Internet Engineering Task Force (IETF)
Request for Comments: 6383
Category: Informational
ISSN: 2070-1721
September 2011

K. Shiomoto
NTT
A. Farrel
Old Dog Consulting

Advice on When It Is Safe to Start Sending Data on Label Switched Paths Established Using RSVP-TE

Abstract

The Resource Reservation Protocol (RSVP) has been extended to support Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks. The protocol enables signaling exchanges to establish Label Switched Paths (LSPs) that traverse nodes and link to provide end-to-end data paths. Each node is programmed with "cross-connect" information as the signaling messages are processed. The cross-connection information instructs the node how to forward data that it receives.

End points of an LSP need to know when it is safe to start sending data so that it is not misdelivered, and so that safety issues specific to optical data-plane technology are satisfied. Likewise, all label switching routers along the path of the LSP need to know when to program their data planes relative to sending and receiving control-plane messages.

This document clarifies and summarizes the RSVP-TE protocol exchanges with relation to the programming of cross-connects along an LSP for both unidirectional and bidirectional LSPs. This document does not define any new procedures or protocol extensions, and defers completely to the documents that provide normative references. The clarifications set out in this document may also be used to help interpret LSP establishment performance figures for MPLS-TE and GMPLS devices.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

1. Introduction

The Resource Reservation Protocol (RSVP) [RFC2205] has been extended to support Traffic Engineering (TE) in Multiprotocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks [RFC3209] [RFC3473]. The protocol enables signaling exchanges to establish Label Switched Paths (LSPs) that traverse nodes and links to provide end-to-end data paths. Each node is programmed with "cross-connect" information as the signaling messages are processed. The cross-connection information instructs the node how to forward data that it receives. In some technologies this requires configuration of physical devices, while in others it may involve the exchange of commands between different components of the node. The nature of a cross-connect is described further in Section 1.1.1.

End points of an LSP need to know when it is safe to start sending data. In this context "safe" has two meanings. The first issue is that the sender needs to know that the data path has been fully established, setting up the cross-connects and removing any old, incorrect forwarding instructions, so that data will be delivered to the intended destination. The other meaning of "safe" is that in optical technologies, lasers must not be turned on until the correct cross-connects have been put in place to ensure that service personnel are not put at risk.

Similarly, all Label Switching Routers (LSRs) along the path of the LSP need to know when to program their data planes relative to sending and receiving control-plane messages.

This document clarifies and summarizes the RSVP-TE protocol exchanges with relation to the programming of cross-connects along an LSP for both unidirectional and bidirectional LSPs. Bidirectional LSPs, it should be noted, are supported only in GMPLS. This document does not define any new procedures or protocol extensions, and defers completely to the documents that provide normative references.

The clarifications set out in this document may also be used to help interpret LSP establishment performance figures for MPLS-TE and GMPLS devices. For example, the dynamic provisioning performance metrics set out in [RFC5814] need to be understood in the context of LSP setup times and not in terms of control message exchange times that are actually only a component of the whole LSP establishment process.

Implementations could significantly benefit from this document definitively identifying any LSR to forward the Path or Resv message [RFC3473] before programming its cross-connect, thereby exploiting pipelining (i.e., doing one action in the background while another is progressing) to try to minimize the total time to set up the LSP. However, while this document gives advice and identifies the issues to be considered, it is not possible to make definitive statements about how much pipelining is safe, since a node cannot "know" much without first probing the network (for example, with protocol extensions) which would defeat the point of pipelining. Due to the number of variables introduced by path length, and other node behavior, ingress might be limited to a very pessimistic view for safety. Furthermore, it seems unlikely that an implementation would necessarily give a full and frank description of how long it takes to program and stabilize its cross-connects. Nevertheless, this document identifies the issues and opportunities for pipelining in GMPLS systems.

1.1. Terminology

It is assumed that the reader is familiar with the basic message flows of RSVP-TE as used in MPLS-TE and GMPLS. Refer to [RFC2205], [RFC3209], [RFC3471], and [RFC3473] for more details.

1.1.1. What is a Cross-Connect?

In the context of this document, the concept of a "cross-connection" should be taken to imply the data forwarding instructions installed (that is, "programmed") at a network node (or "switch").

In packet MPLS networks, this is often referred to as the Incoming Label Map (ILM) and Next Hop Label Forwarding Entry (NHLFE) [RFC3031] which are sometimes considered together as entries in the Label Forwarding Information Base (LFIB) [RFC4221]. Where there is admission control and resource reservation associated with the data forwarding path (such as the allocation of data buffers) [RFC3209], this can be treated as part of the cross-connect programming process since the LSP will not be available to forward data in the manner agreed to during the signaling protocol exchange until the resources are correctly allocated and reserved.

In non-packet networks (such as time-division multiplexing, or optical switching networks), the cross-connect concept may be an electronic cross-connect array or a transparent optical device (such as a microelectromechanical system (MEMS)). In all cases, however, the concept applies to the instructions that are programmed into the forwarding plane (that is, the data plane) so that incoming data for the LSP on one port can be correctly handled and forwarded out of another port.

2. Unidirectional MPLS-TE LSPs

[RFC3209] describes the RSVP-TE signaling and processing for MPLS-TE packet-based networks. LSPs in these networks are unidirectional by definition (there are no bidirectional capabilities in [RFC3209]).

Section 4.1.1.1 of [RFC3209] describes a node's process prior to sending a Resv message to its upstream neighbor.

> The node then sends the new LABEL object as part of the Resv message to the previous hop. The node SHOULD be prepared to forward packets carrying the assigned label prior to sending the Resv message.

This means that the cross-connect should be in place to support traffic that may arrive at the node before the node sends the Resv. This is clearly advisable because the upstream LSRs might otherwise complete their cross-connections more rapidly and encourage the ingress to start transmitting data with the risk that the node that sent the Resv "early" would be unable to forward the data it received and would be forced to drop it, or might accidentally send it along the wrong LSP because of stale cross-connect information.

The use of "SHOULD" [RFC2119] in this text indicates that an implementation could be constructed that sends a Resv before it is ready to receive and forward data. This might be done simply because the internal construction of the node means that the control-plane components cannot easily tell when the cross-connection has been installed. Alternatively, it might arise because the implementation is aware that it will be slow and does not wish to hold up the establishment of the LSP. In this latter case, the implementation is choosing to pipeline the cross-connect programming with the protocol exchange taking a gamble that there will be other upstream LSRs that may also take some time to process, and it will in any case be some time before the ingress actually starts to send data. It should be noted that, as well as the risks described in the previous paragraph, a node that behaves like this must include a mechanism to report a failure to chase the Resv message (using a PathErr) in the event that the pipelined cross-connect processing fails.

3. GMPLS LSPs

GMPLS [RFC3945] extends RSVP-TE signaling for use in networks of different technologies [RFC3471] [RFC3473]. This means that RSVP-TE signaling may be used in MPLS packet switching networks, as well as layer two networks (Ethernet, Frame Relay, ATM), time-division multiplexing networks (Time Division Multiplexer (TDM), i.e., Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH)), Wavelength Division Multiplexing (WDM) networks, and fiber switched network.

The introduction of these other technologies, specifically the optical technologies, brings about the second definition of the "safe" commencement of data transmission as described in Section 1. That is, there is a physical safety issue that means that the lasers should not be enabled until the cross-connects are correctly in place.

GMPLS supports unidirectional and bidirectional LSPs. These are split into separate sections for discussion. The processing rules are inherited from [RFC3209] unless they are specifically modified by [RFC3471] and [RFC3473].

3.1. Unidirectional LSPs

Unidirectional LSP processing would be the same as that described in Section 2 except for the use of the Suggested_Label object defined in [RFC3473]. This object allows an upstream LSR to 'suggest' to its downstream neighbor the label that should be used for forward-direction data by including the object on a Path message. The purpose of this object is to help the downstream LSR in its choice of label, but it also makes it possible for the upstream LSR to 'pipeline' programming its cross-connect with the RSVP-TE signaling exchanges. That means that the cross-connect might be in place before the signaling has completed (i.e., before a Resv message carrying a Label object has been received at the upstream LSR).

We need to know when it is safe to start sending data. There are three sources of information.

- Section 3.4 of [RFC3471] states:

  In particular, an ingress node should not transmit data traffic on a suggested label until the downstream node passes a label upstream.

The implication here is that an ingress node may (safely) start to transmit data when it receives a label in a Resv message.

- Section 2.5 of [RFC3473] states:

  Furthermore, an ingress node SHOULD NOT transmit data traffic using a suggested label until the downstream node passes a corresponding label upstream.

This is a confirmation of the first source.

- Section 4.1.1.1 of [RFC3209] states:

  The node then sends the new LABEL object as part of the Resv message to the previous hop. The node SHOULD be prepared to forward packets carrying the assigned label prior to sending the Resv message.

In this text, the word "prior" is very important. It means that the cross-connect must be in place for forward traffic before the Resv is sent. In other words, each of the transit nodes and the egress node must finish making their cross-connects before they send the Resv message to their upstream neighbors.

Thus, as in Section 2, we can deduce that the ingress must not start to transmit traffic until it has both received a Resv and has programmed its own cross-connect.

3.2. Bidirectional LSPs

A bidirectional LSP is established with one signaling exchange of a Path message from ingress to egress, and a Resv from egress to ingress. The LSP itself is comprised of two sets of forwarding state, one providing a path from the ingress to the egress (the forwards data path), and one from the egress to the ingress (the reverse data path).

3.2.1. Forwards Direction Data

The processing for the forwards direction data path is exactly as described for a unidirectional LSP in Section 3.1.

3.2.2. Reverse Direction Data

For the reverse direction data flow, an Upstream_Label object is carried in the Path message from each LSR to its downstream neighbor. The Upstream_Label object tells the downstream LSR which label to use for data being sent to the upstream LSR (that is, reverse direction data). The use of the label is confirmed by the downstream LSR when it sends a Resv message. Note that there is no explicit confirmation of the label in the Resv message, but if the label was not acceptable to the downstream LSR, it would return a PathErr message instead.

The upstream LSR must decide when to send the Path message relative to when it programs its cross-connect. That is:

- Should it program the cross-connect before it sends the Path message;

- Can it overlap the programming with the exchange of messages; or

- Must it wait until it receives a Resv from its downstream neighbor?

The defining reference is Section 3.1 of [RFC3473]:

> The Upstream_Label object MUST indicate a label that is valid for forwarding at the time the Path message is sent.

In this text, "valid for forwarding" should be taken to mean that it is safe for the LSR that sends the Path message to receive data, and that the LSR will forward data correctly. The text does not mean that the label is "acceptable for use" (i.e., the label is available to be cross-connected).

This point is clarified later in Section 3.1 of [RFC3473]:

> Terminator nodes process Path messages as usual, with the exception that the upstream label can immediately be used to transport data traffic associated with the LSP upstream towards the initiator.

This is a clear statement that when a Path message has been fully processed by an egress node, it is completely safe to transmit data toward the ingress (i.e., reverse direction data).

From this we can deduce several things:

- An LSR must not wait to receive a Resv message before it programs the cross-connect for the reverse direction data. It must be ready to receive data from the moment that the egress completes processing the Path message that it receives (i.e., before it sends a Resv back upstream).

- An LSR may expect to start receiving reverse direction data as soon as it sends a Path message for a bidirectional LSP.

- An LSR may make some assumptions about the time lag between sending a Path message and the message reaching and being processed by the egress. It may take advantage of this time lag to pipeline programming the cross-connect.

3.3. ResvConf Message

The ResvConf message is used in standard RSVP [RFC2205] to let the ingress confirm to the egress that the Resv has been successfully received, and what bandwidth has been reserved. In RSVP-TE [RFC3209] and GMPLS [RFC3473], it is not expected that bandwidth will be modified along the path of the LSP, so the purpose of the ResvConf is reduced to a confirmation that the LSP has been successfully established.

The egress may request that a ResvConf be sent by including a Resv_Confirm object in the Resv message that it sends. When the ingress receives the Resv message and sees the Resv_Confirm object, it can respond with a ResvConf message.

It should be clear that this mechanism might provide a doubly secure way for the egress to ensure that the reverse direction data path is safely in place before transmitting data. That is, if the egress waits until it receives a ResvConf message, it can be sure that the whole LSP is in place.

However, this mechanism is excessive given the definitions presented in Section 3.2.2, and would delay LSP setup by one end-to-end message propagation cycle. It should be noted as well that the generation and of the ResvConf message is not guaranteed. Furthermore, many (if not most) GMPLS implementations neither request nor send ResvConf messages. Therefore, egress reliance on the receipt of a ResvConf as a way of knowing that it is safe to start transmitting reverse direction data is not recommended.

3.4. Administrative Status

GMPLS offers an additional tool for ensuring safety of the LSP. The Administrative Status information is defined in Section 8 of [RFC3471] and is carried in the Admin_Status Object defined in Section 7 of [RFC3473].

This object allows an ingress to set up an LSP in "Administratively Down" state. This state means that [RFC3471]:

> ... the local actions related to the "administratively down" state should be taken.

In this state, it is assumed that the LSP exists (i.e., the cross-connects are all in place), but no data is transmitted (i.e., in optical systems, the lasers are off).

Additionally, the Admin_Status object allows the LSP to be put into "Testing" state. This state means ([RFC3471]) that:

> ... the local actions related to the "testing" mode should be taken.

This state allows the connectivity of the LSP to be tested without actually exchanging user data. For example, in an optical system, it would be possible to run a data continuity test (using some external coordination of errors). In a packet network, a connection verification exchange (such as the in-band Virtual Circuit Connectivity Verification described in Section 5.1.1 of [RFC5085]) could be used. Once connectivity has been verified, the LSP could be put into active mode and the exchange of user data could commence.

These processes may be considered particularly important in systems where the control-plane processors are physically distinct from the data-plane cross-connects (for example, where there is a communication protocol operating between the control-plane processor and the data-plane switch) in which case the successful completion of control-plane signaling cannot necessarily be taken as evidence of correct data-plane programming.

4. Implications for Performance Metrics

The ability of LSRs to handle and propagate control-plane messages and to program cross-connects varies considerably from device to device according to switching technology, control-plane connectivity, and implementation. These factors influence how quickly an LSP can be established.

Different applications have different requirements for the speed of setup of LSPs, and this may be particularly important in recovery scenarios. It is important for service providers considering the deployment of MPLS-TE or GMPLS equipment to have a good benchmark for the performance of the equipment. Similarly, it is important for equipment vendors to be compared on a level playing field.

In order to provide a basis for comparison, [RFC5814] defines a series of performance metrics to evaluate dynamic LSP provisioning performance in MPLS-TE/GMPLS networks. Any use of such metrics must be careful to understand what is being measured, bearing in mind that it is not enough to know that the control-plane message has been processed and forwarded: the cross-connect must be put in place before the LSP can be used. Thus, care must be taken to ensure that devices are correctly conforming to the procedures clarified in Section 2 of this document, and not simply forwarding control-plane messages with the intent to program the cross-connects in the background.

5. Security Considerations

This document does not define any network behavior and does not introduce or seek to solve any security issues.

It may be noted that a clear understanding of when to start sending data may reduce the risk of data being accidentally delivered to the wrong place or individuals being hurt.

6. Acknowledgments

Thanks to Weiqiang Sun, Olufemi Komolafe, Daniel King, and Stewart Bryant for their review and comments.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

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

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

[RFC3945] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

7.2. Informative References

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC4221] Nadeau, T., Srinivasan, C., and A. Farrel, "Multiprotocol Label Switching (MPLS) Management Overview", RFC 4221, November 2005.

[RFC5085] Nadeau, T., Ed., and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, December 2007.

[RFC5814] Sun, W., Ed., and G. Zhang, Ed., "Label Switched Path (LSP) Dynamic Provisioning Performance Metrics in Generalized MPLS Networks", RFC 5814, March 2010.

Authors' Addresses

Kohei Shiomoto
NTT Service Integration Laboratories
3-9-11 Midori
Musashino, Tokyo 180-8585
Japan
Phone: +81 422 59 4402
EMail: shiomoto.kohei@lab.ntt.co.jp

Adrian Farrel
Old Dog Consulting
EMail: adrian@olddog.co.uk