Workgroup: Network Working Group
Internet-Draft:
draft-liu-6man-icmp-verification-00
Published: 22 September 2021
Intended Status: Standards Track
Expires: 26 March 2022
Authors: Y. Liu
ZTE

Extending ICMP for IP-related Information Validation

Abstract

This document introduces the mechanism to verify the data plane against the control plane in IP/SRv6 networks by extending ICMP messages.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

An MPLS label can be related with various FEC information, e.g, VPN IP prefix [RFC4365], LDP IP prefix[RFC5036], flex algorithms[I-D.ietf-lsr-flex-algo] and etc. Most of these information can be advertised via control plane protocols(e.g, IGP, BGP, etc).

Procedures for simple and efficient mechanisms to verify the data plane against the control plane using LSP Ping in MPLS network are well defined in [RFC8029]. Normally, when a new feature is introduced and the MPLS label is associated with new information, the LSP Ping mechanism is still applicable by defining new FEC sub-TLV with the new information encoded in it.

On the other hand, IP addresses, especially the IPv6 address/SRv6 SID, can be related with extra information/function besides basic forwarding/routing semantics.

Below is a non-exhaustive list of the information that can be related with IP addresses/SRv6 SIDs and propagated to the control plane.

*VPN/EVPN Services [I-D.ietf-bess-srv6-services]

*SRv6 Endpoint Behaviors for Network Programming [RFC8986]

- Flex Algorithms [I-D.ietf-lsr-flex-algo] [I-D.ietf-lsr-ip-flexalgo]

- Service Functions [I-D.ietf-spring-sr-service-programming]

- End-to-end Intent of a Path [I-D.hegde-spring-mpls-seamless-sr] [I-D.dskc-bess-bgp-car-problem-statement]

In IP networks, there're requirements to check the consistency between the control plane and the data plane to localize faults.

Take IPv4 VPN as an example, in MPLS, an MPLS label is allocated for the VPN prefix, the label is advertised together with the VPN prefix via BGP [RFC4365]. To verify this information, VPN IPv4 Prefix FEC sub-TLV is defined which carries the VPN prefix to be verified via LSP ping[RFC8029]. Similarly, in SRv6, an SRv6 SID is associated with a VPN prefix, and they are advertised together via BGP[I-D.ietf-bess-srv6-services]. One may want to verify the SID-related VPN prefix just like what is done in MPLS-VPN.

This document introduces the mechanism to verify the data plane against the control plane in IP networks by extending ICMP messages. Currently this document focuses on the extensions of ICMPv6 and the related processing procedures, considering that the requirements are stronger for IPv6/SRv6 networks.

1.1. 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].

2. ICMPv6 Validation Request

The Validation Request message is defined for ICMPv6[RFC4443]. Like any ICMPv6 message, the ICMPv6 Validation Request message is encapsulated in an IPv6 header.

The structure of ICMP Validation Request is shown in Figure 1, where:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |     Code      |          Checksum             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Identifier          |Sequence Number|   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   ICMP Extension Structure
```

Figure 1: Validation Request

*Type: The value is TBD1.

*Code: MUST be set to 0 and MUST be ignored upon receipt.

*Checksum: For ICMPv6, see [RFC4443].

*Identifier: An Identifier to aid in matching Validation Replies to Validation Requests. May be zero.

*Sequence Number: A Sequence Number to aid in matching Validation Replies to Validation Requests. May be zero.

*Reserved: This field MUST be set to 0 and ignored upon receipt.

*ICMP Extension Structure: The ICMP Extension Structure carries the information that needs to be verified. Section 7 of [RFC4884] defines the ICMP Extension Structure. As per [RFC4884], the Extension Structure contains one Extension Header followed by one or more objects. When applied to the ICMP Validation Request message, the ICMP Extension Structure MUST only contain one or more instance of the Validation Information Objects as defined in section 2.1.

2.1. Validation Information Object

The Validation Information Object is shown in Figure 2, where:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Length                |   Class-Num   |   C-Type      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                   // (Object payload) //                      |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Validation Information Object

- *Length: Length of the object, measured in octets, including the Object Header and Object Payload.

- *Class-Num: Validation Information Object. The value is TBD2.

- *Object payload: Variable-length field. C-Type-specific data.

- *C-Type: For this object, the C-Type is used to indicate the type of the information that needs to be verified. The values of C-Type and the corresponding object payload are given below:

C-Type	Object Payload
1	Endpoint Behavior
2	IPv6 Prefix IGP Algorithm
3	SRv6 IGP-Adjacency Segment
4	VPN IPv4 Prefix
5	VPN IPv6 Prefix

Other C-Type values and the corresponding information carried in object payload will be defined as needed.

2.1.1. SRv6 Endpoint Behavior

When the endpoint behavior[RFC8986] of an SRv6 SID needs to be verified, the following format of object payload is used.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |       Endpoint Behavior       |          Must Be Zero         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Endpoint Behavior: 2 octets. The codepoints for the Endpoint behaviors are defined in the "SRv6 Endpoint Behaviors" registry defined in [RFC8986].

2.1.2. IPv6 Prefix IGP Algorithm

IGP Flex-Algorithm can be used with both Segment Routing data planes(i.e, SR-MPLS and SRv6) [I-D.ietf-lsr-flex-algo] and for regular IPv4 and IPv6 prefixes [I-D.ietf-lsr-ip-flexalgo] .

When the algorithm of an SRv6 SID or IPv6 prefix needs to be verified, the following format of object payload is used.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Protocol   |   Algorithm   |          Reserved             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Protocol

Set to 1 if the responder MUST perform validation using OSPF as the IGP protocol. Set to 2 if the responder MUST perform validation using IS-IS as the IGP protocol. Set to 0 if the responder can use any IGP protocol for validation.

Algorithm

Set to 0 if the default algorithm is used. Set to 1 if Strict Shortest Path First (Strict-SPF) algorithm is used. For Flex-Algo, the Algorithm field MUST be set with the algorithm value (values can be 128-255).

SRv6 End SIDs inherit the algorithm from the parent locator.

Reserved

MUST be 0 when originated and MUST be ignored when received.

2.1.3. SRv6 IGP-Adjacency Segment

This object payload is applicable for SRv6 IGP-Adjacency defined in [RFC8402]. The format is as specified below:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Adj. Type   |    Protocol   |   Algorithm   |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                                                               ~
   |               Local Interface ID (4 or 16 octets)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                                                               ~
   |              Remote Interface ID (4 or 16 octets)             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                                                               ~
   |          Advertising Node Identifier (4 or 6 octets)          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   ~                                                               ~
   |           Receiving Node Identifier (4 or 6 octets)           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Adj. Type (Adjacency Type)

Set to 1 when the Adjacency Segment is a Parallel Adjacency as defined in [RFC8402]. Set to 4 when the Adjacency Segment is IPv4 based and is not a Parallel Adjacency. Set to 6 when the

Adjacency Segment is IPv6 based and is not a Parallel Adjacency. Set to 0 when the Adjacency Segment is over an unnumbered interface.

Protocol

Set to 1 if the responder MUST perform validation using OSPF as the IGP protocol. Set to 2 if the responder MUST perform validation using IS-IS as the IGP protocol. Set to 0 if the responder can use any IGP protocol for validation.

Algorithm

Set to 0 if the default algorithm is used. Set to 1 if Strict Shortest Path First (Strict-SPF) algorithm is used. For Flex-Algo, the Algorithm field MUST be set with the algorithm value (values can be 128-255).

The algorithm is specified in the individual SRv6 Adjacency SID.

Local Interface ID

An identifier that is assigned by the local node for a link to which the Adjacency Segment ID is bound. This field is set to a local link address (IPv4 or IPv6). For IPv4, this field is 4 octets; for IPv6, this field is 16 octets. If unnumbered, this field is 4 octets and includes a 32-bit link identifier as defined in [RFC4203] and [RFC5307]. If the Adjacency Segment ID represents Parallel Adjacencies, this field is 4 octets and MUST be set to 4 octets of zeroes.

Remote Interface ID

An identifier that is assigned by the remote node for a link on which the Adjacency Segment ID is bound. This field is set to the remote (downstream neighbor) link address (IPv4 or IPv6). For IPv4, this field is 4 octets; for IPv6, this field is 16 octets. If unnumbered, this field is 4 octets and includes a 32-bit link identifier as defined in [RFC4203] and [RFC5307]. If the Adjacency Segment ID represents Parallel Adjacencies, this field is 4 octets and MUST be set to 4 octets of zeroes.

Advertising Node Identifier

This specifies the Advertising Node Identifier. When the Protocol field is set to 1, then this field is 4 octets and carries the 32-bit OSPF Router ID. If the Protocol field is set to 2, then this field is 6 octets and carries the 48-bit IS-IS System ID. If the Protocol field is set to 0, then this field is 4 octets and MUST be set to zero.

Receiving Node Identifier

This specifies the downstream node identifier. When the Protocol field is set to 1, then this field is 4 octets and carries the

32-bit OSPF Router ID. If the Protocol field is set to 2, then this field is 6 octets and carries the 48-bit IS-IS System ID. If the Protocol field is set to 0, then this field is 4 octets and MUST be set to zero.

2.1.4. VPN IPv4 Prefix

IPv4 VPN Over SRv6 Core is introduced in [I-D.ietf-bess-srv6-services], where an SRv6 service SID is associated with a VPN IPv4 prefix at the egress PE.

When the related VPN IPv4 prefix of an SRv6 service SID needs to be verified, the following format of object payload is used. The Value field consists of the RD advertised with the VPN IPv4 prefix, the IPv4 prefix (with trailing 0 bits to make 32 bits in all), and a prefix length, as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Route Distinguisher                      |
   |                          (8 octets)                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         IPv4 prefix                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Prefix Length |                 Must Be Zero                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The RD is an 8-octet identifier, it does not contain any inherent information. The purpose of the RD is solely to allow one to create distinct routes to a common IPv4 address prefix. The encoding of the RD is not important here. When matching this field to the local information, it is treated as an opaque value.

2.1.5. VPN IPv6 Prefix

IPv6 VPN Over SRv6 Core is introduced in [I-D.ietf-bess-srv6-services], where an SRv6 service SID is associated with a VPN IPv6 prefix at the egress PE.

When the related VPN IPv6 prefix of an SRv6 service SID needs to be verified, the following format of object payload is used.

The object payload field consists of the RD advertised with the VPN IPv6 prefix, the IPv6 prefix (with trailing 0 bits to make 128 bits in all), and a prefix length, as follows:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Route Distinguisher                      |
    |                          (8 octets)                           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                         IPv6 prefix                           |
    |                                                               |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Prefix Length |                 Must Be Zero                  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The RD is an 8-octet identifier, it does not contain any inherent information. The purpose of the RD is solely to allow one to create distinct routes to a common IPv4 address prefix. The encoding of the RD is not important here. When matching this field to the local information, it is treated as an opaque value.

3. ICMPv6 Validation Reply

The Validation Reply message is defined for ICMPv6. Like any ICMPv6 message, the ICMP Extended Echo Reply message is encapsulated in an IPv6 header. Figure 3 describes the ICMPv6 Validation Reply message.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Type      |     Code      |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           Identifier          |Sequence Number|   Reserved    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Validation Reply

ICMP fields:

*Type: Validation Reply. The value is TBD3.

*Code: Values are

(0) Validation passed

(1) Malformed request received

(2) One or more of the objects were not understood

(3) Information mismatch

- *Checksum: For ICMPv6, see [RFC4443].

- *Identifier: Copied from the Identifier field of the invoking Validation Request packet.

- *Sequence Number: Copied from the Sequence Number field of the invoking Validation Request packet.

4. ICMP Validation Message Processing

4.1. Sending a Validation Request

A node that originates an ICMP validation request message SHOULD first determine which IP address needs to be verified with what information. How the sender node get the information is out of scope of the document.

An ICMPv6 validation request contains one or more Validation Information objects, depending on how the user wants to do the validation. For example, an SRv6 service SID is related with an endpoint behavior and an IPv4 VPN prefix, if one wants to verify both information of the SID via one request message, an ICMPv6 validation request is sent with two validation information objects in it. Or one may choose to send two individual ICMPv6 validation requests, each carries one validation information object to verify these two information separately.

The target IP is the IP address/SRv6 SID to be verified and MUST be a unicast address. The ICMPv6 validation request is sent with the target IP address/SRv6 SID set as the destination address of the IP header field without SRH, or set as the last segment with SRH. The Source Address of the ICMPv6 packet MUST be a unicast address belonging to the node.

The Hop Limit SHOULD be set to 255 to prevent transit nodes from processing the validation request.

4.2. Receiving a Validation Request

All transit nodes process the validation request message like any other IPv6 data packet and hence do not require any change.

As specified in [RFC4443], if a router receives a packet with a Hop Limit of zero, or if a router decrements a packet's Hop Limit to zero, it MUST discard the packet and originate an ICMPv6 Time Exceeded message with Code 0 to the source of the packet. The source address SHOULD be set as a local address of the router.

The target node is a node receiving an validation request where the target IP of that message is locally configured as a segment or local interface.

When the validation request packet arrives at the target node, and any of the following conditions apply, the node MUST silently discard the incoming message:

- The node does not recognize ICMP Validation Request messages.
- The node has not explicitly enabled ICMP Validation functionality.
- The incoming ICMP Validation Request carries a Source Address that is not explicitly authorized for the incoming ICMP Validation Request type.
- The Source Address of the incoming message is not a unicast address.
- The Destination Address of the incoming message is not a unicast address.

Otherwise, if the packet is well formed, the target node verifies the information encoded in the Validation Information Object against the corresponding local information.

4.3. Sending a Validation Reply

When a node receives an ICMPv6 Validation Request, it MUST format an ICMPv6 Validation Reply as follows:

- Copy the Source Address from the Validation Request message to the Destination Address of the Validation Reply.
- Copy the Destination Address from the Validation Request message to the Source Address of the Validation Reply.
- Set the Hop Limit to 255
- Set the Next Header to ICMPv6.
- Set the DiffServ codepoint to CS0 [RFC4594].
- Set the ICMP Type to Validation Reply.
- Copy the Identifier from the Validation Request message to the Validation Reply.

- Copy the Sequence Number from the Validation Request message to the Validation Reply.

- Set the Code field as described in Section 4.3.1

- Set the Checksum appropriately.

- Forward the ICMP Validation Reply to its destination.

4.3.1. Return Code

The Code field MUST be set to 0 if all the the information encoded in the Validation Information Object is consistent with the the corresponding local information on the target node.

The Code field MUST be set to 1 if any of the following conditions apply:

- The ICMP Request does not include an ICMP Extension Structure.

- The ICMP Extension Structure does not only include the Validation Information Object(s).

- The query is otherwise malformed.

The Code field MUST be set to 2 if one or more of the objects are not understood by the node.

The Code field MUST be set to 3 if the information in the Validation Information Object(s) is not consistent with the local information and validation is not passed.

4.4. Receiving a Validation Reply

A node should only receive a validation reply in response to a validation request that it sent. Thus, on receipt of a validation reply, the node should parse the packet to ensure that it is well-formed, then attempt to match up the validation reply with a validation request that it had previously sent, using the Identifier and Sequence Number. If no match is found, the node ignores the echo reply.

5. IANA Considerations

This document requests the following actions from IANA:

- Add an entry to the "ICMPv6 "type" Numbers" registry, representing the Validation Request. This entry has one code 0.

- Add an entry to the "ICMPv6 "type" Numbers" registry, representing the Validation Reply. This entry has the following codes:

  (0) Validation passed

  (1) Malformed request received

  (2) One or more of the objects were not understood

  (3) Information mismatch

- Add an entry to the "Validation Information Object Classes and Class Sub-types" registry, representing the Validation Information Object with C-types:

  (1) Endpoint Behavior

  (2) IPv6 Prefix IGP Algorithm

  (3) SRv6 IGP-Adjacency Segment

  (4) VPN IPv4 Prefix

  (5) VPN IPv6 Prefix

  C-Type values are assignable on a first-come-first-serve (FCFS) basis with a range of 0-255.

All codes mentioned above are assigned on a First Come First Serve (FCFS) basis with a range of 0-255.

6. Security Considerations

Security considerations discussed in [RFC4443] and[RFC4884] apply to this document.

To protect against unauthorized sources using validation request messages to obtain network information, it is RECOMMENDED that implementations provide a means of checking the source addresses of validation request messages against an access list before accepting the message.

The validation mechanism SHOULD be only used in the limited domain. The validation request contains the control plane information, policies should be implemented on the edge devices of the domain to prevent the information from being leaked into other domains.

In order to protect local resources, implementations SHOULD rate-limit incoming ICMP Request messages.

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>.

[RFC4203] Kompella, K., Ed. and Y. Rekhter, Ed., "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4203, DOI 10.17487/RFC4203, October 2005, <https://www.rfc-editor.org/info/rfc4203>.

[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>.

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <https://www.rfc-editor.org/info/rfc4594>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <https://www.rfc-editor.org/info/rfc4884>.

[RFC5307] Kompella, K., Ed. and Y. Rekhter, Ed., "IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 5307, DOI 10.17487/RFC5307, October 2008, <https://www.rfc-editor.org/info/rfc5307>.

[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>.

[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>.

7.2. Informative References

[I-D.dskc-bess-bgp-car-problem-statement]

Rao, D., Agrawal, S., Filsfils, C., Talaulikar, K., Decraene, B., Steinberg, D., Jalil, L., Guichard, J., Patel, K., and W. Henderickx, "BGP Color-Aware Routing Problem Statement", Work in Progress, Internet-Draft, draft-dskc-bess-bgp-car-problem-statement-03, 23 May 2021, <https://datatracker.ietf.org/doc/html/draft-dskc-bess-bgp-car-problem-statement-03>.

[I-D.hegde-spring-mpls-seamless-sr]
Hegde, S., Bowers, C., Xu, X., Gulko, A., Bogdanov, A., Uttaro, J., Jalil, L., Khaddam, M., Alston, A., and L. M. Contreras, "Seamless SR Problem Statement", Work in Progress, Internet-Draft, draft-hegde-spring-mpls-seamless-sr-05, 22 February 2021, <https://datatracker.ietf.org/doc/html/draft-hegde-spring-mpls-seamless-sr-05>.

[I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay Services", Work in Progress, Internet-Draft, draft-ietf-bess-srv6-services-07, 11 April 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-bess-srv6-services-07>.

[I-D.ietf-lsr-flex-algo] Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", Work in Progress, Internet-Draft, draft-ietf-lsr-flex-algo-17, 6 July 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-lsr-flex-algo-17>.

[I-D.ietf-lsr-ip-flexalgo] Britto, W., Hegde, S., Kaneriya, P., Shetty, R., Bonica, R., and P. Psenak, "IGP Flexible Algorithms (Flex-Algorithm) In IP Networks", Work in Progress, Internet-Draft, draft-ietf-lsr-ip-flexalgo-03, 14 May 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-lsr-ip-flexalgo-03>.

[I-D.ietf-spring-sr-service-programming]
Clad, F., Xu, X., Filsfils, C., Bernier, D., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", Work in Progress, Internet-Draft, draft-ietf-spring-sr-service-programming-05, 10 September 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-05>.

[RFC4365] Rosen, E., "Applicability Statement for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4365, DOI 10.17487/

RFC4365, February 2006, <https://www.rfc-editor.org/info/rfc4365>.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <https://www.rfc-editor.org/info/rfc5036>.

[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>.

Author's Address

Yao Liu
ZTE
Nanjing
China

Email: liu.yao71@zte.com.cn