Network Working Group Louis Berger (Movaz Networks)
Internet Draft Igor Bryskin (Movaz Networks)
Expiration Date: August 2004 Dimitri Papadimitriou (Alcatel)
Adrian Farrel (Old Dog Consulting)

February 2004

GMPLS Based Segment Recovery

draft-berger-ccamp-gmpls-segment-recovery-00.txt

Status of this Memo

Abstract

This document describes protocol specific procedures for GMPLS (Generalized Multi-Protocol Label Switching) RSVP-TE (Resource ReserVation Protocol - Traffic Engineering) signaling extensions to support LSP segment protection and restoration. These extensions are intended to be compliment and be consistent with the Extensions for End-to-End GMPLS-based Recovery. Implications and interactions with Fast Reroute are also addressed.

Contents

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

In addition, the reader is assumed to be familiar with the terminology used in [RFC3209], [RFC3471], [RFC3473] as well as [TERM], [FUNCT], [E2E-RECOVERY] and [FRR].

1. Introduction

[TERM] covers multiple types of protection, including end-to-end and segment based approaches. [E2E-RECOVERY], RSVP-TE Extensions in support of End-to-End GMPLS-based Recovery, defines a set of extensions to support multiple types of recovery. The supported types include 1+1 unidirectional/ 1+1 bi-directional protection, LSP protection with extra-traffic (including 1:N protection with extra-traffic), pre-planned LSP re- routing without extra-traffic (including shared mesh), and full LSP re-routing. In all cases, the recovery is provided on an end-to-end basis, i.e., the ingress and egress nodes of both the protected and the protecting LSP are the same.

[FRR] provides a form of segment recovery for packet MPLS-TE networks. Two methods of Fast Reroute are defined in [FRR]. The one-to-one backup method creates detour LSPs for each protected LSP at each potential point of local repair. The facility backup method creates a bypass tunnel to protect a potential failure point which is shared by multiple LSPs and uses label stacking. Neither approach supports the full set of recovery types supported by [E2E-RECOVERY]. Additionally, the facility backup method is not applicable to most non-PSC (packet) switching technologies.

The extensions defined in this document allow for support of the full set of recovery types supported by [E2E-RECOVERY] on a segment, or portion of the LSP, basis. The extensions allow (a) the signaling of desired LSP segment protection type, (b) upstream nodes to optionally identify where segment protection starts and stops, and (c) to also optionally identify hops used on protection segments. These extensions are intended to be compatible with, and in some cases used with Fast Reroute.

2. Segment Recovery

Segment recovery is used to provide protection and restoration over a portion of an end-to-end LSP. Such segment protection and restoration is useful to protect against a span failure, a node failure, or failure over a particularly portion of a network used by an LSP.

Consider the following topology:

```
A---B---C---D---E---F
         \     /
          G---I
```

In this topology, end-to-end protection and recovery is not possible for an LSP going between node A and node F, but it is possible to protect/recover a portion of the LSP. Specifically, if the LSP uses a working path of [A,B,C,D,E,F] then a protection or restoration LSP can be established along the path [C,G,I,E]. This LSP protects against failures on spans {C,D} and {D,E} as well as a failure of node D. This form of protection/restoration is referred to as Segment Protection and Segment Restoration, or Segment Recovery collectively. The LSP providing the protection or restoration is referred to as a segment protection LSP or a segment restoration LSP. The term segment recovery LSP is used to cover either a segment protection LSP or a segment restoration LSP. The term branch node is used to refer to a node that initiates a recovery LSP, e.g., node C in the figure shown above. This is equivalent to the point of local repair (PLR) used in [FRR]. As with [FRR], the term merge node is used to refer to a node that terminates a recovery LSP, e.g., node E in the figure shown above.

Segment protection or restoration is signaled using a working LSP and one or more segment recovery LSPs. Each segment recovery LSP is signaled as an independent LSP. Specifically, the Sender_Template object uses the IP address of the node originating the recovery path, e.g., node C in the topology shown above, and the Session object contains the IP address of the node terminating the recovery path, e.g., node E shown above. There is no specific requirement on LSP ID value, Tunnel ID and Extended Tunnel ID. Values for these fields are selected normally, including consideration for make-before-break. Intermediate nodes follow standard signaling procedures when processing segment recovery LSPs. A segment recovery LSP may be protected itself using segment or end-to-end protection/restoration. Note, in PSC environments it may be desirable to construct the Sender_Template and Session objects per [FRR].

When [FRR] isn't being used, the association between segment recovery LSPs with other LSPs is indicated using the Association object

defined in [E2E-RECOVERY]. The Association object is used to associate recovery LSPs with the LSP they are protecting. Working and protecting LSPs, as well as primary and secondary LSPs, are identified using LSP Status as described in [E2E-RECOVERY]. The 0-bit in the segment flags portion of the Protection object is used to identify when a recovery LSP is carrying the normal (active) traffic.

An upstream node can permit downstream nodes to dynamically identify branch and merge points by setting the desired LSP segment protection bits in the Protection object. These bits are defined below.

Optionally, an upstream node, usually the ingress node, can identify the endpoints of a segment recovery LSP. This is accomplished using a new object. This object uses the same format as an ERO and is referred to as a Secondary Explicit Route object or SERO, see section 4.1. SEROs also support a new subobject to indicate the type of protection or restoration to be provided. At a minimum an SERO will indicate a recovery LSP's initiator, protection/restoration type and terminator. Standard ERO semantics, see [RFC3209], can optionally be used within and SERO to explicitly control the recovery LSP. A Secondary Record Route object or SRRO is defined for recording the path of a segment recovery LSP, see section 5.

SEROs are carried between the node creating the SERO, typically the ingress, and the node initiating a recovery LSP. The node initiating a recovery LSP uses the SERO to create the ERO for the recovery LSP. At this (branch) node, all local objects are removed, and the new protection subobject is used to create the Protection object for the recovery LSP. SRROs are carried in Path messages between the node terminating a recovery LSP, the merge node, and the egress. SRROs are used in Resv messages between a branch node and the ingress. The merge node of a recovery LSP creates an SRRO by copying the RRO from the Path message of the associated recovery LSP into a new SRRO object. Any SRROs present in the recovery LSP's Path message are also copied. The branch node of a recovery LSP creates an SRRO by copying the RRO from the Resv message of associated recovery LSP into a new SRRO object. Any SRROs present in the recovery LSP's Resv message are also copied.

Notify request processing is also impacted by LSP segment recovery. Per [RFC3473], only one Notify Request object is meaningful and should be propagated. Additional Notify Request objects are used to identify recovery LSP branch nodes.

2.1. Segment Protection

Three approaches of end-to-end protection are defined in [E2E-RECOVERY]: 1+1 Unidirectional Protection, see Section 5; 1+1 Bi-directional Protection, see Section 6: and 1:1 Protection With Extra Traffic, see Section 7. The segment protection forms of these protection approaches all operate much like their end-to-end counterparts. Each behaves just like there end-to-end counterparts, with the exception that the protection LSP protects only a portion of the working LSP. The type of protection to be used on a segment protection LSP is indicated, to the protection LSP's ingress, using the protection SERO subobject defined in Section 4.1.

The switch-over processing for segment 1+1 Bi-directional protection and 1:1 Protection With Extra Traffic follows the same procedures as end-to-end protection forms, see Section 6.2 and Section 7.2 for details.

2.2. Segment Re-routing and Restoration

Three re-routing and restoration approaches are defined [E2E-RECOVERY]: Re-routing without Extra-Traffic, see Section 8; Shared-Mesh Restoration, see Section 9; (Full) LSP Re-routing, see Section 11. As with protection, these approaches are supported on a segment basis. The segment forms of re-routing and restoration operate exactly like their end-to-end counterparts, with the exception that the restoration LSP recovers only a portion of the working LSP. The type of re-routing or restoration to be used on a segment restoration LSP is indicated, to the restoration LSP's ingress, using the new protection SERO subobject.

3. Association Object

The Association object is used association of segment protection LSPs when [FRR] isn't being used. The Association object is defined in [E2E-RECOVERY]. In this document we define a new type to support make before break, formats and procedures defined in [E2E-RECOVERY] are not otherwise modified.

3.1. Format

Association Type: 16 bits

Value	Type
2	Resource Sharing (R)

See [E2E-RECOVERY] for the definition of other fields and other values.

3.2. Procedures

The Association object is used to associate different LSPs with each other. In the protection and restoration context, the object is used to associate a recovery LSP with the LSP it is protecting. It is also used to support resource sharing during make-before-break. This object MUST NOT be used when association is made according to the methods defined in [FRR].

3.2.1. Resource Sharing Association Type Processing

The Association object with an Association Type with the value Resource Sharing is used to enable resource sharing during make-before-break. Resource sharing during make-before-break is defined in [RFC3209]. The defined support only works with LSPs that share the same LSP egress. With the introduction of segment recovery LSPs, it is now possible for an LSP end-point to change during make-before-break.

A node includes an Association object with a Resource Sharing Association Type in outgoing an Path message when it wishes to indicate resource sharing across an associated set of LSPs. The Association Source is set to the branch node's router address. The Association ID MUST be set to a value that uniquely identifies the association of LSPs. This MAY be set to the upstream LSP's LSP ID. Once included, an Association object with a Resource Sharing Association Type SHOULD NOT be removed from the Path messages associated with an LSP.

When a node is branching an LSP and the associated upstream Path messages is received with an Association object with a Resource Sharing type, the branch node inserts a new Association object with a Resource Sharing type in the Path message of the new LSP. The Association Source is set to the branch node's router address. The

Association ID MUST be set to a value that uniquely identifies the association of LSPs. This MAY be set to the recovery LSP's LSP ID.

Any node processing a Path message for which it does not have matching state, and which contains a Association object with a Resource Sharing type, examines existing LSPs for matching Association Type, Association Source and Association ID values. If any match is found, then [RFC3209] style resource sharing SHOULD be provided between the new and old LSPs. See [RFC3209] for additional details.

4. Explicit Control of LSP Segment Recovery

Secondary Explicit Route objects, or SEROs, may be used to indicate the branch and merge nodes of recovery LSPs. They may also provide additional information that is to be carried in a recovery LSP's ERO. When upstream control of branch and merge nodes are not desired, SEROs are not used.

4.1. Secondary Explicit Route Object Format

The format of a SECONDARY_EXPLICIT_ROUTE object is the same as an EXPLICIT_ROUTE object. This includes the definition of subobjects defined for EXPLICIT_ROUTE object. The class of the SECONDARY_EXPLICIT_ROUTE object is TBA (of form 11bbbbbb).

4.1.1. Protection Subobject

The protection subobject is not valid for use with the Explicit and Record Route objects and MUST NOT be included in those objects.

The format of the Protection Subobject is defined as follows:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |L|    Type     |     Length    |    Reserved   |  C-Type       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  PROTECTION Object Contents                   |
   |                              ...                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L-bit

This is defined in [RFC3209] and MUST be set to zero for protection subobjects.

Type

37 Protection

C-Type

The C-Type of the included Protection object.

PROTECTION Object Contents

The contents of the Protection object with the format matching the indicated C-Type, excluding the object header.

4.2. Explicit Control Procedures

SEROs are carried in Path messages and indicate at which node a recovery LSP is to be initiated relative to the LSP carrying the SERO. More than one SERO MAY be present in a Path message.

To indicate the branch and merge nodes of a recovery LSPs, an SERO is created and added to the Path message of the LSP being recovered. The decision to create and insert an SERO is a local matter and outside the scope of this document.

An SERO SHOULD contain at least three subobjects. The first subobject MUST indicate the node that is to originate the recovery LSP, i.e. the segment branch node. The address used SHOULD also be listed in the ERO or another SERO. This ensures that the branch node is along the LSP path. The second subobject SHOULD be a protection subobject and should indicate the protection or restoration to be provided by the recovery LSP. When the protection subobject is present, the LSP Segment Recovery Flags in the Protection subobject MUST be ignored. The final subobject in the SERO MUST be the merge node of the recovery LSP, and MAY have the L-bit set. Standard ERO subobjects MAY be inserted between the protection subobject and the final subobject. These subobjects MAY be loose or strict.

A node receiving a Path message containing one or more SEROs SHOULD examine each SERO to see if it indicates a local branch point. This determination is made by examining the first object of each SERO and seeing if the address indicated in the subobject can be associated with the local node. If any of indicated addresses are associated

with the local node, then the local node is a branch node. If the local node is not a branch node, all received SEROs MUST be transmitted, without modification, in the corresponding outgoing Path message.

At a branch node, the SERO together with the Path message of LSP being recovered provides the information to create the recovery LSP. If the processing node is unable to support the requested branch, a PathErr message SHOULD sent for the LSP being being protected, and normal processing of the LSP continues. The PathErr message SHOULD indicate an error of "TBD" and the Path_State_Removed flag MUST NOT be set. If no error is generated then a recovery LSP is created.

The Path message for the recovery LSP is created at the branch node by cloning the objects carried in the incoming Path message of the LSP being protected. Certain objects are replaced or modified in the recovery LSP's outgoing Path message. The Sender_template MUST be updated to use an address on the local node, and the LSP ID MUST be updated to ensure uniqueness. The Session object MUST be updated to use the address indicated in the final subobject of the SERO as the tunnel endpoint, the tunnel ID MAY be updated, and the extended tunnel ID MUST be set to the local node. The Protection object is replaced with the contents of the matching SERO subobject, when present. Any RROs and EROs present in the incoming Path message MUST NOT be included in the recovery LSP. A new ERO MUST be included, with the contents of the SERO that indicated a local branch. As with all EROs, no local information (local address and any protection subobjects) is carried in the ERO carried in the recovery LSP's outgoing Path message. The SERO that indicated a local branch MUST be omitted from the recovery LSP's outgoing Path message. Note, by default all other received SEROs are passed in the recovery LSP's outgoing Path message. SEROs MAY be omitted, from the recovery LSP's outgoing Path message as well as the outgoing Path message for the LSP being protected when the SERO does not relate to the outgoing path message.

The resulting Path message is used to create the recovery LSP. From this point on, Standard Path message processing is used in processing the resulting Path message.

4.2.1. Resv Message Processing

Branch nodes will process Resv messages for both recovery LSPs and LSPs being protected. Resv messages are propagated upstream of branch nodes only after a Resv message is received for the protected LSP. Resv messages on recovery LSPs will typically not trigger transmission of upstream Resv messages (for the LSP being protected).

Exceptions to this include when RROs/SRROs are being collected and during certain Admin Status object processing. See below for more information on related processing.

4.2.2. Branch Failure Handling

During setup and during normal operation, PathErr messages may be received at a branch node. In all cases, a received PathErr message is first processed per standard processing rules. When the PathErr messages is not on a recovery LSP and the Path_State_Removed flag is set, then any recovery LSPs associated with the LSP MUST also be torn down.

If the PathErr messages is on a recovery LSP, receipt of the PathErr message SHOULD trigger the generation of a PathErr message upstream on the associated LSP. This outgoing (upstream) PathErr message SHOULD be sent with the Path_State_Removed flag cleared (0) as only the recovery LSP is impacted. However, if a branch node sends a PathErr message with the Path_State_Removed flag set (1), which is not recommended, the node MUST send a PathTear message downstream on all other branches.

Additionally, an outgoing PathErr message MUST include any SEROs carried in a received PathErr message. If no SERO is present in a received PathErr message, then an SERO that matches the errored LSP MUST be added to the outgoing PathErr message.

4.2.3. Admin Status Change

In general, objects in a recovery LSP are created based on the corresponding objects in the LSP being protected. The Admin Status object is created the same way, but it also requires some special coordination at branch nodes. Specifically, in addition to normal processing, a branch node that receives an Admin Status object in a Path message also MUST relay the Admin Status object in a Path on every recovery LSP. All Path messages MAY be concurrently sent downstream.

Downstream nodes process the change in the Admin Status object per [RFC3473], including generation of Resv messages. When the most recently received upstream Admin Status object had the R bit set, branch nodes wait for a Resv message with a matching Admin Status object to be received on all branches before relaying a corresponding Resv message upstream.

4.2.4. Recovery LSP Tear Down

Recovery LSP removal is follows standard the standard procedures defined in [RFC3209] and [RFC3473]. This includes without and with setting the administrative status.

4.2.4.1. Tear Down Without Admin Status Change

The node initiating the tear down originates a PathTear message. Each node that receives a PathTear message process the PathTear message per standard processing, see [RFC3209] and [RFC2205], and also relays a PathTear on every recovery LSP. All PathTear messages (received from upstream and locally originated) may be concurrently sent downstream.

4.2.4.2. Tear Down With Admin Status Change

Per [RFC3473], the ingress node originates a Path message with the D and R bits set in the Admin Status object. The admin status change procedure defined above, see Section 4.2.3, MUST then be followed. Once the ingress receives all expected Resv messages MUST follow the tear down procedure described in the Section 4.2.4.

4.3. Tear Down From Non-Ingress Nodes

As with any LSP, any node along a recovery LSP may initiate removal of the recovery LSP. To do this, the node initiating the tear down sends a PathErr message with the appropriate Error Code and the Path_State_Removed flag cleared (0) toward the LSP ingress. As described above, the recovery LSP ingress will propagate the error to the LSP ingress which can then signal the removal of the recovery LSP.

It is also possible for the node initiating the tear down to remove a Recovery LSP in a non-graceful manner. In this case, the initiator sends a PathTear message downstream and a PathErr message with Error Code TBD and the Path_State_Removed flag set (1) toward the LSP ingress node. This manner of non-ingress node tear down is NOT RECOMMENDED as it can result in the removal of the LSP being protected in some case.

4.3.1. Modified Notify Request Object Processing

When a node is branching a recovery LSP, it SHOULD include a single Notify Request object on the recovery LSP. The notify node address MUST be set to the router address of the branch node. Normal notification procedures are then followed for the recovery LSP. Under local policy control, a node issuing a Notify message MAY also send a Notify message to the Notify Node Address indicated in the last, or any other, Notify_Request object received.

A branch node SHOULD also add a Notify Request object to the LSP being protected. The notify node address is set to the address used in the sender template of the recovery LSP. A locally added Notify Request object MUST be listed first in the outgoing message, any received Notify Request object MUST then be listed in the message in the order that they were received.

Recovery LSP merge nodes remove the object added by the recovery branch node from outgoing Path messages for the LSP being protected. This is done by removing the Notify Request object that matches the source address of the recovery LSP. Note, to cover certain backwards compatibility scenarios the Notify Request object SHOULD NOT be removed if it is the sole Notify Request object.

Note this requires the following change to [RFC3473], Section 4.2.1:
- old text:
  If a message contains multiple Notify_Request objects, only the first object is meaningful. Subsequent Notify_Request objects MAY be ignored and SHOULD NOT be propagated.

- new text:
  If a message contains multiple Notify_Request objects, only the first object used is in notification. Subsequent Notify_Request objects MUST be propagated in the order received.

4.3.2. Modified Notify and Error Message Processing

Branch nodes MUST support the following modification to Notify message processing. When a branch node receives notification of an LSP failure and it is unable to recover from that failure, it MUST notify the node indicated in the first Notify_Request object received in the Path message associated with the LSP.

5. Secondary Record Route Objects

Secondary Record Route objects, or SRROs, are used to record the path used by recovery LSPs.

5.1. Format

The format of a SECONDARY_RECORD_ROUTE object is the same as an RECORD_ROUTE object, Class number 21. This includes the definition of subobjects defined for RECORD_ROUTE object. The class of the SECONDARY_RECORD_ROUTE object is TBA (of form 11bbbbbb).

The protection subobject defined in [E2E-RECOVERY] can also be used in SECONDARY_RECORD_ROUTE objects.

5.2. Path Processing

SRROs may be carried in Path messages and indicate the presence of upstream recovery LSPs. More than one SRRO MAY be add and present in a Path message.

Any received SRRO, MUST be transmitted by transit nodes, without modification, in the corresponding outgoing Path message.

SRROs are inserted in Path messages by recovery LSP merge nodes. The SRRO is created by copying the contents of an RRO received the recovery LSP into a new SRRO object. This SRRO is added to the outgoing Path message of the recovered LSP. Note multiple SRROs may be present. The collection of SRROs is controlled via the segment-recording-desired flag in the SESSION_ATTRIBUTE object. This flag MAY be set even when SEROs are not used.

5.3. Resv Processing

SRROs may be carried in Resv messages and indicate the presence of downstream recovery LSPs. More than one SRRO MAY be add and present in a Resv message.

Any received SRRO, MUST be transmitted by transit nodes, without modification, in the corresponding outgoing Resv message. When Resv messages are merged, the resulting merged Resv should contain all SRROs received in downstream Resv messages.

SRROs are inserted in Resv messages by branch nodes of recovery LSPs. The SRRO is created with the first two objects being the local node

address followed by a protection subobject with the contents of the recovery LSP's Protection object. The remainder of the SRRO is created by copying the contents of the RRO received the recovery LSP. This SRRO is added to the outgoing Resv message of the recovered LSP. Again, multiple SRROs may be present.

6. Dynamic Control of LSP Segment Recovery

Dynamic identification of branch and merge nodes is supported via the LSP Segment Recovery Flags carried in the Protection object. The LSP Segment Recovery Flags are carried within one of Reserved fields defined in the Protection object defined in [E2E-RECOVERY]. LSP Segment Recovery Flags are used to indicate when LSP segment recovery is desired. When these bits are set branch and merge nodes are dynamically identified.

Note, the procedures defined in this section parallel the explicit control procedures defined above in Section 4.2. The primary difference is in creation of a recovery LSP's ERO.

6.1. Modified Protection Object Format

LSP Segment Recovery Flags are carried in the Protection object C-Type defined in [E2E-RECOVERY]. The format of the flags are:

```
    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             | Class-Num(37) | C-Type (TBA)  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |S|P|N|O| Reserved  | LSP Flags |     Reserved      | Link Flags|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |I|     Reserved    | Seg.Flags |        Reserved               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In-Place (I): 1 bit

When set (1) indicates that the desired segment recovery type indicated in the LSP Segment Recovery Flag is already in place for the associated LSP.

(LSP) Segment (Recovery) Flags: 6 bits

This field is used to indicate when an upstream node desires LSP Segment recovery to be dynamically initiated where possible. The values used in this field are identical to the

values defined for LSP Flags, see [E2E-RECOVERY].

See [E2E-RECOVERY] for the definition of other fields.

6.2. Dynamic Control Procedures

LSP Segment Recovery Flags are set to indicate that LSP segment recovery is desired for the LSP being signaled. The type of recovery desired is indicated by the flags. The decision to set the LSP Segment Recovery Flags is a local matter and outside the scope of this document. A value of zero (0) means that no dynamic identification of segment recovery branch nodes are needed for the associated LSP. When the In-Place bit is set, it means that the desired type of recovery is already in place for that particular LSP.

A transit node receiving a Path message containing a Protection object with a non-zero LSP Segment Recovery Flags value and the In-Place bit clear (0) SHOULD consider if it can support the indicated recovery type and if it can identify an appropriate merge node for a recovery LSP. Dynamic identification MUST NOT be done when the processing node is identified as a branch node in an SERO. If a node is unable to provide the indicated recovery type or identify a merge node, the Path message MUST be processed normally and the LSP Segment Recovery Flags MUST NOT be modified.

When a node dynamically identifies itself as a branch node and identifies the merge node for the type of recovery indicated in the LSP Segment Recovery Flags, it attempts to setup a recovery LSP. The dynamically identified information, together with the Path message of LSP being recovered provides the information to create the recovery LSP.

The Path message for the recovery LSP is created at the branch node by cloning the objects carried in the incoming Path message of the LSP being protected. Certain objects are replaced or modified in the recovery LSP's outgoing Path message. The Sender_template MUST be updated to use an address on the local node, and the LSP ID MUST be updated to ensure uniqueness. The Session object MUST be updated to use the address of the dynamically identified merge node as the tunnel endpoint, the tunnel ID MAY be updated, and the extended tunnel ID MUST be set to the local node. The Protection object is updated with the In-Place bit set (1). Any RROs and EROs present in the incoming Path message MUST NOT be included in the recovery LSP. A new ERO MAY be created based on any path information dynamically computed by the local node.

The resulting Path message is used to create the recovery LSP. While

the recovery LSP exists, the Protection object in the original Path message MUST also be updated with the In-Place bit set (1). From this point on, Standard Path message processing is used in processing the resulting and original Path messages.

The merge node of a dynamically controlled recovery LSP SHOULD reset (0) the In-Place bit in the Protection object of the outgoing Path message associated with the terminated recovery LSP.

Unlike with explicit control, if the creation of a dynamically identified recovery LSP fails for any reason, the recovery LSP is removed and no error message or indication is sent upstream. With this exception, all the other procedures for explicitly controlled recovery LSPs apply to dynamically controlled recovery LSPs. These other procedures are defined above in defined in Sections 4.2.1 through 4.2.4.

7. Additional Fast Reroute Considerations

This section is under construction.

8. Updated RSVP Message Formats

This section presents the RSVP message related formats as modified by this document. Where they differ, formats for unidirectional LSPs are presented separately from bidirectional LSPs.

The format of a Path message is as follows:

```
<Path Message> ::=       <Common Header> [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION> <RSVP_HOP>
                         <TIME_VALUES>
                         [ <EXPLICIT_ROUTE> ]
                         <LABEL_REQUEST>
                         [ <PROTECTION> ]
                         [ <LABEL_SET> ... ]
                         [ <SESSION_ATTRIBUTE> ]
                         [ <NOTIFY_REQUEST> ... ]
                         [ <ADMIN_STATUS> ]
                         [ <ASSOCIATION> ... ]
                         [ <SECONDARY_EXPLICIT_ROUTE> ... ]
                         [ <POLICY_DATA> ... ]
                         <sender descriptor>
```

The format of the sender description for unidirectional LSPs is:

```
<sender descriptor> ::=  <SENDER_TEMPLATE> <SENDER_TSPEC>
                         [ <ADSPEC> ]
                         [ <RECORD_ROUTE> ]
                         [ <SUGGESTED_LABEL> ]
                         [ <RECOVERY_LABEL> ]
                         [ <SECONDARY_RECORD_ROUTE> ... ]
```

The format of the sender description for bidirectional LSPs is:

```
<sender descriptor> ::=  <SENDER_TEMPLATE> <SENDER_TSPEC>
                         [ <ADSPEC> ]
                         [ <RECORD_ROUTE> ]
                         [ <SUGGESTED_LABEL> ]
                         [ <RECOVERY_LABEL> ]
                         <UPSTREAM_LABEL>
                         [ <SECONDARY_RECORD_ROUTE> ... ]
```

The format of a PathErr message is as follows:

```
<PathErr Message> ::=    <Common Header> [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION> <ERROR_SPEC>
                         [ <ACCEPTABLE_LABEL_SET> ... ]
                         [ <SECONDARY_EXPLICIT_ROUTE> ... ]
                         [ <POLICY_DATA> ... ]
                         <sender descriptor>
```

The format of a Resv message is as follows:

```
<Resv Message> ::=       <Common Header> [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION> <RSVP_HOP>
                         <TIME_VALUES>
                         [ <RESV_CONFIRM> ]  [ <SCOPE> ]
                         [ <NOTIFY_REQUEST> ... ]
                         [ <ADMIN_STATUS> ]
                         [ <POLICY_DATA> ... ]
                         <STYLE> <flow descriptor list>

<flow descriptor list> ::= <FF flow descriptor list>
                         | <SE flow descriptor>


<FF flow descriptor list> ::= <FLOWSPEC> <FILTER_SPEC>
                         <LABEL> [ <RECORD_ROUTE> ]
                         [ <SECONDARY_RECORD_ROUTE> ... ]
                         | <FF flow descriptor list>
                         <FF flow descriptor>

<FF flow descriptor> ::= [ <FLOWSPEC> ] <FILTER_SPEC> <LABEL>
                         [ <RECORD_ROUTE> ]
                         [ <SECONDARY_RECORD_ROUTE> ... ]

<SE flow descriptor> ::= <FLOWSPEC> <SE filter spec list>

<SE filter spec list> ::= <SE filter spec>
                         | <SE filter spec list> <SE filter spec>

<SE filter spec> ::=     <FILTER_SPEC> <LABEL> [ <RECORD_ROUTE> ]
                         [ <SECONDARY_RECORD_ROUTE> ... ]
```

9. Security Considerations

This document introduces no additional security considerations. See [RFC3473] for relevant security considerations.

10. IANA Considerations

This document requests assignment of a new Association Type within the Association object. It also defines bits previously reserved in the Protection object. Both of these objects were defined in [E2E-RECOVERY].

This document also defines the Secondary Explicit Route Objects and Secondary Record Route Objects. Both of these objects of the form 11bbbbbb. The values 198 and 199 respectively are suggested. The c-type values and sub-objects associated with the Secondary Explicit Route Object should read "Same values as and sub-objects as EXPLICIT_ROUTE (C-Num 20)." The c-type values and sub-objects associated with the Secondary Record Route Object should read "Same values as and sub-objects as RECORD_ROUTE (C-Num 21)."

11. Intellectual Property Considerations

This section is taken from Section 10.4 of [RFC2026].

12. References

12.1. Normative References

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

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

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

[E2E-RECOVERY] Lang, J.P., Rekhter, Y., Papadimitriou, D., Editors, "RSVP-TE Extensions in support of End-to-End GMPLS-based Recovery", Work in Progress, draft-lang-ccamp-gmpls-recovery-e2e-signaling-03.txt, February 2004.

[FRR] Pan, et al, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", draft-ietf-mpls-rsvp-lsp-fastreroute-03.txt, December 2003.

12.2. Informative References

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

[FUNCT] J.P.Lang and B.Rajagopalan (Editors), "Generalized MPLS Recovery Functional Specification," Internet Draft, Work in Progress, draft-ietf-ccamp-gmpls-recovery-functional-01.txt, September 2003.

[TERM] E.Mannie and D.Papadimitriou (Editors), "Recovery (Protection and Restoration) Terminology for GMPLS," Internet Draft, Work in progress, draft-ietf-ccamp-gmpls-recovery-terminology-02.txt, May 2003.

13. Contributors

Lou Berger
Movaz Networks, Inc.
7926 Jones Branch Drive
Suite 615
McLean VA, 22102
Phone: +1 703 847-1801
Email: lberger@movaz.com

Igor Bryskin
Movaz Networks, Inc.
7926 Jones Branch Drive
Suite 615
McLean VA, 22102
Email: ibryskin@movaz.com

Adrian Farrel
Old Dog Consulting
Phone: +44 (0) 1978 860944
Email: adrian@olddog.co.uk

Dimitri Papadimitriou (Alcatel)
Francis Wellesplein 1
B-2018 Antwerpen, Belgium
Phone: +32 3 240-8491
Email: dimitri.papadimitriou@alcatel.be

14. Full Copyright Statement

Generated on: Sun Feb 8 20:51:23 2004