MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 19, 2015

R. Torvi
R. Bonica
Juniper Networks
I. Minei
Google, Inc.
M. Conn
D. Pacella
L. Tomotaki
M. Wygant
Verizon

May 18, 2015

LSP Self-Ping
draft-bonica-mpls-self-ping-06

Abstract

When certain RSVP-TE optimizations are implemented, ingress LSRs can receive RSVP RESV messages before forwarding state has been installed on all downstream nodes. According to the RSVP-TE specification, the ingress LSR can forward traffic through an LSP as soon as it receives a RESV message. However, if the ingress LSR forwards traffic through the LSP before forwarding state has been installed on all downstream nodes, traffic can be lost.

This memo describes LSP Self-ping. When an ingress LSR receives an RESV message, it can invoke LSP Self-ping procedures to ensure that forwarding state has been installed on all downstream nodes.

LSP Self-ping is an extremely light-weight mechanism. It does not consume control plane resources on transit or egress LSRs.

Requirements Language

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 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Table of Contents

1. Introduction

Ingress Label Switching Routers (LSR) use RSVP-TE [RFC3209] to establish MPLS Label Switched Paths. The following paragraphs describe RSVP-TE procedures.

The ingress LSR calculates path between itself and an egress LSR. The calculated path can be either strictly or loosely routed. Having calculated a path, the ingress LSR constructs an RSVP PATH message.

The PATH message includes an Explicit Route Object (ERO) that represents the path between the ingress and egress LSRs.

The ingress LSR forwards the PATH message towards the egress LSR, following the path defined by the ERO. Each transit LSR that receives the PATH message executes admission control procedures. If the transit LSR admits the LSP, it sends the PATH message downstream, to the next node in the ERO.

When the egress LSR receives the PATH message, it binds a label to the LSP. The label can be implicit null, explicit null, or non-null. The egress LSR then installs forwarding state (if necessary), and constructs an RSVP RESV message. The RESV message contains a Label Object that includes the label that has been bound to the LSP.

The egress LSR sends the RESV message upstream towards the ingress LSR. The RESV message visits the same transit LSRs that the PATH message visited, in reverse order. Each transit LSR binds a label to the LSP, updates its forwarding state and updates the RESV message. As a result, the Label Object in the RESV message contains the label that has been bound to the LSP most recently. Finally, the transit LSR sends the RESV message upstream, along the reverse path of the LSP.

When the ingress LSR receives the RESV message, it installs forwarding state. Once the ingress LSR installs forwarding state it can forward traffic through the LSP.

Some implementations optimize the above-described procedure by allowing LSRs to send RESV messages before installing forwarding state. This optimization is desirable, because it allows LSRs to install forwarding state in parallel, thus accelerating the process of LSP signaling and setup. However, this optimization creates a race condition. When the ingress LSR receives a RESV message, some downstream LSRs may not have installed forwarding state yet. If the ingress LSR forwards traffic through the LSP before forwarding state has been installed on all downstream nodes, traffic can be lost.

This memo describes LSP Self-ping. When an ingress LSR receives an RESV message, it can invoke LSP Self-ping procedures to verify that forwarding state has been installed on all downstream nodes. By verifying the installation of downstream forwarding state, the ingress LSR eliminates this particular cause of traffic loss.

LSP Self-ping is an extremely light-weight mechanism. It does not consume control plane resources on transit or egress LSRs.

Although LSP Ping and LSP Self-ping are named similarly, each is a unique protocol. Each protocol listens on its own UDP port and executes is own unique procedures.

2. Applicability

LSP Self-ping is applicable in the following scenario:

- The ingress LSR receives a RESV message
- The RESV message indicates that all downstream nodes have begun the process of forwarding state installation
- The RESV message does not guarantee that all downstream nodes have completed the process of forwarding state installation
- The ingress LSR needs to confirm that all downstream nodes have completed the process for forwarding state installation
- The ingress LSR does not need to confirm the correctness of downstream forwarding state, because there is a very high likelihood that downstream forwarding state is correct
- Control plane resources on the egress LSR may be scarce
- The need to conserve control plane resources on the egress LSR outweighs the need to determine whether downstream forwarding state is correct

Unlike LSP Ping [RFC4379] and S-BFD [I-D.akiya-bfd-seamless-base], LSP Self-ping is not a general purpose MPLS OAM mechanism. It cannot reliably determine whether downstream forwarding state is correct. For example, if a downstream LSR installs a forwarding state that causes an LSP to terminate at the wrong node, LSP Self-ping will not detect an error. Furthermore, LSP Self-ping fails when either of the following conditions are true:

- The LSP under test is signaled by the Label Distribution Protocol (LDP) Independent Mode [RFC5036]
- Reverse Path Forwarding (RPF) [RFC3704] filters are enabled on links that connect the ingress LSR to the egress LSR

While LSP Ping and S-BFD are general purpose OAM mechanisms, they are not applicable in the above described scenario because:

- LSP Ping consumes control plane resources on the egress LSR

- An S-BFD implementation either consumes control plane resources on the egress LSR or requires special support for S-BFD on the forwarding plane.

By contrast, LSP Self-ping requires nothing from the egress LSR beyond the ability to forward an IP datagram.

LSP Self-ping's purpose is to determine whether forwarding state has been installed on all downstream LSRs. Its primary constraint is to minimize its impact on egress LSR performance. This functionality is required during network convergence events that impact a large number of LSPs.

Therefore, LSP Self-ping is applicable in the scenario described above, where the LSP is signaled by RSVP, RPF is not enabled, and the need to conserve control plane resources on the egress LSR outweighs the need to determine whether downstream forwarding state is correct.

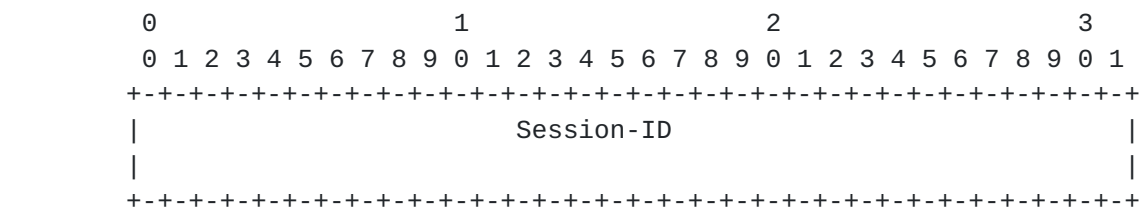
3. The LSP Self-ping Message

The LSP Self-ping Message is a User Datagram Protocol (UDP) [RFC0768] packet. If the RSVP messages used to establish the LSP under test were delivered over IPv4 [RFC0791], the UDP datagram MUST be encapsulated in an IPv4 header. If the RSVP messages used to establish the LSP were delivered over IPv6 [RFC2460], the UDP datagram MUST be encapsulated in an IPv6 header.

In either case:

- The IP Source Address MAY be configurable. By default, it MUST be the address of the egress LSR
- The IP Destination Address MUST be the address of the ingress LSR
- The IP Time to Live (TTL) / Hop Count MAY be configurable. By default, it MUST be 255
- The IP DSCP MAY be configurable. By default, it MUST be CS6 (Ox48) [RFC4594]
- The UDP Source Port MUST be selected from the dynamic range (49152-65535) [RFC6335]
- The UDP Destination Port MUST be LSP Self-ping. (Value to be assigned by IANA. See Section 7)

UDP packet contents have the following format:



LSP Self-ping Message

The Session-ID is a 64-bit field that associates an LSP Self-ping message with an LSP Self-ping session.

4. LSP Self Ping Procedures

In order to verify that an LSP is ready to carry traffic, the ingress LSR creates a short-lived LSP Self-ping session. All session state is maintained locally on the ingress LSR. Session state includes the following information:

- Session-ID: A 64-bit number that identifies the LSP Self-ping session
- Retry Counter: The maximum number of times that the ingress LSR probes the LSP before terminating the LSP Self-ping session. The initial value of this variable is determined by configuration.
- Retry Timer: The number of milliseconds that the LSR waits after probing the LSP. The initial value of this variable is determined by configuration.
- Status: A boolean variable indicating the completion status of the LSP Self-ping session. The initial value of this variable is FALSE.

Implementation MAY represent the above mentioned information in any format that is convenient to them.

The ingress LSR executes the following procedure until Status equals TRUE or Retry Counter equals zero:

- Format a LSP Self-ping message.
- Set the Session-ID in the LSP Self-ping message to the Session-ID mentioned above
- Send the LSP Self-ping message through the LSP under test

- Set a timer to expire in Retry Timer milliseconds

- Wait until either an LSP Self-ping message associated with the session returns or the timer expires. If an LSP Self-ping message associated with the session returns, set Status to TRUE. Otherwise, decrement the Retry Counter. Optionally, increase the value of Retry Timer according to an appropriate back off algorithm.

In the process described above, the ingress LSR addresses an LSP Self-ping message to itself and forwards that message through the LSP under test. If forwarding state has been installed on all downstream LSRs, the egress LSR receives the LSP Self-ping message and determines that it is addressed to the ingress LSR. So, the egress LSR forwards LSP Self-ping message back to the ingress LSR, exactly as it would forward any other IP packet.

The LSP Self-ping message can arrive at the egress LSR with or without an MPLS header, depending on whether the LSP under test executes penultimate hop-popping procedures. If the LSP Self-ping message arrives at the egress LSR with an MPLS header, the egress LSR removes that header.

If the egress LSR's most preferred route to the ingress LSR is through an LSP, the egress LSR forwards the LSP Self-ping message through that LSP. However, if the egress LSR's most preferred route to the ingress LSR is not through an LSP, the egress LSR forwards the LSP Self-ping message without MPLS encapsulation.

When an LSP Self-ping session terminates, it returns its completion status to the invoking protocol. For example, if RSVP-TE invokes LSP Self-ping as part of the LSP set-up procedure, LSP Self-ping returns its completion status to RSVP-TE.

5. Bidirectional LSP Procedures

A bidirectional LSP has an active side and a passive side. The active side calculates the ERO and signals the LSP in the forward direction. The passive side reverses the ERO and signals the LSP in the reverse direction.

When LSP Self-ping is applied to a bidirectional LSP:

- The active side calculates ERO, signals LSP and runs LSP Self-ping

- The Passive side reverses ERO, signals LSP and runs another instance of LSP Self-ping

- Neither side forwards traffic through the LSP until local LSP Self-ping returns TRUE

The two LSP Self-ping sessions, mentioned above, are independent of one another. They are not required to have the same Session-ID. Each endpoint can forward traffic through the LSP as soon as the its local LSP Self-ping returns TRUE. Endpoints are not required to wait until both LSP Self-ping sessions have returned TRUE.

6. Rejected Approaches

In a rejected approach, the ingress LSR uses LSP-Ping to verify LSP readiness. This approach was rejected for the following reasons.

While an ingress LSR can control its control plane overhead due to LSP Ping, an egress LSR has no such control. This is because each ingress LSR can, on its own, control the rate of the LSP Ping originated by the LSR, while an egress LSR must respond to all the LSP Pings originated by various ingresses. Furthermore, when an MPLS Echo Request reaches an egress LSR it is sent to the control plane of the egress LSR, which makes egress LSR processing overhead of LSP Ping well above the overhead of its data plane (MPLS/IP forwarding). These factors make LSP Ping problematic as a tool for detecting LSP readiness to carry traffic when dealing with a large number of LSPs.

By contrast, LSP Self-ping does not consume any control plane resources at the egress LSR, and relies solely on the data plane of the egress LSR, making it more suitable as a tool for checking LSP readiness when dealing with a large number of LSPs.

In another rejected approach, the ingress LSR does not verify LSP readiness. Alternatively, it sets a timer when it receives an RSVP RESV message and does not forward traffic through the LSP until the timer expires. This approach was rejected because it is impossible to determine the optimal setting for this timer. If the timer value is set too low, it does not prevent black-holing. If the timer value is set too high, it slows down the process of LSP signalling and setup.

Moreover, the above-mentioned timer is configured on a per-router basis. However, its optimum value is determined by a network-wide behavior. Therefore, changes in the network could require changes to the value of the timer, making the optimal setting of this timer a moving target.

7. IANA Considerations

This memo request that IANA assign a UDP port from the user range (1024-49151) for LSP Self-ping.

8. Security Considerations

LSP Self-ping messages are easily forged. Therefore, an attacker can send the ingress LSR a forged LSP Self-ping message, causing the ingress LSR to terminate the LSP Self-ping session prematurely. In order to mitigate these threats, implementations SHOULD NOT assign Session-ID's in a predictable manner. Furthermore, operators SHOULD filter LSP Self-ping packets at network ingress points.

9. Acknowledgements

Thanks to Yakov Rekhter, Ravi Singh, Eric Rosen, Eric Osborne, Greg Mirsky and Nobo Akiya for their contributions to this document.

10. References

10.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[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.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", RFC 5036, October 2007.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, August 2011.

10.2. Informative References

[I-D.akiya-bfd-seamless-base]
Akiya, N., Pignataro, C., Ward, D., Bhatia, M., and J. Networks, "Seamless Bidirectional Forwarding Detection (S-BFD)", draft-akiya-bfd-seamless-base-03 (work in progress), April 2014.

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, August 2006.

Authors' Addresses

Ravi Torvi
Juniper Networks

Email: rtorvi@juniper.net

Ron Bonica
Juniper Networks

Email: rbonica@juniper.net

Ina Minei
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
U.S.A.

Email: inaminei@google.com

Michael Conn
Verizon

Email: michael.e.conn@verizon.com

Dante Pacella
Verizon

Email: dante.j.pacella@verizon.com

Luis Tomotaki
Verizon

Email: luis.tomotaki@verizon.com

Mark Wygant
Verizon

Email: mark.wygant@verizon.com