Label-Routes

The INGRESS_PROTECTION object in a PATH message from the primary ingress to the backup ingress may have a Label-Routes subobject containing the labels and routes that the next hops of the ingress use. The Type of the subobject is 9. The subobject has the following body:




The Subobjects in Label-Routes are copied from those in the RECORD_ROUTE objects in the RESV messages that the primary ingress receives from its next hops for the primary LSP. They MUST contain the first hops of the LSP, each of which is paired with its label.

## 6. Behavior of Ingress Protection

### 6.1. Overview

There are two different proposed signaling approaches to transfer the information for ingress protection. They both use the same new INGRESS_PROTECTION object. The object is sent in both PATH and RESV messages.

#### 6.1.1. Relay-Message Method

The primary ingress relays the information for ingress protection of an LSP to the backup ingress via PATH messages. Once the LSP is created, the ingress of the LSP sends the backup ingress a PATH message with an INGRESS_PROTECTION object with a Label-Routes subobject, which is populated with the next hops and labels. This provides sufficient information for the backup ingress to create the appropriate forwarding state and backup LSP(s).

The ingress also sends the backup ingress all the other PATH messages for the LSP with an empty INGRESS_PROTECTION object. An INGRESS_PROTECTION object without any TRAFFIC_DESCRIPTOR subobject is called an empty INGRESS_PROTECTION object. Thus, the backup ingress has access to all the PATH messages needed for modification to refresh the control-plane state after a failure.

The empty INGRESS_PROTECTION object is for efficient processing of ingress protection for a P2MP LSP. A P2MP LSP’s primary ingress may have more than one PATH message, each of which is sent to a next hop

along a branch of the P2MP LSP. The PATH message along a branch will be selected and sent to the backup ingress with an INGRESS_PROTECTION object containing the TRAFFIC_DESCRIPTOR subobject; all the PATH messages along the other branches will be sent to the backup ingress containing an INGRESS_PROTECTION object without any TRAFFIC_DESCRIPTOR subobject (empty INGRESS_PROTECTION object). For a P2MP LSP, the backup ingress only needs one TRAFFIC_DESCRIPTOR.

### 6.1.2. Proxy-Ingress Method

Conceptually, a proxy ingress is created that starts the RSVP signaling. The explicit path of the LSP goes from the proxy ingress to the backup ingress and then to the real ingress. The behavior and signaling for the proxy ingress is done by the real ingress; the use of a proxy-ingress address avoids problems with loop detection. Note that the proxy ingress MUST reside within the same router as the real ingress.

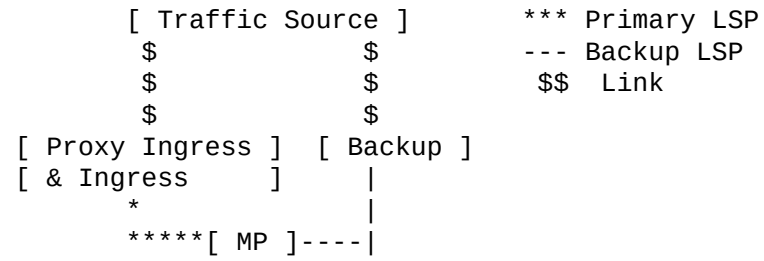



Figure 2: Example of a Protected LSP with a Proxy-Ingress Node

The backup ingress MUST know the merge points or next hops and their associated labels. This is accomplished by having the RSVP PATH and RESV messages go through the backup ingress, although the forwarding path need not go through the backup ingress. If the backup ingress fails, the ingress simply removes the INGRESS_PROTECTION object and forwards the PATH messages to the LSP's next hop(s). If the ingress has its LSP configured for ingress protection, then the ingress can add the backup ingress and itself to the Explicit Route Object (ERO) and start forwarding the PATH messages to the backup ingress.

Slightly different behavior can apply for the on-path and off-path cases. In the on-path case, the backup ingress is a next-hop node after the ingress for the LSP. In the off-path case, the backup ingress is not any next-hop node after the ingress for all associated sub-LSPs.

The key advantage of this approach is that it minimizes the special handling code required. Because the backup ingress is on the signaling path, it can receive various notifications. It easily has

access to all the PATH messages needed for a modification to be sent to refresh the control-plane state after a failure.

## 6.2. Ingress Behavior

The primary ingress MUST be configured with a couple of pieces of information for ingress protection.

- Backup Ingress Address: The primary ingress MUST know the IP address of the backup ingress it wants to be used before it can use the INGRESS_PROTECTION object.

- Proxy-Ingress-Id (only needed for Proxy-Ingress Method): The Proxy-Ingress-Id is only used in the RECORD_ROUTE object for recording the proxy ingress. If no Proxy-Ingress-Id is specified, then a local interface address that will not otherwise be included in the RECORD_ROUTE object can be used. A similar technique is used in Section 6.1.1. of [RFC4090].

- Application Traffic Identifier: The primary ingress and backup ingress MUST both know what application traffic should be directed into the LSP. If a list of prefixes in the TRAFFIC_DESCRIPTOR subobject will not suffice, then a commonly understood Application Traffic Identifier can be sent between the primary ingress and backup ingress. The exact meaning of the identifier should be configured similarly at both the primary ingress and backup ingress. The Application Traffic Identifier is understood within the unique context of the primary ingress and backup ingress.

- A Connection between Backup Ingress and Primary Ingress: If there is not any direct link between the primary ingress and the backup ingress, a tunnel MUST be configured between them.

With this additional information, the primary ingress can create and signal the necessary RSVP extensions to support ingress protection.

### 6.2.1. Relay-Message Method

To protect the primary ingress of an LSP, the primary ingress MUST do the following after the LSP is up.

1. Select a PATH message P0 for the LSP.

2. If the backup ingress is off path (the backup ingress is not the next hop of the primary ingress for P0), then send it a PATH message P0’ with the content from P0 and an INGRESS_PROTECTION object; else (the backup ingress is a next hop, i.e., on-path case) add an INGRESS_PROTECTION object into the existing PATH

message to the backup ingress (i.e., the next hop). The object contains the TRAFFIC_DESCRIPTOR subobject, the Backup Ingress Address subobject and the Label-Routes subobject. The options field is set to indicate whether a backup P2MP LSP is desired. The Label-Routes subobject contains the next hops of the primary ingress and their labels. Note that for the on-path case, there is an existing PATH message to the backup ingress (i.e., the next hop), and we just add an INGRESS_PROTECTION object into the existing PATH message to be sent to the backup ingress. We do not send a separate PATH message to the backup ingress for this existing PATH message.

3. For each Pi of the other PATH messages for the LSP, send the backup ingress a PATH message Pi' with the content copied from Pi and an empty INGRESS_PROTECTION object.

For every PATH message Pj' (i.e., P0'/Pi') to be sent to the backup ingress, it has the same SESSION as Pj (i.e., P0/Pi). If the backup ingress is off path, the primary ingress updates Pj' according to the backup ingress as its next hop before sending it. It adds the backup ingress to the beginning of the ERO and sets RSVP_HOP based on the interface to the backup ingress. The primary ingress MUST NOT set up any forwarding state to the backup ingress if the backup ingress is off path.

### 6.2.2. Proxy-Ingress Method

The primary ingress is responsible for starting the RSVP signaling for the proxy-ingress node. To do this, the following MUST be done for the RSVP PATH message.

1. Compute the EROs for the LSP as normal for the ingress.

2. If the selected backup ingress node is not the first node on the path (for all sub-LSPs), then insert it at the beginning of the ERO first, then the backup ingress node, and then the ingress node.

3. In the PATH RECORD_ROUTE Object (RRO), instead of recording the ingress node's address, replace it with the Proxy-Ingress-Id.

4. Leave the hop (HOP) object populated as usual with information for the ingress node.

5. Add the INGRESS_PROTECTION object to the PATH message. Include the Backup Ingress Address (IPv4 or IPv6) subobject and the TRAFFIC_DESCRIPTOR subobject. Set or clear the options indicating that a backup P2MP LSP is desired.

6. Optionally, add the FAST-REROUTE object [RFC4090] to the Path message. Indicate whether one-to-one backup is desired. Indicate whether facility backup is desired.

7. The RSVP PATH message is sent to the backup node as normal.

If the ingress detects that it can't communicate with the backup ingress, then the ingress SHOULD instead send the PATH message to the next hop indicated in the ERO computed in step 1. Once the ingress detects that it can communicate with the backup ingress, the ingress SHOULD follow steps 1-7 to obtain ingress failure protection.

When the ingress node receives an RSVP PATH message with an INGRESS_PROTECTION object and the object specifies that node as the ingress node and the Previous Hop (PHOP) as the backup ingress node, the ingress node SHOULD remove the INGRESS_PROTECTION object from the PATH message before sending it out. Additionally, the ingress node MUST store that it will install ingress forwarding state for the LSP rather than midpoint forwarding.

When an RSVP RESV message is received by the ingress, it uses the Next Hop (NHOP) to determine whether the message is received from the backup ingress or from a different node. The stored associated PATH message contains an INGRESS_PROTECTION object that identifies the backup ingress node. If the RESV message is not from the backup node, then the ingress forwarding state SHOULD be set up, and the INGRESS_PROTECTION object MUST be added to the RESV before it is sent to the NHOP, which SHOULD be the backup node. If the RESV message is from the backup node, then the LSP SHOULD be considered available for use.

If the backup ingress node is on the forwarding path, then a RESV is received with an INGRESS_PROTECTION object and an NHOP that matches the backup ingress. In this case, the ingress node's address will not appear after the backup ingress in the RRO. The ingress node SHOULD set up the ingress forwarding state, just as is done if the ingress node of the LSP weren't protected.

## 6.3. Backup Ingress Behavior

A Label Edge Router (LER) determines that the ingress local protection is requested for an LSP if the INGRESS_PROTECTION object is included in the PATH message it receives for the LSP. The LER can further determine that it is the backup ingress if one of its addresses is in the Backup Ingress Address subobject of the INGRESS_PROTECTION object. The LER as the backup ingress will assume full responsibility of the ingress after the primary ingress fails. In addition, the LER determines that it is off path if it is not any

node of the LSP. The LER determines whether it uses the Relay-Message Method or the Proxy-Ingress Method according to configurations.

### 6.3.1. Backup Ingress Behavior in the Off-Path Case

The backup ingress considers itself a Point of Local Repair (PLR) and the primary ingress its next hop, and it provides a local protection for the primary ingress. It behaves very similarly to a PLR providing fast reroute where the primary ingress is considered to be the failure point to protect. Where not otherwise specified, the behavior given in [RFC4090] for a PLR applies.

The backup ingress MUST follow the control options specified in the INGRESS_PROTECTION object and the flags and specifications in the FAST-REROUTE object. This applies to providing a P2MP backup if the "P2MP backup" is set, a one-to-one backup if "one-to-one desired" is set, a facility backup if the "facility backup desired" is set, and backup paths that support both the desired bandwidth and administrative groups that are requested.

If multiple non-empty INGRESS_PROTECTION objects have been received via multiple PATH messages for the same LSP, then the most recent one MUST be the one used.

The backup ingress creates the appropriate forwarding state for the backup LSP tunnel(s) to the merge point(s).

When the backup ingress sends a RESV message to the primary ingress, it MUST add an INGRESS_PROTECTION object into the message. It MUST set or clear the flags in the object to report "Ingress local protection available", "Ingress local protection in use", and "bandwidth protection".

If the backup ingress doesn’t have a backup LSP tunnel to each of the merge points, it SHOULD clear "Ingress local protection available" and set NUB to the number of the merge points to which there is no backup LSP.

When the primary ingress fails, the backup ingress redirects the traffic from a source into the backup P2P LSPs or the backup P2MP LSP transmitting the traffic to the next hops of the primary ingress, where the traffic is merged into the protected LSP.

In this case, the backup ingress MUST keep the PATH message with the INGRESS_PROTECTION object received from the primary ingress and the RESV message with the INGRESS_PROTECTION object to be sent to the primary ingress. The backup ingress MUST set the "local protection

in use" flag in the RESV message, which indicates that the backup ingress is actively redirecting the traffic into the backup P2P LSPs or the backup P2MP LSP for locally protecting the primary ingress failure.

Note that the RESV message with this piece of information will not be sent to the primary ingress because the primary ingress has failed.

If the backup ingress has not received any PATH messages from the primary ingress for an extended period of time (e.g., a cleanup timeout interval) and a confirmed primary ingress failure did not occur, then the standard RSVP soft-state removal SHOULD occur. The backup ingress SHALL remove the state for the PATH message from the primary ingress and either tear down the one-to-one backup LSPs for protecting the primary ingress if one-to-one backup is used or unbind the facility backup LSPs if facility backup is used.

When the backup ingress receives a PATH message from the primary ingress for locally protecting the primary ingress of a protected LSP, it MUST check to see if any critical information has been changed. If the next hops of the primary ingress are changed, the backup ingress SHALL update its backup LSP(s) accordingly.

#### 6.3.1.1. Relay-Message Method

When the backup ingress receives a PATH message with a non-empty INGRESS_PROTECTION object, it examines the object to learn what traffic associated with the LSP. It determines the next hops to be merged to by examining the Label-Routes subobject in the object.

The backup ingress MUST store the PATH message received from the primary ingress but NOT forward it.

The backup ingress responds with a RESV message to the PATH message received from the primary ingress. If the backup ingress is off path, the LABEL object in the RESV message contains IMPLICIT-NULL. If the INGRESS_PROTECTION object is not "empty", the backup ingress SHALL send the RESV message with the state indicating protection is available after the backup LSP(s) are successfully established.

#### 6.3.1.2. Proxy-Ingress Method

When receiving a RESV message for an LSP from a router that is not primary ingress, the backup ingress collects the pair of (IPv4/IPv6 subobject, Label subobject) in the second place to the top pair in the RECORD_ROUTE object of the message. It determines the next hops to be merged according to the set of the pairs collected. If a Label-Routes subobject is included in the INGRESS_PROTECTION object,

the included IPv4/IPv6 subobjects are used to filter the set down to the specific next hops where protection is desired. An RESV message MUST have been received before the backup ingress can create or select the appropriate backup LSP.

When the backup ingress receives a PATH message with the INGRESS_PROTECTION object, the backup ingress examines the object to learn what traffic associated with the LSP. The backup ingress forwards the PATH message to the ingress node with the normal RSVP changes.

When the backup ingress receives a RESV message with the INGRESS_PROTECTION object, the backup ingress records an IMPLICIT-NULL label in the RRO. Then, the backup ingress forwards the RESV message to the ingress node, which is acting for the proxy ingress.

### 6.3.2. Backup Ingress Behavior in the On-Path Case

An LER as the backup ingress determines that it is on path if one of its addresses is a next hop of the primary ingress; for the Proxy-Ingress Method, the primary ingress is determined as not its next hop by checking the PATH message with the INGRESS_PROTECTION object received from the primary ingress. The LER on path MUST send the corresponding PATH messages without any INGRESS_PROTECTION object to its next hops. It creates a number of backup P2P LSPs or a backup P2MP LSP from itself to the other next hops (i.e., the next hops other than the backup ingress) of the primary ingress. The other next hops are from the Label-Routes subobject.

It also creates a forwarding entry, which sends/multicasts the traffic from the source to the next hops of the backup ingress along the protected LSP when the primary ingress fails. The traffic is described by the TRAFFIC_DESCRIPTOR.

After setting up all the backup P2P LSPs or the backup P2MP LSP, the backup ingress creates forwarding entry(s) for importing the traffic into the backup LSP(s) from the source when the primary ingress fails. Then, it MUST send the primary ingress a RESV message with an INGRESS_PROTECTION object. The object contains the state of the local protection, such as having the "local protection available" flag set to one, which indicates that the primary ingress is locally protected.

When the primary ingress fails, the backup ingress sends/multicasts the traffic from the source to its next hops along the protected LSP and imports the traffic into each of the backup P2P LSPs or to the

backup P2MP LSP transmitting the traffic to the other next hops of the primary ingress, where the traffic is merged into a protected LSP.

During the local repair, the backup ingress MUST continue to send the PATH messages to its next hops as before and keep the PATH message with the INGRESS_PROTECTION object received from the primary ingress and the RESV message with the INGRESS_PROTECTION object to be sent to the primary ingress. It MUST set the "local protection in use" flag in the RESV message.

### 6.3.3. Failure Detection and Refresh PATH Messages

As described in [RFC4090], it is necessary to refresh the PATH messages via the backup LSP(s). The backup ingress MUST wait to refresh the PATH messages until it can accurately detect that the ingress node has failed. An example of such an accurate detection would be that the IGP has no bidirectional links to the ingress node, or a BFD session to the primary ingress' loopback address has failed and stayed failed after the network has reconverged.

As described in Section 6.4.3 of [RFC4090], the backup ingress, acting as PLR, MUST modify and send any saved PATH messages associated with the primary LSP to the corresponding next hops through backup LSP(s). Any PATH message sent will not contain any INGRESS_PROTECTION objects. The RSVP_HOP object in the message contains an IP source address belonging to the backup ingress. The SENDER_TEMPLATE object has the Backup Ingress Address as its tunnel sender address.

## 6.4. Revertive Behavior

Upon a failure event in the (primary) ingress of a protected LSP, the protected LSP is locally repaired by the backup ingress. There are a couple of basic strategies for restoring the LSP to a full working path.

- Revert to Primary Ingress: When the primary ingress is restored, it resignals each of the LSPs that start from the primary ingress. The traffic for every LSP successfully resignaled is switched back to the primary ingress from the backup ingress.

- Global Repair by Backup Ingress: After determining that the primary ingress of an LSP has failed, the backup ingress computes a new optimal path, signals a new LSP along the new path, and switches the traffic to the new LSP.

#### 6.4.1. Revert to Primary Ingress

If "Revert to Primary Ingress" is desired for a protected LSP, the (primary) ingress of the LSP SHOULD resignal the LSP that starts from the primary ingress after the primary ingress restores. After the LSP is resignaled successfully, the traffic SHOULD be switched back to the primary ingress from the backup ingress on the source node and redirected into the LSP starting from the primary ingress.

The primary ingress can specify the "Revert to Ingress" control option in the INGRESS_PROTECTION object in the PATH messages to the backup ingress. After receiving the "Revert to Ingress" control option, the backup ingress MUST stop sending/refreshing PATH messages for the protected LSP.

#### 6.4.2. Global Repair by Backup Ingress

When the backup ingress has determined that the primary ingress of the protected LSP has failed (e.g., via the IGP), it can compute a new path and signal a new LSP along the new path so that it no longer relies upon local repair. To do this, the backup ingress MUST use the same tunnel sender address in the SENDER_TEMPLATE object and allocate an LSP ID different from the one of the old LSP as the LSP ID of the new LSP. This allows the new LSP to share resources with the old LSP. Alternately, the backup ingress can create a new LSP with no bandwidth reservation that duplicates the path(s) of the protected LSP, move traffic to the new LSP, delete the protected LSP, and then resignal the new LSP with bandwidth.

## 7. Security Considerations

In principle, this document does not introduce new security issues. The security considerations pertaining to [RFC4090], [RFC4875], [RFC2205], and [RFC3209] remain relevant.

## 8. Compatibility

This extension reuses and extends semantics and procedures defined in [RFC2205], [RFC3209], [RFC4090], and [RFC4875] to support ingress protection. The new object defined to indicate ingress protection has a Class Number of the form 0bbbbbbb. Per [RFC2205], a node not supporting this extension will not recognize the new Class Number and should respond with an "Unknown Object Class" error. The error message will propagate to the ingress, which can then take action to avoid the incompatible node as a backup ingress or may simply terminate the session.

## 9. IANA Considerations

This document has no IANA actions.

## 10. References

### 10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.

[RFC3936] Kompella, K. and J. Lang, "Procedures for Modifying the Resource reSerVation Protocol (RSVP)", BCP 96, RFC 3936, DOI 10.17487/RFC3936, October 2004, <https://www.rfc-editor.org/info/rfc3936>.

[RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <https://www.rfc-editor.org/info/rfc4090>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <https://www.rfc-editor.org/info/rfc4875>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

### 10.2. Informative References

[RFC6378] Weingarten, Y., Ed., Bryant, S., Osborne, E., Sprecher, N., and A. Fulignoli, Ed., "MPLS Transport Profile (MPLS-TP) Linear Protection", RFC 6378, DOI 10.17487/RFC6378, October 2011, <https://www.rfc-editor.org/info/rfc6378>.

## Acknowledgements

The authors would like to thank Nobo Akiya, Rahul Aggarwal, Eric Osborne, Ross Callon, Loa Andersson, Daniel King, Michael Yue, Alia Atlas, Olufemi Komolafe, Rob Rennison, Neil Harrison, Kannan Sampath, Gregory Mirsky, and Ronhazli Adam for their valuable comments and suggestions on this document.

## Contributors

The following people contributed significantly to the content of this document and should be considered coauthors:

Autumn Liu
Ciena
United States of America
Email: hliu@ciena.com

Zhenbin Li
Huawei Technologies
Email: zhenbin.li@huawei.com

Yimin Shen
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
United States of America
Email: yshen@juniper.net

Tarek Saad
Cisco Systems
Email: tsaad@cisco.com

Fengman Xu
Verizon
2400 N. Glenville Dr
Richardson, TX 75082
United States of America
Email: fengman.xu@verizon.com

The following people also contributed to the content of this document:

Ning So
Tata Communications
2613 Fairbourne Cir.
Plano, TX 75082
United States of America
Email: ningso01@gmail.com

Mehmet Toy
Verizon
United States of America
Email: mehmet.toy@verizon.com

Lei Liu
United States of America
Email: liulei.kddi@gmail.com

Renwei Li
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
United States of America
Email: renwei.li@huawei.com

Quintin Zhao
Huawei Technologies
Boston, MA
United States of America
Email: quintin.zhao@huawei.com

Boris Zhang
Telus Communications
200 Consilium Pl Floor 15
Toronto, ON M1H 3J3
Canada
Email: Boris.Zhang@telus.com

Markus Jork
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
United States of America
Email: mjork@juniper.net

Authors' Addresses

Huaimo Chen (editor)
Huawei Technologies
Boston, MA
United States of America

Email: huaimo.chen@huawei.com

Raveendra Torvi (editor)
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
United States of America

Email: rtorvi@juniper.net