6man Working Group
Internet-Draft
Intended status: Standards Track
Expires: 17 December 2025

Z. Ali
Cisco Systems, Inc.
K. Szarkowicz
Juniper Networks
S. Joy
Cisco Systems, Inc.

15 June 2025

ICMP Error Handling in SRv6 based VPN Networks
draft-ali-6man-srv6-vpn-icmp-error-handling-00

Abstract

The document specifies procedures for handling ICMP error in SRv6-based Virtual Private Network (VPN).

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 17 December 2025.

Copyright Notice

Table of Contents

1. Terminology and Reference Topology 2
2. Introduction 3
3. CE-to-CE Traceroute 5
3.1. CE-to-CE Traceroute from IPv6 customer edge 5
3.1.1. Illustration 5
3.2. CE-to-CE Traceroute from IPv4 Customer Edge 8
3.2.1. Illustration 8
4. Security Considerations 12
5. IANA Considerations 12
6. References 12
6.1. Normative References 12
6.2. Informative References 12
Appendix A. Acknowledgements 13
Authors' Addresses 13

1. Terminology and Reference Topology

Throughout the document, the following terminology and simple topology is used for illustration.

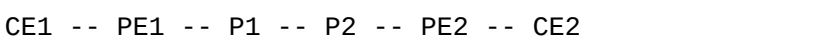

Figure 1 Reference Topology

In the reference topology:

CE1 and CE2 are Customer Edge devices of IPv4 or IPv6 data plane capability.

CE1::1 is an IPv6 address assigned to CE1.

CE2::1 is an IPv6 address assigned to CE2.

CE1.1.1.1 is an IPv4 address assigned to CE1.

CE2.2.2.2 is an IPv4 address assigned to CE2.

PE1 and PE2 are provider edge nodes.

Node PEj has an IPv6 loopback address PEj::1/128.

P1 and P2 are provider core nodes.

Node Pj has an IPv6 loopback address Pj::1/128.

Node Pj might have or might not have IPv4 address; if IPv4 address is assigned, it has an IPv4 loopback address Pj.j.j.j/32.

(SA1,DA1,HL=x,NH=IPv6)(SA2,DA2,HL=y,NH=UDP)(UDP payload) represents an IPv6 packet with:

* Outer IPv6 header with source address SA1, destination address DA1, and IPv6 as the next header. The hop limit of the packet is x.

* IPv6 follows the inner IPv6 header with source address SA2 and destination address DA2 with hop limit of y and NH = UDP.

* (UDP payload) represents the UDP payload of the packet.

(SA1,DA1,HL=x,NH=IPv4)(SA2,DA2,TTL=y)(UDP payload) represents a similar IPv6 packet with an IPv4 inner header with TTL of y.

Please note that traceroute to a SID is exemplified using UDP probes. However, the procedure is equally applicable to other implementations of the traceroute mechanism. The UDP-encoded message to traceroute a SID uses the UDP ports assigned by IANA for "traceroute use."

2. Introduction

SRv6 refers to Segment Routing instantiated on the IPv6 data plane [RFC8402].

SRv6-based VPN (SRv6-VPN) refers to the creation of VPN between the PEs leveraging the SRv6 dataplane [RFC9252].

This document specifies procedures for handling ICMP errors in SRv6-VPN where the provider network deploys Uniform Model [RFC3443]. The Uniform Model makes all the nodes that the customer packet traverses visible to nodes outside the provider core. This is acheived by allowing the time-to-live (TTL) or hop-limit (HL) fields to propagate during the SRv6 encapsulation of the customer packet.

The issue addressed by this draft is illustrated using a traceroute scenario using the reference topology. CE1 and CE2 are IPv6-capable routers.

Consider processing the following traceroute packet at P1:

```
CE1_out : (CE1::1, CE2::1, HL=2, NH=UDP)(Traceroute probe)
```

Please note, in PE1 HL propagation is enabled, thus the value of HL in the outer header is propagated from the inner header, i.e.,:

```
PE1_out : (PE1::1, PE2:DT6::, HL=1, NH=IPv6)
            ((CE1::1, CE2::1, HL=1, NH=UDP)(Traceroute probe))
```

When the packet comes to P1, the hop limit expires at P1. Without ICMP tunneling enabled on P1, P1 implements the standard procedure from RFC9259, and therefore sends an ICMPv6 packet towards PE1 as follows:

```
P1_out : (P1::1, PE1::1, HL=64, NH=ICMPv6)
         (ICMPv6, Time Exceeded,
         [copy of the invoking packet =
         (PE1::1, PE2:DT6::, HL=1, NH=IPv6)
         ((CE1::1, CE2::1, HL=1, NH=UDP)(Traceroute probe))])
```

Rules for processing the above-mentioned ICMP error message at PE1 is a local decision at PE1. However, the Processing the message at PE1 requires context that the packet was not sourced at PE1. In addition to the context at PE1, PE1 needs to peek into the copy of the invoking packet to generate an ICMPv6 response for CE1. If CE1 is an IPv4 node, PE1 must also ensure it generates an ICMPv4 message to CE1. While processing the above-mentioned packet at PE1 is complex, the advantage of this approach is that the node generating ICMP error (P1 in this example) can be a node without SRv6 capability.

This draft proposes an alternate procedure for ICMP Error Handling in SRv6 based VPN Networks that does not require support from the PE1 node. Specifically, the SRv6 capable P node sends (tunneles) the error towards the end of SRv6 cloud using the VPN SID of the egress PE. The VPN SID of the egress PE is obtained from the invoking packet. This is similar to handling such cases in MPLS network where message is sent (tunneled) to end of the LSP. Once the packet reaches the egress PE, ICMP packet is decapsulated. Subsequent forwarding table lookup on ICMP packet DA results in the ICMPv6 packet being forwarded to the ingress PE using the VPN SID of the Ingress PE. The ingress PE decapsulates the packet and send it to the intended CE.

The CE nodes can be IPv6 or IPv4 nodes.

Without loss of generality, the rest of the document focuses on ce-to-ce traceroute in SRv6-VPN. However, the procedure applies to all ICMPv6 error types handling in SRv6-based VPN.

3. CE-to-CE Traceroute

[RFC9259] describes how the IPv6 OAM mechanisms can be used to diagnose problems within the SRv6 networks. For example, [RFC9259] explains how the existing traceroute mechanisms for PE-to-PE traceroute. However, [RFC9259] does not describe a mechanism for a CE-to-CE traceroute when the provider network deploys Uniform Model [RFC3443].

3.1. CE-to-CE Traceroute from IPv6 customer edge

CE-to-CE traceroute from IPv6 customer edge when the customer enables HL propagation, as well as ICMP tunneling, works in a way similar to its MPLS counterpart. The P node where HL expires generates an ICMPv6 Time Exceeded message for the CE that initiated the traceroute request. The ICMP error is sent (tunneled) towards the end of the SRv6 cloud using the VPN SID of the egress PE (similar to MPLS, where it is sent to the end of the LSP). The VPN SID of the egress PE is obtained from the packet that the ICMP error invoking packet. Once the packet reaches egress PE, the ICMPv6 packet is decapsulated. Subsequent forwarding table lookup on the ICMPv6 packet destination address results in the ICMPv6 packet being forwarded to the CE that initiated the traceroute request.

3.1.1. Illustration

This section illsutares the CE-to-CE Traceroute from IPv6 customer edge in an SRv6-VPN network where the provider network deploys Uniform Model [RFC3443]

Using the reference topology, consider a scenario where CE1 initiates a traceroute request to CE2. CE1 and CE2 are both IPv6 routers.

The tracerout packet with hop limit set to 2 as it leaves CE1 is encoded as follows:

CE1_out : (CE1::1, CE2::1, HL=2, NH=UDP)(Traceroute probe)

On PE1, HL propagation is enabled, thus HL value from inner packet is propagated to outer header.

```
PE1_out : (PE1::1, PE2:DT6::, HL=1, NH=IPv6)
            ((CE1::1, CE2::1, HL=1, NH=UDP)(Traceroute probe))
```

Note: In some scenarios, a packet might have multiple transport outer IPv6 headers preceding the customer's inner IPv6 header. TI LFA is an example of such a scenario. Specifically, when SRv6 encapsulation mode is used to construct backup TI LFA next-hop, an additional IPv6 header is pushed on the packet. The proposed procedure handles such scenarios. The traceroute displays the TI-LFA backup path when it is active. However, the illustration does not cover such scenarios.

When the packet comes to P1, the hop limit expires at P1. Based on a local policy at P1, P1 doesn't follow the standard procedure described in RFC9259 and sends an ICMPv6 packet towards CE1 as follows:

```
P1_out : (PE1::1:,PE2:DT6::,HL=64;NH=IPv6)
               ((P1::1,CE1::1;HL=64,NH=ICMPv6)
                 (ICMPv6,Time Exceeded,
                   [copy of the invoking packet =
                   ((CE1::1, CE2::1;HL=1,NH=UDP)
                   (Traceroute probe)])
```

Essentially, the encapsulation of the generated ICMPv6 packet is constructed as follows:

* Copy of the original inner IPv6 header and packet (original Traceroute probe).

* ICMPv6 packet with:

  SA = IPv6 address of local node

  DA = IPv6 source address of the original inner IPv6 header

  HL = 64

* IPv6 header with:

  SA = IPv6 source address of the original IPv6 outer header

  DA = IPv6 destination address of the original IPv6 outer header

  HL = 64

When the original packet has multiple "transport" outer IPv6 headers, the procedure is as follows (from inner to outer):

* Copy of the original most inner IPv6 header and packet (original Traceroute probe).

* ICMPv6 packet with:

  SA = IPv6 address of local node

  DA = IPv6 source address of the original most inner IPv6 header

  HL = 64

* Copy of all "transport" outer IPv6 headers from the original packet, except for the most outer IPv6 header.

* IPv6 header with:

  SA = IPv6 source address of the original IPv6 outer header

  DA = IPv6 destination address of the original IPv6 most outer header

  HL = 64

The encapsulated packet as it leaves P2 is as follows:

```
P2_out: (PE1::1, PE2:DT6::, HL=63, NH=IPv6)(
        (P1::1, CE1::1, HL=64, NH=ICMPv6)
        (ICMPv6, Time Exceeded,
          [copy of the invoking packet =
            ((CE1::1, CE2::1, HL=1, NH=UDP)
            (Traceroute probe))
          ])
      )
```

When a packet arrives at PE2, it performs the local END.DT6 (or END.DT46) processing, decapsulating the packet and processing the inner IPv6 header. As the destination in the inner packet is CE1::1, PE2 encapsulates it so it can be delivered to CE1::1.

Assuming HL/TTL propagation is enabled to PE2, PE2 generates the following packet:

```
PE2_out: (PE2::1, PE1:DT6::, HL=62, NH=IPv6)(
          (P1::1, CE1::1, HL=62, NH=ICMPv6)
          (ICMPv6, Time Exceeded,
            [copy of the invoking packet =
              (CE1::1, CE2::1, HL=1, NH=UDP)(Traceroute probe)
            ])
        )
```

The packet is IPv6 routed back to PE1. When this packet arrives at PE1, PE1 performs the local END.DT6 (or END.DT46) processing, decapsulates the packet, and processes the inner IPv6 header. Assuming HL/TTL propagation is enabled on PE1, the inner packet is IPv6 routed back to the CE1 node as follows:

```
PE1_out: (P1::1, CE1::1, HL=59, NH=ICMPv6)
         (ICMPv6, Time Exceeded,
           [copy of the invoking packet =
             ((CE1::1, CE2::1, HL=1, NH=UDP)(Traceroute probe))
           ])
```

CE1 processes the hop limit exceeded packet sourced at P1.

CE1 generates a new Traceroute probe with incremented HL. This time, the packet expires at P2. The processing of the packet at P2 is similar to the packet processing at P1.

3.2. CE-to-CE Traceroute from IPv4 Customer Edge

The CE-to-CE traceroute from the IPv4 customer edge works in the same fashion as the CE-to-CE traceroute from the IPv6 customer edge. The exception is that it is assumed the P node, where TTL expiry occurs, is capable of generating an ICMPv4 Time Exceeded message. That ensures that the IPv4 CE can consume the ICMP message properly. For cases when the provider router is not capable of generating an ICMPv4 packet, we recommend not enabling TTL propagation.

3.2.1. Illustration

This section outlines CE-to-CE traceroute for IPv4 customer edge. It uses the reference topology in Section 1.

Consider the following traceroute packet received at P1 and its processing there:

```
CE1_out: (CE1.1.1.1, CE2.2.2.2, TTL=2)(Traceroute probe)
```

On PE1, TTL/HL propagation is enabled; thus, the TTL value from the inner packet is propagated to the outer header:

```
PE1_out: (PE1::1, PE2:DT4::, HL=1, NH=IPv4)
         ((CE1.1.1.1, CE2.2.2.2, TTL=1)
         (Traceroute probe))
```

When the packet reaches P1, its hop limit expires. With ICMP tunneling enabled on P1, it does not follow the standard procedure described in RFC9259 but instead generates and sends an ICMPv4 packet destined to CE1 tunneled via PE2.

P1 is assumed to be a node capable of generating ICMPv4. P1 will inspect the payload of the invoking packet and decide to generate an ICMPv4 Time Exceeded message. However, P1 might or might not have a local IPv4 address that could be used as the source address of the ICMPv4 Time Exceeded message.

If P1 has a local IPv4 address, it generates and sends an ICMPv4 packet as follows:

```
P1_out: (PE1::1, PE2:DT4::, HL=64, NH=IPv4)(
         P1.1.1.1, CE1.1.1.1, TTL=64, ICMPv4 Time Exceeded,
         [copy of the invoking packet =
           (CE1.1.1.1, CE2.2.2.2, TTL=1)(Traceroute probe)])
```

If P1 has no local IPv4 address, it sends:

```
P1_out: (PE1::1, PE2:DT4::, HL=64, NH=IPv4)(
         192.0.0.8, CE1.1.1.1, TTL=64, ICMPv4 Time Exceeded, NIO=P1::1,
         [copy of the invoking packet =
           (CE1.1.1.1, CE2.2.2.2, TTL=1)(Traceroute probe)])
```

P1 uses 192.0.0.8 as the dummy address for the IPv4 source of the ICMPv4 message, in accordance with Section 4.8 of RFC 7600. It also attaches an NIO (Node Identification Object) to the ICMPv4 message, as specified in Section 3 of [I-D.ietf-intarea-extended-icmp-nodeid]. The NIO provides information about P1’s real IPv6 address.

Encapsulation of the ICMPv4 packet generated at P1 is as follows (from inner to outer):

* Copy of the original inner IPv4 header and packet (original Traceroute probe).

* ICMPv4 packet with:

SA = IPv4 address of local node (if exists), or 198.0.0.8 (if local IPv4 address doesn’t exist)

DA = IPv4 source address of the original inner IPv4 header

TTL = 64

NIO = IPv6 address of local node (only if IPv4 address doesn’t exist)

* IPv6 header with:

SA = IPv6 source address of the original IPv6 outer header

DA = IPv6 destination address of the original IPv6 outer header

HL = 64

When the original packet has multiple "transport" outer IPv6 headers, the procedure is:

* Copy of the original most inner IPv4 header and packet (original Traceroute probe).

* ICMPv4 packet with:

SA = IPv4 address of local node (if exists), or 198.0.0.8 (if local IPv4 address doesn’t exist)

DA = IPv4 source address of the original inner IPv4 header

TTL = 64

NIO = IPv6 address of local node (only if IPv4 address doesn’t exist)

* Copy of all "transport" outer IPv6 headers from the original packet.

* IPv6 header with:

SA = IPv6 source address of the original IPv6 outer header

DA = IPv6 destination address of the original IPv6 most outer header

HL = 64

Examples:

```
P2_out: (PE1::1, PE2:DT4::, HL=63, NH=IPv4)(
         P1.1.1.1, CE1.1.1.1, TTL=64, ICMPv4 Time Exceeded,
         [copy of the invoking packet =
           (CE1.1.1.1, CE2.2.2.2, TTL=1)(Traceroute probe)])

P2_out: (PE1::1, PE2:DT4::, HL=63, NH=IPv4)(
         192.0.0.8, CE1.1.1.1, TTL=64, ICMPv4 Time Exceeded, NIO=P1::1,
         [copy of the invoking packet =
           (CE1.1.1.1, CE2.2.2.2, TTL=1)(Traceroute probe)])
```

When a packet arrives at PE2, it performs the local END.DT4 (or END.DT46) processing, decapsulates the packet, and processes the inner IPv4 header. As the destination is CE1.1.1.1, PE2 re-encapsulates the packet for delivery to CE1.

Assuming TTL propagation is enabled, PE2 generates the following:

```
PE2_out: (PE2::1, PE1:DT4::, HL=62, NH=IPv4)(
          P1.1.1.1, CE1.1.1.1, TTL=62, ICMPv4 Time Exceeded,
          [copy of the invoking packet =
            (CE1.1.1.1, CE2.2.2.2, TTL=1)(Traceroute probe)])

PE2_out: (PE2::1, PE1:DT4::, HL=62, NH=IPv4)(
          192.0.0.8, CE1.1.1.1, TTL=62,
          ICMPv4 Time Exceeded, NIO=P1::1,
          [copy of the invoking packet =
            (CE1.1.1.1, CE2.2.2.2, TTL=1)(Traceroute probe)])
```

Once received at PE1, it performs END.DT4 (or END.DT46), decapsulates the packet, and routes it to CE1:

```
PE1_out: (P1.1.1.1, CE1.1.1.1, TTL=59, ICMPv4 Time Exceeded,
          [copy of the invoking packet =
            (CE1.1.1.1, CE2.2.2.2, TTL=1)(Traceroute probe)])

PE1_out: (192.0.0.8, CE1.1.1.1, TTL=59,
          ICMPv4 Time Exceeded, NIO=P1::1,
          [copy of the invoking packet =
            (CE1.1.1.1, CE2.2.2.2, TTL=1)(Traceroute probe)])
```

CE1 processes the ICMPv4 Time Exceeded message sourced at P1. If the packet has IPv4 source address 192.0.0.8 with an NIO, CE1 does not use the IPv4 address but instead uses the NIO for operator display. If CE1 does not support NIO, it displays the IPv4 address 192.0.0.8.

CE1 then generates a new traceroute probe with incremented TTL. This time, the packet expires at P2. P2’s processing of the packet is similar to that of P1.

4. Security Considerations

This document does not impose any additional security challenges to be considered beyond the security threats described in [RFC9252].

5. IANA Considerations

This document has no IANA actions.

6. References

6.1. 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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

6.2. 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>.

[RFC9252] Dawra, G., Ed., Talaulikar, K., Ed., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)", RFC 9252, DOI 10.17487/RFC9252, July 2022, <https://www.rfc-editor.org/info/rfc9252>.

[RFC9259] Ali, Z., Filsfils, C., Matsushima, S., Voyer, D., and M. Chen, "Operations, Administration, and Maintenance (OAM) in Segment Routing over IPv6 (SRv6)", RFC 9259, DOI 10.17487/RFC9259, June 2022, <https://www.rfc-editor.org/info/rfc9259>.

Appendix A. Acknowledgements

The authors would like to thank Syed Kamran Raza.

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

Krzysztof Grzegorz Szarkowicz
Juniper Networks
Email: kszarkowicz@juniper.net

Sijo Joy
Cisco Systems, Inc.
Email: sijoy@cisco.com