Routing area
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2019

K. Arora
S. Hegde
Juniper Networks Inc.
S. Aldrin
Google
S. Litkowski
Orange Business Service
M. Durrani
Equinix

February 21, 2019

TTL Procedures for SR-TE Paths in Label Switched Path Traceroute Mechanisms
draft-arora-mpls-spring-ttl-procedures-srte-paths-01

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 will expire on August 25, 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 below.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). This document does not propose any change to ([RFC8287]) if the constituent tunnels are ultimate-hop-popping (UHP) or Egress LSR advertizes explicit NULL.

Section 2 describes problems in 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.

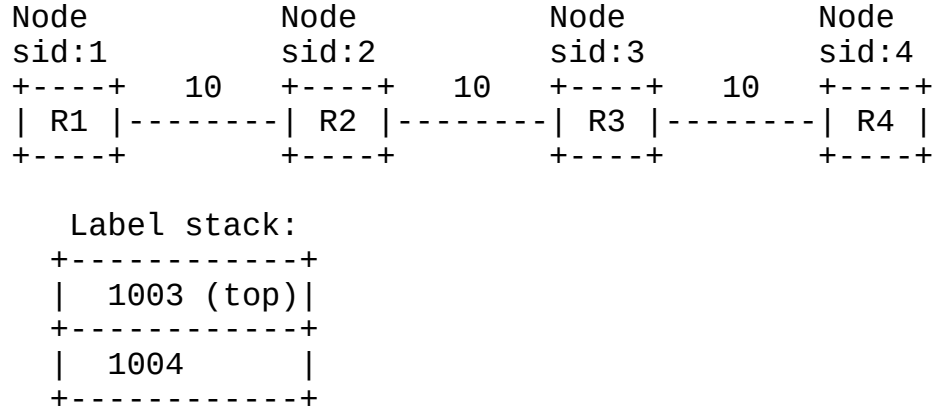

Figure 1: Example topology with SRGB 1000-2000

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 traceroute 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 the label 1003 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 may cause R3 to drop the packet or rate limit it.

6. Even if R3’s local software processes the packet and validates the FEC for 1003 and sends egress code in echo-reply, the next packet will have ((1003,255), (1004, 1)) which causes TTL to expire again on R3 as the 1003 label is popped at the penultimate.

RFC 8287 suggests that when R1’s LSP Echo Request has reached the egress of the outer tunnel, R1 should begin to trace the inner tunnel by sending a LSP Echo Request with label stack ((1003,255),(1004,1)). However, as explained in step 6, the traceroute procedure does not work correctly.

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 packet is not a good practice as it can result in forwarding loops.

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 or rate limit it.

6. Even if R3's local software processes the packet and validates the FEC for 1003 and sends egress code in echo-reply, the next packet will have ((1003,255), (1004, 1)) which causes TTL to expire again on R3 as the 1003 label is popped at the penultimate.

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.

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

When a LSR receives an echo request it MUST validate the outermost FEC in the echo request. LSR SHOULD 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 [RFC8029] and [RFC8287].

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

When a LSR receives an echo request it MUST validate the outermost FEC in the echo request. Egress cases should work as explained in [RFC8029] and [RFC8287].

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

When an ingress LSR receives an echo response it MUST behave as defined below depending on the return code in the echo response.

1. When an ingress LSR receives an echo response with return code as 8 (Label switched at stack-depth), Ingress LSR MUST check if the LSR that sent the echo response is PHP for the outermost FEC in the FEC

stack. If the LSR that sent the echo response is PHP for the outermost FEC 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 it is tracing. Ingress LSR can detect that LSR that sent the echo response is a PHP router for the outermost FEC, either by looking at 'P' bit set in the DS flags of downstream mapping TLV or if Ingress LSR has received LABEL '3' in the label stack TLV of downstream detailed mapping TLV. For all other cases ingress should work as explained in [RFC8029] and [RFC8287].

2. When an Ingress LSR receives an echo response with return code as 3 (Replying router is an egress for the FEC at stack-depth) for the outermost FEC and this is not the only FEC in the FEC stack, then ingress LSR SHOULD remove the outermost FEC from the FEC stack and send the next traceroute request with the same TTL value for all the labels in the label stack as the previous echo request. This will ensure the egress of the tunnel is visited twice, once as egress for top label and again as a transit for next tunnel.

3.1.4. 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 increments 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 outermost label from the label stack and forwards 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 processes the echo request.

9. R3 validates the outermost FEC and sends echo response to R1 with return code as the egress for outermost FEC (Node-Sid R3).

10. When R1 receives echo response with return code as egress, R1 should remove outermost FEC (Node-Sid R3) from the FEC stack and send the next echo request with the same TTL value as the previous one i.e ((1003,2),(1004,1)).

11. Since R3 is the PHP router for FEC (Node-Sid R4) in the label stack. R3 should set ’P’ bit in the in the DS flags in the DDMT TLV in echo response with return code as Transit.

12. R1 should send the next mpls LSP Echo Request with label stack ((1003,2),(1004,2)) with FEC Node-Sid-R4 .

13. 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 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.

3.2. Procedures for handling binding-sids

Inorder to provide greater scalability, network opacity, and service independence, SR architecture [RFC8402] defines a Binding SID (BSID). A Binding SID is bound to an SR policy which typically involves a list of SIDs. These Binding SIDs may appear in another SR Policy or may be used to steer service traffic from the service origin. The TTL handling mechanisms for MPLS traceroute procedures involving Binding SIDs is described below.

3.2.1. Uniform Model

When the node advertising the Binding SID is operating in uniform mode [RFC3443], it SHOULD send FEC stack change sub-TLV as in sec 4.5.1 of [RFC8029]. The ingress node SHOULD increment the TTL of

Binding SID label at every step until "egress" return code is sent for all the new FECs included due to FEC stack change and all the Tunnels replaced by the Binding SID are completely traced. It is required that all the label popping nodes involved in these tunnels MUST support uniform model and copy the TTL to bottom label when the label is popped.

3.2.2. Shortpipe Model

When the node advertising the Binding SID is operating in short pipe model [RFC3443], it SHOULD not send FEC stack change sub-TLV. The Binding SID is treated as single hop and the nodes internal to the Tunnel represented by Binding SID SHOULD NOT be traced.

4. Backward Compatibility

The extension proposed in this document is backward compatible with procedures described in [RFC8029] and [RFC8287]. 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. Acknowledgements

Thanks to Przemyslaw Krol for careful review and comments.

8. References

8.1. 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>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

8.2. 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

Sam Aldrin
Google

Email: aldrin.ietf@gmail.com

Stephane Litkowski
Orange Business Service

Email: stephane.litkowski@orange.com

Muhammad Durrani
Equinix

Email: mdurrani@equinix.com