Network Working Group
Internet Draft
Expiration Date: February 2002

Ping Pan (Juniper Networks)
Yakov Rekhter (Juniper Networks)
Kireeti Kompella (Juniper Networks)
Fong Liaw (Zaffire Inc.)
Dimitrios Pendarakis (Tellium, Inc.)
George Swallow (Cisco Systems)
John Drake (Calient Networks)

Graceful Restart Mechanism for RSVP-TE

draft-pan-rsvp-te-restart-01.txt

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026, except that the right to produce derivative works is not granted.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ``work in progress.''

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

2. Abstract

This document describes a mechanism that helps to minimize the negative effects on MPLS traffic caused by LSR's control plane restart, and specifically by the restart of its RSVP-TE component, on LSRs that are capable of preserving the MPLS forwarding component across the restart.

This document also describes a mechanism that helps to minimize the negative effects on MPLS traffic caused by the disruption of the communication channel that is used to exchange RSVP messages between a pair of LSR, when the communication channel is separate from the channels carrying the actual LSPs, and the channels carrying the actual LSPs are not disrupted.

3. Summary for Sub-IP Area

3.1. Summary

This document describes a mechanism that helps to minimize the negative effects on MPLS traffic caused by LSR's control plane restart, and specifically by the restart of its RSVP-TE component, on LSRs that are capable of preserving the MPLS forwarding component across the restart.

This document also describes a mechanism that helps to minimize the negative effects on MPLS traffic caused by the disruption of the communication channel that is used to exchange RSVP messages between a pair of LSR, when the communication channel is separate from the channels carrying the actual LSPs, and the channels carrying the actual LSPs are not disrupted.

3.2. Related documents

See the Reference Section

3.3. Where does it fit in the Picture of the Sub-IP Work

This work fits squarely in either MPLS or CCAMP box.

3.4. Why is it Targeted at this WG

The RSVP TE is a product of the MPLS WG. This document specifies procedures to minimize the negative effects caused by either restart of RSVP TE module of the control plane, or by a temporary failure of communication channel used to exchange RSVP messages between a pair of LSRs. Since the procedures described in this document are directly related to RSVP TE, it would be logical to target this document at the MPLS WG.

At the same time, since the procedures described in this document use one of the RSVP objects defined in Generalized MPLS Signaling, and since Generalized MPLS Signaling belongs to CCAMP, one could also say that this document could be targeted at the CCAMP WG.

3.5. Justification

The WG should consider this document, as it allows to minimize the negative effects caused by either restart of RSVP TE module of the control plane, or by a temporary failure of communication channel used to exchange RSVP messages between a pair of LSRs.

4. Motivation

In the case where an LSR could preserve its MPLS forwarding state across restart of its control plane, and specifically its RSVP-TE component, it may be desirable not to perturb the LSPs going through that LSR (and specifically, the LSPs established by RSVP-TE). In this document, we describe a mechanism, termed "RSVP-TE Graceful Restart", that allows to accomplish this goal.

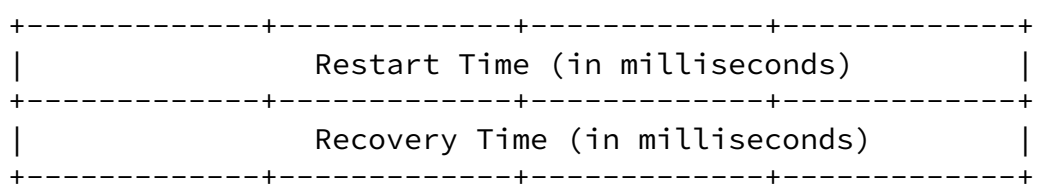
5. RSVP-TE Extension

The RSVP-TE Graceful Restart requires one new object, RESTART_CAP. The RSVP-TE Graceful Restart also uses one of the objects, SUGGESTED_LABEL, defined in GMPLS (an alternative to using the SUGGESTED_LABEL object would be to define a new object).

5.1. RESTART_CAP Object

To indicate to a neighbor the Graceful Restart capability (as well as several parameters associated with this capability), an LSR uses a new object, RESTART_CAP. This object is carried in RSVP Hello messages. The object has the following format:

Class = RESTART_CAP Class, C_Type = 1



Restart Time

This is a time (in milliseconds) that the sender of the RESTART_CAP object would like the receiver of that object to wait after the receiver detects the failure of RSVP communication with the sender. While waiting, the receiver should retain the RSVP and MPLS forwarding state for the (already established) LSPs that traverse a link between the sender and the receiver.

The Restart Time should long enough to allow the restart of the control plane, and specifically its RSVP-TE component. Likewise, the Restart Time should be long enough to allow the restart of the communication channel that is used, among other things, for RSVP communication.

Recovery Time

For a restarting LSR, this is the time (in milliseconds) that the restarting LSR is willing to retain its MPLS forwarding state that it preserved across the restart. The time is from

the moment the LSR sends the RSVP Hello message carrying this information. Setting this time to 0 indicates that the forwarding state wasn't preserved across the restart (or even if it was preserved, is no longer available).

For an (non restarting) LSR that re-established an RSVP adjacency with a neighbor, this is the time (in milliseconds) that the LSR is willing to retain its RSVP and MPLS state for the (already established) LSPs that traverse a link between the LSR and the neighbor.

The Recovery Time should be long enough to allow the neighboring LSR's to re-sync all the LSP's in a graceful manner, without creating congestion in the RSVP-TE control plane.

Value 0xffffffff in the Recovery Time is used to indicate that the MPLS forwarding state has been preserved across the restart, and will be retained until removed by means outside of

the mechanisms described in this document.

To support RSVP-TE Graceful Restart, a RSVP Hello message can be as follows:

<Hello Message> ::= <Common Header> [<INTEGRITY>] <HELLO>

[<RESTART_CAP>]

6. Operations

For the sake of brevity in the context of this document by "the control plane" we mean "the RSVP-TE component of the control plane".

An LSR that is capable of retaining its MPLS forwarding state across restart of its control plane should advertise this capability to its neighbors by carrying the RESTART_CAP object in the Hello messages it sends to the neighbors.

Note that an LSR should advertise this capability to a neighbor only

when the Dst_instance that the LSR advertises to the neighbor is 0.

6.1. Procedures for the restarting LSR

After an LSR restarts its control plane, the LSR should check whether it was able to preserve its MPLS forwarding state from prior to the restart. If no, then the LSR must set the Recovery Time to 0 in the Hellos the LSR sends to its neighbors.

If the forwarding state has been preserved, then the LSR starts its internal timer, called MPLS Forwarding State Holding timer (the value of that timer should be configurable), and marks all the MPLS forwarding state entries as "stale". At the expiration of the timer all the entries still marked as stale should be purged. The value of the Recovery Time advertised in RSVP Hello messages should be set to the (current) value of the timer at the point when the Hello message carrying the Recovery Time is sent.

We say that an LSR is in the process of restarting when the MPLS Forwarding State Holding timer is not expired. Once the timer expires, we say that the LSR completed its restart.

The following procedures apply when an LSR is in the process of restarting.

When the LSR receives a Path message from an (upstream) neighbor, the LSR first checks if it has an RSVP state associated with the message. If the state is found, then the LSR handles this message according to the procedures defined in [RSVP-TE] (this is irrespective of whether the message carries the SUGGESTED_LABEL object or not). In addition, if the LSR is not the tail-end of the LSP associated with the Path message, and the downstream neighbor is in the process of restarting, the LSR places the outgoing label (the label that was received in the LABEL object from that neighbor prior to the neighbor's restart) in the SUGGESTED_LABEL object of the Path message that the LSR sends to the neighbor.

If the RSVP state is not found, and the message does not carry the SUGGESTED_LABEL object, the LSR treats this as a setup for a new LSP, and handles it according to the procedures defined in [RSVP-TE].

If the RSVP state is not found, and the message carries the SUGGESTED_LABEL object, the LSR searches its MPLS forwarding table (the one that was preserved across the restart) for an entry whose incoming label is equal to the label carried in the SUGGESTED_LABEL object (in the case of link bundling, this may also involve first identifying the appropriate incoming component link).

If the MPLS forwarding table entry is not found, the LSR treats this as a setup for a new LSP, and handles it according to the procedures defined in [RSVP-TE].

If the MPLS forwarding table entry is found, the appropriate RSVP state is created, the entry is bound to the LSP associated with the message, and the entry is no longer marked as stale. In addition, if the LSR is not the tail-end of the LSP, and the next hop LSR is in the process of restarting, the outgoing label from the entry is sent in the SUGGESTED_LABEL object of the Path message further downstream (in the case of link bundling the found entry also identifies the appropriate outgoing component link).

For bidirectional LSP [GMPLS], in addition to the procedures described above, the LSR extracts the label from the UPSTREAM_LABEL object carried in the received Path message, and searches its MPLS forwarding table for an entry whose outgoing label is equal to the label carried in the object (in the case of link bundling, this may also involved first identifying the appropriate incoming component link).

If the MPLS forwarding table entry is not found, the LSR treats this as a setup for a new LSP, and handles it according to the procedures defined in [RSVP-TE].

If the MPLS forwarding table entry is found, the entry is bound to the LSP associated with the Path message, and the entry is no longer marked as stale. In addition, if the LSR is not the tail-end of the LSP, the incoming label from that entry is sent in the UPSTREAM_LABEL object of the Path message further downstream (in the case of link bundling the found entry also identifies the appropriate outgoing component link).

The Resv messages are processed as specified in [RSVP-TE], except that if the LSR, while in the process of restarting, receives a Resv message for which the LSR has no matching Path State Block, the LSR should not generate an RERR message specifying "no path information for this Resv", but just should drop the Resv message.

6.2. Procedures for restart of RSVP communication with a neighbor LSR

An RSVP communication between an LSR and its neighbor could go down for two reasons: (1) the channel that carries the RSVP messages between the LSR and its neighbor went down, and (2) the neighbor's control plane went down.

When an LSR detects that its communication with a neighbor's control plane went down, and the LSR knows that the neighbor is capable of preserving its MPLS forwarding state across restart (as was indicated by the presence of the RESTART_CAP object in the Hello messages received from the neighbor), the LSR should wait certain amount of time before taking any further actions.

The amount of time the LSR is willing to wait is set to the lesser of the Restart Time, as was advertised by the neighbor, and a local timer. The local timer is started when the LSR detects that its communication with the neighbor's control plane went down. The value of the local timer should be configurable. While waiting, the LSR should try to re-establish RSVP communication with the neighbor.

While attempting to re-establish the RSVP communication with the neighbor, the LSR MUST use 0 as the Dst_instance, and the same Src_instance as the one it used before the communication went down. The Recovery Time carried in the RESTART_CAP object of the Hellos that the LSR sends to the neighbor should be set to the amount of time the LSR is willing to wait before taking any further actions.

While waiting, the LSR should preserve the RSVP and MPLS forwarding state for the (already) established LSPs that traverse the link(s) between the LSR and the neighbor. In a sense with respect to such LSPs the LSR should behave as it continues to receive periodic RSVP refresh messages from the neighbor. At the same time, the LSR may

clear LSPs that are in the process of being established when their

refresh timers expire.

6.2.1. Neighbor's control plane restart

The following specifies the procedures that apply when the LSR re-establishes communication with the neighbor's control plane within the Restart Time, the LSR determines (using the procedures defined in Section 5 of [RSVP-TE]) that the neighbor's control plane re-started, and the neighbor was able to preserve its forwarding state across the restart (as was indicated by a non-zero Recovery Time carried in the RESTART_CAP object of the RSVP Hello messages received from the neighbor).

For each LSP that traverses the LSR, and for which the neighbor is the next hop, the LSR places the outgoing label (the label that was received in the LABEL object from that neighbor prior to the neighbor's restart) in the SUGGESTED_LABEL object of the Path message that the LSR sends to the neighbor.

When the LSR receives a Path message from the neighbor, and the LSR itself is in the process of restarting, the LSR handles the Path message as described in the previous section.

When the LSR receives a Path message from the neighbor, and the LSR completed its restart, the LSR handles this message according to the procedures defined in [RSVP-TE] (this is irrespective of whether the message carries the SUGGESTED_LABEL object or not).

The Resv messages are processed as specified in [RSVP-TE], except that the LSR should send no Resv messages to the restarting neighbor until it first receives the Path messages from the neighbor.

If there are many LSP's going through the restarting LSR, the neighbor LSR should avoid sending Path messages in a short time interval, as to avoid unnecessary stressing the restarting LSR's CPU. Instead, it should spread the messages across the Recovery Time interval.

6.2.2. Restart of RSVP control channel

If the RSVP communication is re-established, the received Src_instance is unchanged, and the Recovery Time received from the neighbor is non-zero, then the LSR should treat the situation as simply an RSVP communication channel restart (and not as a restart of the neighbor's control plane).

Once the communication gets re-established, the LSR SHOULD send an RSVP Summary Refresh to the neighbor. A Summary Refresh messages containing the message_IDs for all acknowledged messages should be sent. IF the number of message_IDs causes the message to exceed the MTU, multiple messages are sent. These messages should carry their own message_ID with the ack requested flag set. This simply ensures that the Summary Refresh messages are reliably sent. From this point normal Summary Refresh procedures are followed. For any message that have not be acked, or did not carry a message_ID, normal procedures are followed. Note that if a large number of messages are due for immediate refresh, some pacing should be applied.

7. RSVP Refresh Overhead Reduction Extensions

The mechanisms described in this document may benefit when combined with the RSVP Refresh Overhead Reduction Extensions, as specified in [RFC2961].

8. Fast Reroute and Graceful Restart

Fast reroute [FR-Juniper, FR-Cisco] and RSVP-TE graceful restart are two complement techniques that are designed to protect traffic during failures. Here are the conditions for their usage:

(1) If the interface to a neighbor is up, and the LSR does not detect any communication problem with the neighbor's control plane, do nothing.

(2) If the interface to a neighbor is up, and the LSR detects that its communication with a neighbor's control plane went down, the LSR should activate RSVP-TE graceful restart.

(3) If the interface to a neighbor is up, but the LSR cannot receive Hello messages from the neighbor, the LSR should activate RSVP-TE graceful restart.

(4) If the interface to a neighbor goes down, the LSR should activate fast reroute.

9. Security Consideration

This document does not introduce new security issues. The security considerations pertaining to the original RSVP protocol remain relevant.

10. Intellectual Property Considerations

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

11. Acknowledgments

We acknowledge the helpful comments from Arthi Ayyangar, Bruce Cole, Manoj Leelanivas, Der-Hwa Gan, Nischal Sheth, Ewart Tempest, and Jonathan Lang.

12. References

[RFC2961] L. Berger, et al, "RFC 2961: RSVP Refresh Overhead Reduction Extensions", RFC2961.

[RSVP] R. Braden, Ed., et al, "Resource ReSerVation protocol (RSVP) -- version 1 functional specification," RFC2205.

[RSVP-TE] D. Awduche, et al, "RSVP-TE: Extensions to RSVP for LSP tunnels," Internet Draft.

[GMPLS] P. Ashwood-Smith, et al, "Generalized MPLS Signaling - RSVP-TE Extensions", Internet Draft.

[FR-Juniper] D. Gan, et al, "A Method for MPLS LSP Fast-Reroute Using

RSVP Detours", Internet Draft.

[FR-Cisco] R. Goguen, et al, "RSVP Label Allocation for Backup Tunnels", Internet Draft.

13. Author Information

Ping Pan
Juniper Networks
1194 N.Mathilda Ave
Sunnyvale, CA 94089
e-mail: pingpan@juniper.net

Yakov Rekhter
Juniper Networks
1194 N.Mathilda Ave
Sunnyvale, CA 94089
e-mail: yakov@juniper.net

Kireeti Kompella
Juniper Networks
1194 N.Mathilda Ave
Sunnyvale, CA 94089
e-mail: kireeti@juniper.net

Fong Liaw
Zaffire Inc.
2630 Orchard Parkway,
San Jose, CA 95134
e-mail: fliaw@zaffire.com

Dimitrios Pendarakis
Tellium, Inc.
2 Crescent Place
P.O. Box 901
Oceanport, NJ 07757
tel. 732 923-4254
e-mail: dpendarakis@tellium.com

George Swallow
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Voice: +1 978 244 8143
e-mail: swallow@cisco.com

John Drake
Calient Networks
5853 Rue Ferrari
San Jose, CA 95138
e-mail: jdrake@calient.net