This document describes extensions to the RSVP Graceful Restart defined in [RFC3473]. The extensions enable the recovery of RSVP signaling state based on the Path message last sent by the node being restarted. Previously defined Graceful Restart mechanisms, also called nodal faults, permit recovery of signaling state from adjacent nodes when the data plane has retained the associated forwarding state across a restart. These mechanisms do not fully support recovery of ingress nodes or recovery of all RSVP objects. The presented extensions use the RSVP Hello Extensions defined in [RFC3209], and extensions for state recovery on nodal faults defined in [RFC3473]. With the presented extensions the restarting node can recover all previously transmitted Path state including the ERO and the downstream (outgoing) interface identifiers. The extensions can also be used to recover signaling state after the restart of an ingress node.
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].

1. Introduction

RSVP Graceful Restart is defined in [RFC3473] and uses mechanisms defined in [RFC3209]. [RFC3209] describes a mechanism, using RSVP Hello messages, to detect the state of an adjacent RSVP signaling process. [RFC3473] extends this mechanism to advertise the capability of retaining data plane state across the restart of a node or a "nodal fault". [RFC3473] also defines the Recovery Label object for use in the Path message of the RSVP neighbor upstream of a restarting node, to indicate that the Path message is for existing data plane state.

This document presents extensions to address two aspects of graceful restart not previously supported. The presented extensions enable the recovery of an ERO previously transmitted by a restarting node, from its downstream neighbor. The extensions also enable graceful restart of an ingress node that does not preserve full control plane state across restarts.

Per [RFC3473], a restarting node can distinguish Path messages associated with LSPs being recovered by the presence of the Recovery Label object. To determine the downstream (outgoing) interface, the restarting node must consult the data plane. This may not be possible for all types of nodes. Furthermore, data plane information is not sufficient to reconstruct EROs in many cases. The restarting node may have previously performed some form of path computation on the received ERO, such as ERO expansion to a loose next hop or a partial path computation up to the Egress node if an ERO was not received. If the restarting node is an ingress node, it may have performed a full path computation as part of the LSP setup process. The restarting node has to recover the ERO it had sent in its Path message prior to the restart, so that it can continue to include the same ERO in its Path messages after the restart. If the restarting node is an ingress node, the only source of RSVP state is the downstream RSVP neighbor.

The defined extensions provide a restarting upstream node with information previously transmitted by the node in Path messages. This is accomplished by the downstream RSVP neighbor, after reestablishing RSVP communication with the restarted node, sending a new message for every Path message it has previously received from...
the restarting node.

The new message is called the RecoveryPath message. The message conveys the contents of the last received Path message back to the restarting node. The restarting node can use the RecoveryPath message along with the state in the received Path message to associate control and data plane state and to validate the forwarding state with the state presented by the neighboring RSVP nodes.

If the restarting node is a transit node for an LSP being recovered, it will receive a Path message with a Recovery Label object from its upstream RSVP neighbor. Additionally, the RecoveryPath message allows such transit nodes to reconstruct any state that was previously dynamically constructed by the node, e.g., ERO sub-objects. If the restarting node is an ingress node, all significant signaling state can be recovered based on the RecoveryPath message.

Restarting egress nodes, and Resv message processing are not impacted by the presented extensions, see [RFC3473] for details.

2. Extensions to Nodal Fault Handling

This section presents the protocol modifications to Section 9 of [RFC3473].

2.1. RecoveryPath Message Format

The format of a RecoveryPath message is the same as the format of a Path message as defined in [RFC3473]:

\[
\text{<RecoveryPath Message>} ::= \text{<Path Message>}
\]

The destination address used in the IP header of a RecoveryPath message MUST be the same as the destination address used in the IP header of the corresponding Resv message last sent by the sending node. Except as specified below all objects in a RecoveryPath message are identical to the objects in the corresponding Path message last received by the sending node.
2.2. Related Procedures

This document does not modify existing procedures for sending and receiving RSVP Hello messages as defined in [RFC3209] and the Restart_Caps object in the RSVP Hello messages as defined in [RFC3473]. The procedures for control channel faults are defined in [RFC3473] and are not changed by this document.

The presented extensions require the use of RSVP Hellos as defined in [RFC3209] and the use of the Restart_Caps object extension as defined in [RFC3473]. The presented extensions addresses only "Nodal Faults" as defined in [RFC3473]. Control channel faults are fully addressed in [RFC3473].

Note: There are no changes to the procedures defined in Section 9.5.3 in [RFC3473] (Procedures for the Neighbor of a Restarting node). There are no changes to the procedures defined in Section 9.5.2 in [RFC3473] if the restarting node is an egress node.

The following sections assume previously defined procedures are followed, except where explicitly modified.

2.3. Procedures For The Downstream Neighbor

After a downstream RSVP neighbor has detected that its upstream node has restarted and is capable of recovery, as defined in [RFC3473], the downstream RSVP neighbor MUST send a RecoveryPath message for each LSP associated with the restarting node for which it has sent a Resv message.

The RecoveryPath message is constructed by copying all objects from the last received associated Path message, with the following exceptions:

The Message ID object is not copied. Any Message ID objects used in RecoveryPath messages are generated based on procedures defined in [RFC2961].

The Integrity object is not copied. Any Integrity objects used in RecoveryPath messages are generated based on procedures defined in [RFC2747].

The RSVP Hop object is copied from the most recent associated Resv message sent to the restarted node, for the LSP being recovered.

In the sender descriptor, the Recovery Label object MUST be included, with the label value copied from the label value in the
Label object in the most recent associated Resv message sent to the restarted node, for the LSP being recovered.

All other objects from the most recent received Path message MUST be included in the RecoveryPath message.

After sending a RecoveryPath message and during the Recovery Period, the node SHOULD periodically re-send the RecoveryPath message until it receives a corresponding response. A corresponding response is a Message ID acknowledgment or a Path message matching the RecoveryPath message. Note, per [RFC3473], Resv messages are suppressed during this recovery period until a corresponding Path message is received.

2.4. Procedures for the Restarting Node

These procedures apply during the "state recovery process" and "Recovery Period" as defined in Section 9.5.2 in [RFC3473]. Any RecoveryPath message received after the Recovery Period has expired MUST be discarded. A node MAY send a PathTear message downstream matching the discarded message.

The remaining procedures are broken down into three sub-sections. The term "resynchronized state" originally defined in [RFC3473] is used and modified in these sections. This term refers to LSP state that is fully recovered.

Signaling state may be recovered from sources other than the mechanisms defined in this document. The ingress node SHOULD consider signaling state as resynchronized for all such LSPs and follow corresponding procedures defined below. Further, recovery procedures defined below may be overridden by local policy.

Again, there are no changes to the procedures defined in Section 9.5.2 in [RFC3473] if the restarting node is an egress node.

2.4.1. Path and RecoveryPath Message Processing Related Procedures

When a node receives a RecoveryPath message during the Recovery Period, the node first checks if it has resynchronized RSVP state associated with the message. If there is resynchronized state, and a Message ID object is present in the RecoveryPath message, the node MUST follow Message ID acknowledgement procedures as defined in [RFC2961], and, consider the message as processed. If there is resynchronized state and there is no Message ID object, the node MAY send a triggered Path message, and, consider the message as processed.
If non-resynchronized state is found or the node is the ingress, the node saves the information contained in the RecoveryPath message and continues with processing as defined in the next section.

If no associated RSVP state is found and the node is not the ingress node, the node saves the information contained in the RecoveryPath message for later use.

Note the following modifies Section 9.5.2 of [RFC3473]:

When a node receives a Path message during the Recovery Period, the node first checks if it has an RSVP state associated with the message. If resynchronized RSVP state is found, then the node handles this message according to previously defined procedures.

If non-resynchronized state is found, the node saves the information contained in Recovery_Label object and continues with processing as defined in the next section.

Per [RFC3473], if the RSVP state is not found, and the message does not carry a Recovery_Label object, the node treats this as a setup for a new LSP, and handles it according to previously defined procedures.

If the RSVP state is not found, and the message carries a Recovery_Label object, the node saves the information contained in the Recovery_Label object for later use.

2.4.2. Re-Synchronization Procedures

After receipt of the RecoveryPath message and, for non-ingress LSPs, the corresponding Path message with a Recovery Label object, the restarting node SHOULD locate and associate corresponding forwarding state using the received information. The restarting node associates the corresponding active forwarding plane state from the following signaled information:

The upstream data interface is recovered from the received RSVP HOP object in the Path message.

The label on the upstream data interface is recovered from the Recovery Label object in the received Path message. If the LSP is bidirectional, the label for the reverse direction is recovered from the Upstream Label object in the received Path message.

The downstream data interface is recovered from the RSVP HOP object in the received RecoveryPath message.
The label on the downstream data interface is recovered from the Recovery Label object in the received RecoveryPath message. If the LSP is bidirectional, the label for the reverse direction is recovered from the Upstream Label object in the RecoveryPath message.

If complete forwarding state is located, the restarted node MUST treat the LSP as resynchronized and MUST send a triggered Path message downstream. The Explicit Route object in the Path message SHOULD match the Explicit Route object received RecoveryPath message. In addition, a node SHOULD recover state from the other objects received in the RecoveryPath message. The optimal result is for the resulting Path message to not cause any redundant or unnecessary re-processing of state along the remaining downstream nodes. Ideally, except for Message Id processing and recovery processing, the transmitted Path message will be treated as a refresh by the downstream RSVP neighbor (and hence should not trigger any generation of Path messages with changed state further downstream).

If no forwarding state is located, the node treats the path message as a setup for a new LSP. The outgoing interface and label(s) indicated in the RecoveryPath message SHOULD be reused, when possible. All other information contained in the RecoveryPath message MAY also be used.

2.4.3. Procedures on Expiration of Recovery Period

There are several cleanup steps to follow at the end of the Recovery Period. At the end of the Recovery Period, any state that was installed as a result of a received RecoveryPath message and is not resynchronized SHOULD be discarded.

Any received Path messages that were received containing a Recovery_Label have not been resynchronized, SHOULD be treated as being received during the Recovery Period and processed as per [RFC3473].

Per [RFC3473], any other state that is not resynchronized during the Recovery Period SHOULD be removed at the end of the Period.
3. Compatibility notes

This document introduces a new RSVP signaling message to be generated by the downstream RSVP neighbor of a restarting node.

If the restarting node does not support the RecoveryPath message and associated procedures, it will discard all received RecoveryPath messages, and revert to recovery processing as defined in [RFC3473].

If the downstream RSVP neighbor does not support the RecoveryPath message and associated procedures, the restarting node processes received Path messages as defined above, which essentially reverts to the processing defined in [RFC3473].

4. Security Considerations

This document introduces a new RSVP message that is restricted to one RSVP hop. This document introduces no new security considerations beyond those already addressed for existing RSVP hop-by-hop messages.

5. IANA Considerations

A new RSVP message type is defined in this document. The RSVP message type is TBA by IANA.

6. References

6.1. Normative References


7. Authors’ Addresses

Arun Satyanarayana  
Movaz Networks, Inc.  
7926 Jones Branch Drive  
Suite 615  
McLean VA, 22102  
Email: aruns@movaz.com

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

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

8. Full Copyright Statement

Copyright (c) The Internet Society (2004). All Rights Reserved. 
This document and translations of it may be copied and furnished to 
others, and derivative works that comment on or otherwise explain it 
or assist in its implementation may be prepared, copied, published 
and distributed, in whole or in part, without restriction of any 
kind, provided that the above copyright notice and this paragraph are 
included on all such copies and derivative works. However, this
document itself may not be modified in any way, such as by removing
the copyright notice or references to the Internet Society or other
Internet organizations, except as needed for the purpose of
developing Internet standards in which case the procedures for
copyrights defined in the Internet Standards process must be
followed, or as required to translate it into languages other than
English.

The limited permissions granted above are perpetual and will not be
revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an
"AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING
TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING
BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION
HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.