6man
Internet-Draft
Intended status: Standards Track
Expires: 17 September 2026

B. Varga
Ericsson
J. Halpern
JMH

16 March 2026

ICMP Error Handling for VPNs in SRv6 Networks
draft-varhal-6man-icmp-srv6-vpn-01

Abstract

This document specifies ICMP error handling in SRv6-based Virtual Private Networks, that support direct localization of failures. It provides a solution for connectivity check and fault localization without adding complexity to P nodes and keeps P nodes service agnostic. ICMP processing is changed only on ingress PE nodes and gains from adding VPN-specific information to the SRv6 encapsulated packet. Egress PE nodes are not involved in the forwarding of the ICMP error messages. Therefore, the solution provides visibility upto the failure even if ingress PE to egress PE connectivity is broken within the SR domain.

Table of Contents

1. Introduction

Troubleshooting is an essential part of all IP networks. IP VPN service of transport networks always represented a special challenge to provide ICMP error handling, as the hosts in a VPN are not part of the transport network.

For VPNs the main challenge to use ICMP for connectivity check and fault localization is that network internal nodes (a.k.a. P routers) are not service (i.e., VPN) aware. Therefore, P routers cannot route VPN-specific ICMP error messages back to the original source of the packet, that triggered the generation of the ICMP error message. Furthermore, in SRv6 networks the P routers may be IPv6-only (i.e., may not support the protocol (e.g. IPv4) or address space used by the VPN clients).

The Uniform Model defined in [RFC3443] allows visibility of traversed nodes outside the provider network.

This document proposes a solution that is capable to allow (1) ping or trace for VPN endpoints with transport network visibility; and (2) find broken link or node within the transport network (e.g., SRv6 domain).

2. Terminology

2.1. Terms Used in This Document

This document uses the Segment Routing terminology established in [RFC8402] and in [RFC9252]. The reader is assumed to be familiar with those documents and their terminology.

The following terms used within this document are defined in [RFC8402]: Segment Routing, SR domain, Segment ID (SID), SRv6, SRv6 SID, Active Segment, and SR Policy.

The following terms used within this document are defined in [RFC8754]: Segment Routing Header (SRH).

The following terms used within this document are defined in [RFC8986]: NH (next header), SL (the Segments Left field of the SRH), FIB (Forwarding Information Base), SA (Source Address), DA (Destination Address), and SRv6 Endpoint Behavior.

2.2. 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.

3. ICMP Error Handling for SRv6-VPNs

3.1. Overview

While a solution for diagnostics in MPLS VPNs has been created, the solution designed for MPLS based VPN ping or traceroute has many inherited drawbacks. MPLS technology has its special encapsulation, i.e., the MPLS header is a label stack. In case of MPLS, P routers have no options to identify the ingress of the MPLS tunnel, as labels in the header point towards the network egress point. This characteristic restricts the possible solutions to provide VPN-specific ICMP handling in MPLS networks and resulted in involving of egress-PE nodes in the forwarding of the ICMP error messages.

IPv6 encapsulation used by SRv6 has an IP SA field referring to the originator of the IP packet, i.e., the ingress endpoint of the SRv6 tunnel. Therefore the MPLS restriction does not have to apply for SRv6 networks. The solution described here takes advantages from this presence of the ingress endpoint information to provide an optimal method and not to change [RFC4443] procedures for P nodes.

Node functions in the described method are as follows:

1. Ingress PE (ingress node of the SRv6 tunnel): VPN packet encapsulation follows [RFC3443] Uniform model. The node adds VPN-specific information to the encapsulated packet (i.e., IP SA=VPN-specific-SID of the Ingress PE) and forwards it over the SRv6 network.

2. P node = Originator of the ICMP error (within the SRv6 domain): it does standard [RFC4443] operation, so an ICMP error message is sent to the originator (i.e., the ingress PE) of the SRv6 encapsulated packet, that caused the ICMP message generation (e.g., when the Hop Limit of the packet expired).

3. Ingress PE: it processes the ICMP error message and forwards it to the original source of the (payload) packet, what is located within the VPN context. This processing is done by a VPN-associated-ICMP-process-function and is described in detail in Section 3.2.

3.2. Details of ICMP Error Handling for VPNs in SRv6 Networks

Figure 1 shows the reference topology used to describe the ICMP error handling.

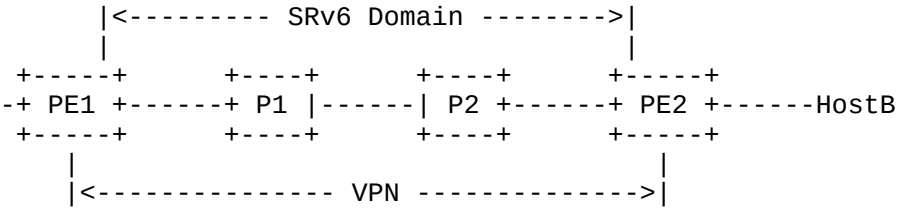

Figure 1: ICMP Error Handling for VPNs in SRv6 Networks: Reference Topology

Packet processing works as follows:

1. HostA sends a packet to HostB.

2. PE1 encapsulates the packet in an SRv6 tunnel (using the Uniform model). The IP SA of the encapsulation is a VPN-specific SID of PE1.

3. Encapsulated packet reaches P2 where Hop Limit expires.

4. P2 generates an ICMP Error Message and sends it to PE1, using the VPN-specific SID as an IP DA.

5. PE1 processes the ICMP Error Message according to its VPN-associated-ICMP-process-function and identifies the related VPN instance.

6. PE1 sends the processed ICMP Error Message to HostA.

7. HostA is informed about the Hop Limit expire event and its network location (i.e., P2).

The VPN-specific SID of PE1 refers to the VPN instance where the prefixes of the VPN can be looked up. The VPN-specific SID is allocated by the PE. SIDs processing already defines the upper-layer header steps (as per [RFC8986], section 4.1.1). The upper-layer = ICMPv6, therefore there is no need for extra parsing rules. There might be no need for extra SID allocation for a VPN. The solution uses a SID per VPN, what is allocated for the VPN service. More specifically for a VPN service the PE node can allocate SID(s) per-prefix (e.g., End.DX6) or per-vrf (e.g., End.DT6). The solution uses a per-vrf SID (e.g., End.DT6) in the IP SA of the SRv6 encapsulated packets.

For more sophisticated VPN configurations (e.g., Hub-and-Spoke VPN) where multiple VRFs (and SIDs) are configured for a given VPN, the VPN specific SID of PE1 always refers to the VRF instance (and its per-vrf SID) where the prefixes of the connected customer site(s) can be looked up.

As the locator part of the VPN-specific SID is routable within the SRv6 domain other PE and P nodes of the SRv6 domain can send/route packets to it.

The SRv6 encapsulation process on the ingress PE node needs several input information to construct the outer SRv6 header. One group of information is related to the IP DA and the SRH part of the SRv6 encapsulation. They are derived from the remote service information (e.g., VPN SID on the egress PE) and the SR policy (if exists). The SR policy defines the path to which an ingress PE node steers a packet flow. Applying a SR policy means to select the path (e.g., defined by a SID list) and placing the path descriptors into the IP DA and the SRH fields of the outer SRv6 encapsulation.

Another group of information is needed as well for the SRv6 encapsulation, like IP SA, Traffic Class, FlowLabel, HopLimit, NextHeader. They are derived by various local functionalities. The here described solution impacts only the selection of the IP SA. As per [RFC9256] the source IP MUST resolve to a unique node in the SRv6 domain, what is fulfilled by the above described VPN-specific SID. All other fields are defined by related RFCs. For example, Traffic Class might be copied from the inner packet, FlowLabel might be locally generated, etc.

The VPN-associated-ICMP-process-function operation contains the following steps:

1. It processes the received ICMPv6 error message (originated e.g., from a P node within the SRv6 domain).

2. It identifies the related VPN, based on the VPN-specific IP SA value in the SRv6 encapsulation of the received ICMPv6 error message.

3. It modifies the ICMP error message:

    * It removes the SRv6 domain specific encapsulation/header(s) of the received ICMPv6 error message.

    * It identifies the VPN-specific source of the original packet that caused the ICMPv6 error message, based on the invoking packet header part of the ICMPv6 error message payload.

    * It removes the SRv6 domain specific header(s) from the invoking packet header part of the ICMPv6 error message payload.

    * It creates a new header for the ICMP error message, where the IP SA refers to the Originator-of-the-ICMPv6-error-message and the IP DA=SourceIP-of-the-invoking-packet.

4. Forwards the modified ICMP error message according to the local VPN routing table (VRF).

The VPN-associated-ICMP-process-function may translate the IP address of the Originator-of-the-ICMPv6-error-message (e.g., a P node) to limit the VPN-specific visibility characteristics. For example, if the SRv6 domain operator does not want to export the real NodeIP or SID values used by the SRv6 domain nodes.

3.3. VPNv6 illustration

This section illustares a VPNv6 Traceroute from a customer host.

```
HostA sends the following traceroute packet to HostB.
HostA_out : (A::1, B::1, HL=3, NH=UDP)
            (Traceroute probe)

PE1 encapsulates in SRv6. In PE1 HL propagation is enabled.
PE1_out   : (PE1:VPN1::, PE2:DT6::, HL=2, NH=IPv6)
            ((A::1, B::1, HL=2, NH=UDP)
             (Traceroute probe))

P1 forwards the SRv6 packet.
P1_out    : (PE1:VPN1::, PE2:DT6::, HL=1, NH=IPv6)
            ((A::1, B::1, HL=2, NH=UDP)
             (Traceroute probe))

Hop limit expires at P2. P2 implements the standard procedure
from [RFC4443] and generates an ICMPv6 error message.
P2_out    : (P2::1:, PE1:VPN1::, HL=64; NH=ICMPv6)
            (ICMPv6, Time Exceeded,
             [copy of the invoking packet =
              (PE1:VPN1::, PE2:DT6::, HL=1, NH=IPv6)
              ((A::1, B::1, HL=2, NH=UDP)(Traceroute probe))
             ])

P1 forwards the ICMPv6 packet.
P1_out    : (P2::1:, PE1:VPN1::, HL=63; NH=ICMPv6)
            (ICMPv6, Time Exceeded,
             [copy of the invoking packet =
              (PE1:VPN1::, PE2:DT6::, HL=1, NH=IPv6)
              ((A::1, B::1, HL=2, NH=UDP)(Traceroute probe))
             ])
```

PE1 modifies the received ICMPv6 packet, by removing the SRv6 encapsulation related information. Invoking packet specific VPN service is explicitly identified based on the IP SA of the received ICMPv6 error message.

```
PE1_out   : (P2::1, A::1, HL=62, NH=ICMPv6)
            (ICMPv6, Time Exceeded,
             [copy of the invoking packet =
              (A::1, B::1, HL=2, NH=UDP)(Traceroute probe)
             ])
```

3.4. Dealing with IPv4-VPNs

In case of IPv4-VPN service the VPN-associated-ICMP-process-function operates as follows (v4/v6 are noted for clarity):

1. It processes the received ICMPv6 error message (originated e.g., from a P node within the SRv6 domain).

2. It identifies the related VPN, based on the VPN-specific IP SA value in the SRv6 encapsulation of the received ICMPv6 error message.

3. It synthesizes an ICMPv4 error message based on the received ICMPv6 error message:

    * It identifies the VPNv4 specific source of the original IPv4 packet that caused the ICMPv6 error message, based on the invoking packet header parts of the ICMPv6 error message payload.

    * It creates the header for the ICMPv4 error message, in accordance with [RFC7600] (Section 4.8) and [I-D.ietf-intarea-extended-icmp-nodeid] (Section 3), i.e., IPv4 SA=192.0.0.8, Node Identification Object containing the IPv6 SA of the ICMPv6 error message and IPv4 DA=IPv4-SA-of-the-original-packet.

4. Forwards the modified ICMPv4 error message according to the local VPNv4 routing table (VRF).

When PE node is aware of the IPv4 address of the SRv6 node that generated the ICMPv6 error message, then the PE node may use it as the IPv4 SA of the synthesized ICMPv4 message. How the PE node is aware of that information is out-of-scope in this document.

3.5. VPNv4 illustration

This section illustares a VPNv4 Traceroute from a customer host.

```
HostA sends the following traceroute packet to HostB.
HostA_out : (A.1.1.1, B.2.2.2, TTL=3, Prot=UDP)
            (Traceroute probe)

PE1 encapsulates in SRv6. In PE1 HL propagation is enabled.
PE1_out   : (PE1:VPN1::, PE2:DT6::, HL=2, NH=IPv4)
            ((A.1.1.1, B.2.2.2, TTL=2, Prot=UDP)
             (Traceroute probe))

P1 forwards the SRv6 packet.
P1_out    : (PE1:VPN1::, PE2:DT6::, HL=1, NH=IPv4)
            ((A.1.1.1, B.2.2.2, TTL=2, Prot=UDP)
             (Traceroute probe))

Hop limit expires at P2. P2 implements the standard procedure
from [RFC4443] and generates an ICMPv6 error message.
P2_out    : (P2::1:, PE1:VPN1::, HL=64; NH=ICMPv6)
            (ICMPv6, Time Exceeded,
             [copy of the invoking packet =
              (PE1:VPN1::, PE2:DT6::, HL=1, NH=IPv4)
              ((A.1.1.1, B.2.2.2, TTL=2, Prot=UDP)(Traceroute probe))
             ])

P1 forwards the ICMPv6 packet.
P1_out    : (P2::1:, PE1:VPN1::, HL=63; NH=ICMPv6)
            (ICMPv6, Time Exceeded,
             [copy of the invoking packet =
              (PE1:VPN1::, PE2:DT6::, HL=1, NH=IPv4)
              ((A.1.1.1, B.2.2.2, TTL=2, Prot=UDP)(Traceroute probe))
             ])
```

PE1 processes the received ICMPv6 packet and removes the SRv6 encapsulation related information. Invoking packet specific service is explicitly identified based on the IP SA of the received ICMPv6 error message. An ICMPv4 packet is synthetized.

```
PE1_out   : (192.0.0.8, A.1.1.1, TTL=62, Prot=ICMPv4)
            (ICMPv4, Time Exceeded, NIO=P2::1,
             [copy of the invoking packet =
              (A.1.1.1, B.2.2.2, TTL=2, Prot=UDP)(Traceroute probe)
             ])
or when IPv4 address of P2 is known by PE1
PE1_out   : (P2.2.2.2, A.1.1.1, TTL=62, NH=ICMPv4)
            (ICMPv4, Time Exceeded,
             [copy of the invoking packet =
              (A.1.1.1, B.2.2.2, TTL=2, Prot=UDP)(Traceroute probe)
             ])
```

3.6. Multi-level encapsulations

Some network scenarios result in a packet having multiple transport outer IPv6 headers preceding the customer’s inner IP header. For example in TI-LFA scenarios within the SRv6 domain. The solution described in this document handles TI-LFA scenarios and a traceroute may display the TI-LFA backup path when activated.

Note: other multiple encapsulation scenarios need further discussions by the WG.

3.7. Characteristics of the solution

ICMP Error Handling for VPNs in SRv6 Networks has the following characteristics:

* It eliminates the shortcomings of the MPLS based solutions, as (1) it works in case of failures between ingress-PE and egress-PE and (2) it supports direct localization of failures.

* It defines new functions only for Ingress PE nodes.

* It uses a VPN specifc SID as a source address on ingress PE nodes.

* It does not result in additional complexity on P nodes.

* It is compliant to existing standards on P nodes, like [RFC4443].

* It makes P nodes service agnostic and allows building IPv6-only core networks.

* It does not involve Egress PE nodes in the forwarding of the ICMP error messages.

* It can hide the SIDs used inside the SRv6 domain and can provide different visibility for served VPNs if needed.

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 makes no IANA requests.

6. Acknowledgements

Authors extend their appreciation to Janos Farkas, Ferenc Fejes, Xiao Min, Liu Yao, Greg Mirsky, and Krzysztof Szarkowicz for their insightful comments and productive discussion that helped to improve the document.

7. Normative References

[I-D.ietf-intarea-extended-icmp-nodeid]
Fenner, B. and R. Thomas, "Adding Extensions to ICMP Errors for Originating Node Identification", Work in Progress, Internet-Draft, draft-ietf-intarea-extended-icmp-nodeid-04, 19 August 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-intarea-extended-icmp-nodeid-04>.

[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>.

[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>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC7600] Despres, R., Jiang, S., Ed., Penno, R., Lee, Y., Chen, G., and M. Chen, "IPv4 Residual Deployment via IPv6 - A Stateless Solution (4rd)", RFC 7600, DOI 10.17487/RFC7600, July 2015, <https://www.rfc-editor.org/info/rfc7600>.

[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>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[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>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/info/rfc9256>.

Authors' Addresses

Balazs Varga
Ericsson
Email: balazs.a.varga@ericsson.com

Joel Halpern
JMH
Email: jmh@joelhalpern.com