Routing area
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2019

K. Arora
S. Hegde
Juniper Networks Inc.

October 16, 2018

TTL Procedures for SR-TE Paths in Label Switched Path Traceroute Mechanisms
draft-arora-mpls-spring-ttl-procedures-srte-paths-00

Abstract

Segment routing supports the creation of explicit paths using adjacency-sids, node-sids, and anycast-sids. The SR-TE paths are built by stacking the labels that represent the nodes and links in the explicit path. A very useful Operations And Maintenance requirement is to be able to trace these paths as defined in [RFC8029]. This document specifies a uniform mechanism to support MPLS traceroute for the SR-TE paths when the nodes in the network are following uniform mode or short-pipe mode [RFC3443].

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 RFC 2119 [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 https://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on April 19, 2019.

Table of Contents

1. Introduction

The mechanisms to handle TTL procedures for SR-TE paths are described in ([RFC8287]). Section 7.5 of ([RFC8287]) defines the TTL manipulation procedures for short pipe model as the LSR initiating the traceroute SHOULD start by setting the TTL to 1 for the tunnel in the LSP's label stack it wants to start the tracing from, the TTL of all outer labels in the stack to the max value, and the TTL of all the inner labels in the stack to zero. However this mechanism has issues when the constituent tunnels are penultimate-hop-popping(PHP). c

Section 2 describes problems tracing SR-TE paths and the need for a specialized mechanism to trace SR-TE paths. Section 3 describes the solution applied to mpls echo request/response to trace adjacency-sids and node-sids trace SR-TE path in uniform model and short pipe model.

2. Problem with SR-TE Paths

The topology shown in Figure 1. illustrates a example network topology with SPRING enabled on each node.

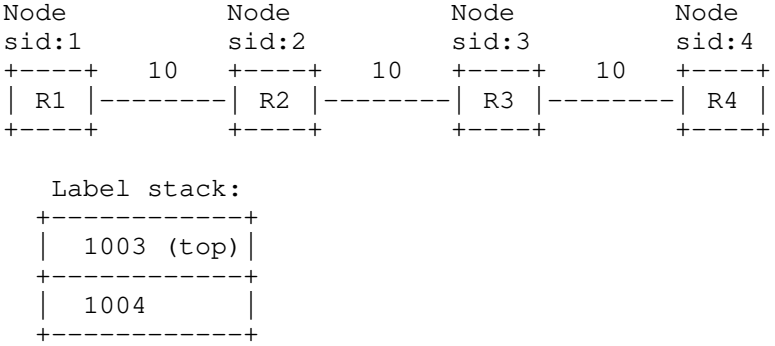

Figure 1: Example topology with SRGB 1000-2000

Consider an explicit path in the topology in Figure 1 from R1->R4 via R1->R2->R3->R4. The label stack to instantiate this path contains two node-sids 1003 and 1004. The 1003 label will take the packet from R1 to R3. The next label in the stack 1004 will take the packet from R3 to the destination R4. consider the mechanism below for the TTL procedures specified in RFC 8287 for short pipe model and uniform model for PHP LSPs.

Notation: ((X,Y>,(Z,W)) refers to a label stack whose top label stack entry has the label corresponding to the node-SID of X, with TTL Y, and whose second label stack entry has the label corresponding to the node-SID of Z, with TTL W.

According to the procedure in Section 7.5 of [RFC8287], the LSP traceroute is done as follows in short pipe model and uniform model:

2.1. Short Pipe model

Refer the diagram in Figure 1.

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp ping packet and R2 sends an echo reply to R1 with return code as 'transit'.

3. R1 receives the LSP Echo Reply from R2, and then sends next LSP Echo Request with label stack ((1003,2),(1004,0)).

4. R2 forwards packet to R3 as ((1004,0)) (i.e. R2 being PHP pops stack and does not propagate TTL)

5. R3 receives a packet with TTL=0 at the top of the stack. Receipt of a packet with TTL=0 will cause R3 to drop the packet.

RFC 8287 suggests that when R1's LSP Echo Request has reached the egress of the outer tunnel, R1 should being to trace the inner tunnel by sending a LSP Echo Request with label stack ((1003,2),(1004,1)). However there is no way for R1 to do that in this scenario, because R1 cannot tell when the egress of the outer tunnel has been reached.

2.2. Uniform Model

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp ping packet and R2 sends an echo reply to R1 with return code as 'transit'.

3. R1 receives the LSP Echo Reply from R2, and then sends next LSP Echo Request with label stack ((1003,2),(1004,0)).

4. It is expected that R2 should propogate the TTL of outer label to inner label before forwarding the packet to R3. However most of the PFEs implementations generally do not increase a label stack entry's TTL when they do TTL propagation. So when (1003,2) is popped, we might still end up with (1004,0) at R3, even if we have TTL propagation configured. Increasing the TTL of a traveling packet may not be a good practice.

5. R3 receives a packet with TTL=0 at the top of the stack. Receipt of a packet with TTL=0 will cause R3 to drop the packet.

So in either case (uniform model or short pipe model) traceroute may not work for SR-TE paths with PHP Lsps.

3. Detailed Solution For TTL procedures for SR-TE paths

3.1. P bit in DDMT TLV

DS flags has 4 unused bits from position '0' to '3'. This document uses bit '3' in DS flags of downstream mapping TLV.

3.2. Procedures for a PHP router of the tunnel being traced

When a LSR receives an echo request it MUST validates the outermost FEC in the echo request. LSR must set the 'P' bit in the DS flags of downstream mapping TLV if its a PHP router for the outermost FEC. Other cases it should work as explained in RFC8287 and RFC 8209

3.3. Procedures for a egress router of the tunnel being traced

When a LSR receives an echo request it MUST validates the outermost FEC in the echo request. If LSR is egress for the outermost FEC Then it MUST look for the next label in the FEC stack if exists any. If the LSP is the PHP router for the next FEC (next to outermost FEC in FEC stack if any), Then LSR MUST set 'P' bit in the downstream mapping TLV. Other cases it should work as explained in RFC8287 and RFC 8209

3.4. Procedures for a ingress router of the SR-TE path

When an ingress LSR receives an echo response with 'P' bit set in the DS flags of downstream mapping TLV, Then while sending next echo request Ingress LSR MUST increase the TTL value of inner label also (if exists) in addition to increasing the TTL value of the tunnel its tracing. Other cases it should work as explained in RFC8287 and RFC 8209

3.5. Example describing the solution

This section provides a detailed description of how PHP router helps ingress in handling TTL procedures for SR-TE paths. Below are the procedures performed by PHP router and ingress router to perform TTL procedure for mpls traceroute for SR-TE paths. Below solution works for both uniform model and short pipe model.

1. Ingress R1 sends mpls LSP Echo Request with label stack of ((1003,1),(1004,0)) to R2.

2. Since R2 receives mpls LSP Echo Request with TTL as 1 for outer most label, R2's local software processes the Lsp ping packet. R2's local software validates the outermost FEC and looking at the FEC R2 knows that its the PHP router for outermost FEC (Node-Sid R3).

3. R2 sets a bit in the DS flags in the DDMT TLV in echo response (P bit, One of the reserved bits).

4. When R1 looks at the echo response from R2 it sees P bit in DDMT TLV .

5. So R1 increment the TTL value of Node-R3 by 1 (make it 2) and TTL value of next element in the label stack also

6. R1 should send the next mpls LSP Echo Request with label stack ((1003,2),(1004,1)).

7. R2 being PHP pops the ouetrmost label from the label stack and forward the packet to R3 with with label (1004, 1)

8. R3 receives mpls LSP Echo Request with TTL as 1 for outer most label, R3's local software process the echo request.

9. R3 validates the outermost FEC and knows that R3 is the egress for outermost FEC (Node-Sid R3).

10. Since R3 is the egress for outermost FEC so R3 should look at the next FEC in the FEC stack (Node-Sid-R4) and identify if R3 is the PHP router for next FEC in the label stack. Since R3 is the PHP router for next FEC (Node-Sid R4) R3 should set 'P' bit in the in the DS flags in the DDMT TLV in echo response with return code as 'Egress'.

11. When R1 receives 'P' in the DDMT TLV as well as return code as egress then R1 knows that the ouetrmost tunnel is traced.

11. R1 should send the next mpls LSP Echo Request with label stack ((1003,2),(1004,2)) with FEC Node-Sid-R4 (Since its received Egress for ouetrmost FEC Node-Sid-R3).

12. R2 pops the first label from the label stack and R3 pops the second label from the label stack.

14. R4 receives an unlabelled packet with RA bit set in ip options. R4 delivers the packet to local software for processing.

15. R4's local software validates the ouetmost FEC as 'egress' and there is no more FEC in the FEC stack TLV.

16. R4 sends an echo reply with return code as egress.

17. R1 receives an echo reply with return code as egress for the last FEC in the FEC stack TLV and completes the traceroute.

4. Backward Compatibility

If the LSR with the proposed solution is the Ingress and all other LSR in the SR tunnel are not with the extension, Then no LSR is going to set 'P' bit so ingress LSR with new extension will work as per [RFC8029] and [RFC8287].If the LSR with the proposed extension is the one of the transit router and if its the PHP then it may set 'P' bit based on the section 3. Ingress may not react to the 'P' bit and traceroute will continue to work as per [RFC8029] and [RFC8287].

5. Security Considerations

TBD

6. IANA Considerations

IANA has created and now maintains a registry entitled "DS Flags". The registration policy for this registry is Standards Action [RFC5226]. IANA has made the following assignments: Bit Number Name Reference ---------- ---------- --------- 7 N: Treat as a Non-IP Packet [RFC8029] 6 I: Interface and Label Stack Object Request [RFC8029] 5 E: ELI/EL push indicator [RFC8012] 4 L: Label-based load balance indicator [RFC8012] 3 P: Penulimate Hop router 2-0 Unassigned

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[RFC8287] Kumar, N., Ed., Pignataro, C., Ed., Swallow, G., Akiya, N., Kini, S., and M. Chen, "Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes", RFC 8287, DOI 10.17487/RFC8287, December 2017, <https://www.rfc-editor.org/info/rfc8287>.

7.2. Informative References

[RFC3443] Agarwal, P. and B. Akyol, "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks", RFC 3443, DOI 10.17487/RFC3443, January 2003, <https://www.rfc-editor.org/info/rfc3443>.

Authors' Addresses

Kapil Arora
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: kapilaro@juniper.net

Shraddha Hegde
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: shraddha@juniper.net