Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2016

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

October 19, 2015

Extensions to RSVP-TE for LSP Ingress Local Protection
draft-ietf-teas-rsvp-ingress-protection-04.txt

Abstract

This document describes extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for locally protecting the ingress node of a Traffic Engineered (TE) Label Switched Path (LSP), which is a Point-to-Point (P2P) LSP or a Point-to-Multipoint (P2MP) LSP.

Table of Contents

1. Co-authors

Ning So, Autumn Liu, Alia Atlas, Yimin Shen, Tarek Saad, Fengman Xu, Mehmet Toy, Lei Liu

2. Introduction

For a MPLS LSP it is important to have a fast-reroute method for protecting its ingress node and transit nodes. Protecting an ingress is not covered either in the fast-reroute method defined in [RFC4090] or in the P2MP fast-reroute extensions to fast-reroute in [RFC4875].

An alternate approach to local protection (fast-reroute) is to use global protection and set up a secondary backup LSP (whether P2MP or P2P) from a backup ingress to the egresses. The main disadvantage of this is that the backup LSP may reserve additional network bandwidth.

This specification defines a simple extension to RSVP-TE for local protection of the ingress node of a P2MP or P2P LSP.

2.1. An Example of Ingress Local Protection

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 R1 to three egresses: L1, L2 and L3. The backup LSP is from backup ingress Ra to the next hops R2 and R4 of ingress R1.

```
                  [R2]******[R3]*****[L1]
                 *   |                             **** Primary LSP
                *    |                             ---- Backup LSP
               *     /                             .... BFD Session
              *     /                                $  Link
        ....[R1]*******[R4]****[R5]*****[L2]        $
        :  $  $    /     /       *                 $
        : $   $   /     /         *
       [S]    $  /     /           *
          $   $ /     /             *
           $  $/     /               *
            [Ra]----[Rb]              [L3]
```

Figure 1: Backup P2MP LSP for Locally Protecting Ingress

In normal operations, source S sends the traffic to primary ingress R1. R1 imports the traffic into the primary LSP.

When source S detects the failure of R1, it switches the traffic to backup ingress Ra, which imports the traffic from S into the backup LSP to R1’s next hops R2 and R4, where the traffic is merged into the primary LSP, and then sent to egresses L1, L2 and L3. Source S detects the failure of R1 and switches the traffic within 10s of ms.

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.

2.2. Ingress Local Protection with FRR

Through using the ingress local protection and the FRR, we can locally protect the ingress, all the links and the transit nodes of an LSP. The traffic switchover time is within 10s of ms whenever the ingress, any of the links and the transit nodes of the LSP fails.

The ingress node of the LSP can be locally protected through using the ingress local protection. All the links and all the transit nodes of the LSP can be locally protected through using the FRR.

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. The backup ingress is ready to import the traffic from the source into the backup LSP after the backup LSP 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, where the traffic is merged into the primary LSP.

For a P2P LSP, after the primary ingress fails, the backup ingress MUST use a method to reliably detect the failure of the primary ingress before the PATH message for the LSP expires at the next hop of the primary ingress. After reliably detecting the failure, the backup ingress sends/refreshes the PATH message to the next hop through the backup LSP as needed.

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.

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.

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 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, we call it is off-path. If a backup ingress is a next-hop of the primary ingress of the LSP, we call it is on-path. 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, and 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. It is backward compatible.

5.1. INGRESS_PROTECTION Object

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. It has the following format:

```
   Class-Num = TBD      C-Type = 1 for INGRESS_PROTECTION_IPv4
                        C-Type = 2 for INGRESS_PROTECTION_IPv6
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         Length (bytes)        |    Class-Num  |    C-Type     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |        Reserved (zero)        |      Flags    |    Options    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~                          (Subobjects)                         ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

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

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

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

- o Ingress local protection available: The backup ingress sets 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.

- o Ingress local protection in use: The backup ingress sets this flag when it detects a failure in the primary ingress. The backup ingress keeps it and does not send it to the primary ingress since the primary ingress is down.

- o Bandwidth protection: The backup ingress sets this flag if the backup LSPs guarantee to provide 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.

- o Revert to Ingress: The primary ingress sets this option indicating that the traffic for the primary LSP successfully re-signaled will be switched back to the primary ingress from the backup ingress when the primary ingress is restored.

- o P2MP Backup: This option is set to ask for the backup ingress to use P2MP backup LSP to protect the primary ingress. Note that one spare bit of the flags in the FAST-REROUTE object can be used to indicate whether P2MP or P2P backup LSP is desired for protecting an ingress and transit node.

The INGRESS_PROTECTION object may contain some sub objects below.

5.1.1. 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 may have a Backup Ingress IPv4 Address sub object containing an IPv4 address belonging to the backup ingress. The Type of the sub object is TBD1 (the exact number to be assigned by IANA), and the body of the sub object is given below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Backup ingress IPv4 address (4 bytes)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

5.1.2. 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 may have a Backup Ingress IPv6 Address sub object containing an IPv6 address belonging to the backup ingress. The Type of the sub object is TBD2, the body of the sub object is given below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Backup ingress IPv6 address (16 bytes)            |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

5.1.3. Subobject: Ingress IPv4 Address

The INGRESS_PROTECTION object may have an Ingress IPv4 Address sub object containing an IPv4 address belonging to the primary ingress. The Type of the sub object is TBD3. The sub object has the following body:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Ingress IPv4 address (4 bytes)                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ingress IPv4 address: An IPv4 host address of ingress

5.1.4. Subobject: Ingress IPv6 Address

The INGRESS_PROTECTION object may have an Ingress IPv6 Address sub object containing an IPv6 address belonging to the primary ingress. The Type of the sub object is TBD4. The sub object has the following body:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Ingress IPv6 address (16 bytes)                |
~                                                               ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Ingress IPv6 address: An IPv6 host address of ingress

5.1.5. Subobject: Traffic Descriptor

The INGRESS_PROTECTION object may have a Traffic Descriptor sub object describing the traffic to be mapped to the backup LSP on the backup ingress for locally protecting the primary ingress. The Type of the sub object is TBD5, TBD6, TBD7 or TBD8 for Interface, IPv4 Prefix, IPv6 Prefix or Application Identifier respectively. The sub object has the following body:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Traffic Element 1                      |
~                                                               ~
|                        Traffic Element n                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Traffic Descriptor sub object may contain multiple Traffic Elements of same type as follows:

- o Interface Traffic (Type TBD5): Each of the Traffic Elements is a 32 bit index of an interface, from which the traffic is imported into the backup LSP.

- o IPv4 Prefix Traffic (Type TBD6): Each of the Traffic Elements is an IPv4 prefix, containing an 8-bit prefix length followed by an IPv4 address prefix, whose length, in bits, is specified by the prefix length, padded to a byte boundary.

- o IPv6 Prefix Traffic (Type TBD7): 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, padded to a byte boundary.

- o Application Traffic (Type TBD8): Each of the Traffic Elements is a 32 bit identifier of an application, from which the traffic is imported into the backup LSP.

5.1.6. Subobject: Label-Routes

The INGRESS_PROTECTION object in a PATH message from the primary ingress to the backup ingress will have a Label-Routes sub object containing the labels and routes that the next hops of the ingress use. The Type of the sub object is TBD9. The sub object has the following body:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                           Subobjects                          ~
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Subobjects in the 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

There are four parts of ingress protection: 1) setting up the necessary backup LSP forwarding state; 2) identifying the failure and providing the fast repair (as discussed in Sections 3 and 4); 3) maintaining the RSVP-TE control plane state until a global repair is done; and 4) performing the global repair(see Section 6.3).

6.1. Ingress Behavior

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

- o Backup Ingress Address: The primary ingress MUST know an IP address for it to be included in the INGRESS_PROTECTION object.

- o 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 sub-object 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.

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

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 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. Thus, the backup ingress has access to all the PATH messages needed for modification to refresh control-plane state after a failure.

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

1. Select a PATH message.

2. If the backup ingress is off-path, then send it a PATH message with the content from the selected PATH message 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 sub-object, the Backup Ingress Address sub-object and the Label-Routes sub-object. The options is set to indicate whether a Backup P2MP LSP is desired. The Label-Routes sub-object contains the next-hops of the ingress and their labels.

3. For each of the other PATH messages, send the backup ingress a PATH message with the content copied from the message and an empty INGRESS_PROTECTION object, which is an object without any Traffic-Descriptor sub-object.

6.2. Backup Ingress Behavior

An 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 sub-object 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.

6.2.1. Backup Ingress Behavior in Off-path Case

The backup ingress considers itself as a PLR and the primary ingress as its next hop and provides a local protection for the primary ingress. It behaves very similarly to a PLR providing fast-reroute where the primary ingress is considered as 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, facility backup if the "facility backup desired" is set, and backup paths that support the desired bandwidth, and administrative-colors 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". [Editor Note: It is possible to indicate the number or which are unprotected via a sub-object if desired.]

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, indicating 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 message 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 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.

When the backup ingress receives a PATH message with an 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 sub-object 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 to the PATH message received from the primary ingress. 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.

6.2.2. Backup Ingress Behavior in 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. 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 sub object.

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 the forwarding entry is created, all the backup P2P LSPs or the backup P2MP LSP is up and associated with the protected LSP, the backup ingress MUST send the primary ingress the RESV message with the INGRESS_PROTECTION object containing the state of the local protection such as "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 the backup P2MP LSP transmitting the traffic to the other next hops of the primary ingress, where the traffic is merged into protected LSP.

During the local repair, the backup ingress MUST continue to send the PATH messages to its next hops as before, 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.

6.2.3. 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 bi-directional links to the ingress node and the last change was long enough in the past that changes should have been received (i.e., an IGP network convergence time or approximately 2-3 seconds) or a BFD session to the primary ingress' loopback address has failed and stayed failed after the network has reconverged.

As described in [RFC4090 Section 6.4.3], 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 object. 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.

6.3. 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 re-signals each of the LSPs that start from the primary ingress. The traffic for every LSP successfully re-signaled 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.

6.3.1. Revert to Primary Ingress

If "Revert to Primary Ingress" is desired for a protected LSP, the (primary) ingress of the LSP SHOULD re-signal the LSP that starts from the primary ingress after the primary ingress restores. After the LSP is re-signaled 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.

6.3.2. 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 a 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. In addition, if the Ingress recovers, the Backup Ingress SHOULD send it RESVs with the INGRESS_PROTECTION object where the "Revert to Ingress" is specified. The Ingress can learn from the RESVs what to signal. The Backup Ingress can reoptimize the new LSP as necessary until the Ingress recovers. 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 RFC 4090, RFC 4875 and other RSVP protocols remain relevant.

8. IANA Considerations

IANA is requested to administer the assignment of new values defined in this document and summarized in this section.

8.1. A New Class Number

IANA maintains a registry called "Class Names, Class Numbers, and Class Types" under "Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Parameters". IANA is requested to assign a new Class Number for new object INGRESS_PROTECTION as follows:

Class Names	Class Numbers	Class Types
INGRESS_PROTECTION	TBD (>192)	1: INGRESS_PROTECTION_IPv4
		2: INGRESS_PROTECTION_IPv6

IANA is requested to assign Types for new TLVs in the new objects as follows:

Type	Name	Allowed in
1	BACKUP_INGRESS_IPv4_ADDRESS	INGRESS_PROTECTION_IPv4
2	BACKUP_INGRESS_IPv6_ADDRESS	INGRESS_PROTECTION_IPv6
3	INGRESS_IPv4_ADDRESS	INGRESS_PROTECTION_IPv4
4	INGRESS_IPv6_ADDRESS	INGRESS_PROTECTION_IPv6
5	TRAFFIC_DESCRIPTOR_INTERFACE	INGRESS_PROTECTION
6	TRAFFIC_DESCRIPTOR_IPv4_PREFIX	INGRESS_PROTECTION_IPv4
7	TRAFFIC_DESCRIPTOR_IPv6_PREFIX	INGRESS_PROTECTION_IPv6
8	TRAFFIC_DESCRIPTOR_APPLICATION	INGRESS_PROTECTION
9	LabeL_Routes	INGRESS_PROTECTION

9. Contributors

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

Quintin Zhao
Huawei Technologies
Boston, MA
USA
Email: quintin.zhao@huawei.com

Zhenbin Li
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA
Email: zhenbin.li@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
USA
Email: mjork@juniper.net

10. Acknowledgement

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

11. 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, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <http://www.rfc-editor.org/info/rfc3031>.

[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, <http://www.rfc-editor.org/info/rfc3209>.

[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, <http://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, <http://www.rfc-editor.org/info/rfc4875>.

Appendix A. Authors' Addresses

Huaimo Chen
Huawei Technologies
Boston, MA
USA
Email: huaimo.chen@huawei.com

Raveendra Torvi
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA
Email: rtorvi@juniper.net

Ning So
Tata Communications
2613 Fairbourne Cir.
Plano, TX 75082
USA
Email: ningso01@gmail.com

Autumn Liu
Ericsson
300 Holger Way
San Jose, CA 95134
USA
Email: autumn.liu@ericsson.com

Alia Atlas
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA
Email: akatlas@juniper.net

Yimin Shen
Juniper Networks
10 Technology Park Drive
Westford, MA 01886
USA
Email: yshen@juniper.net

Tarek Saad
Cisco Systems
Email: tsaad@cisco.com

Fengman Xu
Verizon
2400 N. Glenville Dr
Richardson, TX 75082
USA
Email: fengman.xu@verizon.com

Mehmet Toy
Comcast
1800 Bishops Gate Blvd.
Mount Laurel, NJ 08054
USA
Email: mehmet_toy@cable.comcast.com

Lei Liu
UC Davis
USA
Email: liulei.kddi@gmail.com