Authors: H. Song, Ed.   Z. Li   T. Zhou
         Futurewei Technologies   Huawei   Huawei
         L. Andersson   Z. Zhang
         Bronze Dragon Consulting   Juniper Networks
         R. Gandhi   J. Rajamanickam   J. Bhattacharya
         Cisco Systems   Cisco Systems   Cisco Systems

Support MPLS Network Actions using Post-Stack Extension Headers

Abstract

Motivated by the need to support multiple in-network services and
functions in an MPLS network (a.k.a. MPLS Network Actions (MNA)),
this document describes a generic and extensible method to
encapsulate MNA instructions as well as possible ancillary data in
an MPLS packet. All the post-stack MNAs are encapsulated in a
structure called Post-stack MNA Header (PAH). A PAH is composed of a
common header plus a chain of extension headers; each extension
header is a container for an MNA. The encapsulation method allows
chaining multiple post-stack extension headers and provides the
means to enable fast access to them as well as the original upper
layer headers. This document confines to the solution of PAH
encoding and leaves the specification of PAH indicator to the
overall MNA solution. We show how PAH can be used to support several
new MNAs as a generic post-stack mechanism.

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 5 March 2023.

Copyright Notice

Table of Contents

1. Motivation

Some applications require adding sizable action instructions and/or
ancillary data to packets within an MPLS network. Such examples
include In-situ OAM (IOAM) [RFC9197] and Service Function Chaining
(SFC) [RFC7665]. New applications are emerging. It is possible that
the instructions and/or ancillary data for multiple MNAs are stacked
together in one packet to support a compound service.

Such instructions and/or ancillary data would need to be encoded and
encapsulated as new headers in packets. Such headers may require to
be processed in fast path due to performance considerations.

Moreover, such headers may require being attended at each hop on the forwarding path (i.e., hop-by-hop or HBH) or at designated end nodes (i.e., end-to-end or E2E).

The need and requirements to support such applications in MPLS networks, i.e., MPLS Network Actions (MNA), are described in [I-D.ietf-mpls-mna-usecases] and [I-D.ietf-mpls-mna-requirements]. It is clear that some header should be located after the MPLS label stack. We call such a header Post-stack MNA Header (PAH). The encapsulation of PAH poses some challenges to MPLS networks, because the MPLS label stack contains no explicit indicator for the upper layer protocols by design.

The mechanism to indicate the presence of the PAH is outside the scope of this document. The indication for the presence of the PAH can be achieved using several mechanisms, including carrying a Special Purpose Label (SPL) or signaling it with the label Forwarding Equivalence Class (FEC) as described in [I-D.ietf-mpls-mna-fwk]. In this document, we focus on the encoding and encapsulation of the PAH in an MPLS packet.

The conventional header encoding and encapsulation methods face some challenges in the case of post-stack MNA:

  *A solution may rely on either the built-in next-protocol
   indicator in the header or the knowledge of the format and size
   of the header to access the following packet headers. This method
   requires each node to be able to parse the new header, which is
   unrealistic in an incremental deployment environment.

  *Some works provide only piecemeal solutions which assume the new
   header is the only extra header and its location in the packet is
   fixed by default (e.g., Encapsulation of SFC NSH in MPLS
   [RFC8596]). It is impossible or difficult to support multiple new
   headers in one packet due to the conflicting location assumption.

  *Some previous work such as G-ACH [RFC5586] was explicitly defined
   for control channel only, but we need the mechanism to also work
   for user packets.

To solve the aforementioned problems, we introduce PAH as a general and extensible means to support new MNAs which involve instructions and/or ancillary data for each MNA. The concept is similar to IPv6 extension headers which offer a huge potential for extending IPv6's capability (e.g, network security, SRv6 [RFC8754], network programming [RFC8986], SFC [I-D.ietf-spring-sr-service-programming], etc.). Thanks to the mechanism of extension headers, it is straightforward to continue introducing new network services into IPv6 networks.

Nevertheless, when applying the extension headers to MPLS, some
issues of the IPv6 EH should be avoided:

   *IPv6's extension headers are chained with the original upper
    layer protocol headers in a flat stack. One must scan all the
    extension headers to access the upper layer protocol headers and
    the payload. This is inconvenient and raises some performance
    concerns for some applications (e.g., Deep Packet Inspection
    (DPI) and Equal Cost Multi Path (ECMP)). The new PAH scheme for
    MPLS needs to improve this.

   *[RFC8200] enforces many constraints to IPv6 extension headers
    (e.g., EH can only be added or deleted by the end nodes specified
    by the IP addresses in the IPv6 header, and there is only one
    Hop-by-Hop EH that can be processed on the path nodes), which are
    not suitable for MPLS networks. For example, MPLS label stacks
    are added and changed in network, and there could be tunnel
    within tunnel, so the extension headers need more flexibility.

2. MPLS Post-stack Network Action Header

The concept and design of the PAH comply with the requirements laid
out in [I-D.ietf-mpls-mna-requirements]. All the post-stack MNAs are
encapsulated in a PAH. A PAH is composed of a common header plus a
chain of extension headers; each extension header is a container for
an MNA. Here we highlight some design objectives of PAH (Note: these
should be covered by the MNA requirement document):

Performance: Unnecessary full extension header chain scanning for
   all MNAs or the upper layer headers should be avoided. The
   extension headers should be ordered according to the access need.
   Each extension header should serve only one MNA to avoid the need
   of packing multiple TLV options in one extension header.

Scalability: New MNAs can be supported by introducing new extension
   headers. Multiple extension headers can be easily stacked
   together to support multiple services simultaneously.

Backward Compatibility: Legacy devices which do not recognize the
   PAH should still be able to forward the packets based on the top
   label as usual. If a PAH-aware device recognizes some of the MNAs
   but not the others in an extension header chain, it can process
   the known MNAs only while ignoring the others.

Flexibility: A node (i.e., an LER or LSR) can be configured to
   process or not process any EH. Any tunnel end nodes in the MPLS
   domain can add new EH to the packets which shall be removed on
   the other end of the tunnel.

We assume the MPLS label stack has included some indicator of the
PAH. The actual PAH is inserted between the MPLS label stack and the
original upper layer header. The format of the MPLS packets with PAH
is shown in Figure 1.

```
0                                31
+--------+--------+--------+--------+
|                                   |
~            MPLS Label Stack       ~
|                                   |
+--------+--------+--------+--------+
|                BoS Label          |
+--------+--------+--------+--------+
|     Common Header of PAH (CH)     | \
+--------+--------+--------+--------+  |
|                                   | |
~        Extension Header (EH) 1    ~ |
|                                   | |
+--------+--------+--------+--------+  >MPLS PAH
~               ... ...             ~ |
+--------+--------+--------+--------+  |
|                                   | |
~        Extension Header (EH) n    ~ |
|                                   | /
+--------+--------+--------+--------+
|                Original           |
~       Upper Layer Headers/Payload ~
|                                   |
+--------+--------+--------+--------+
```
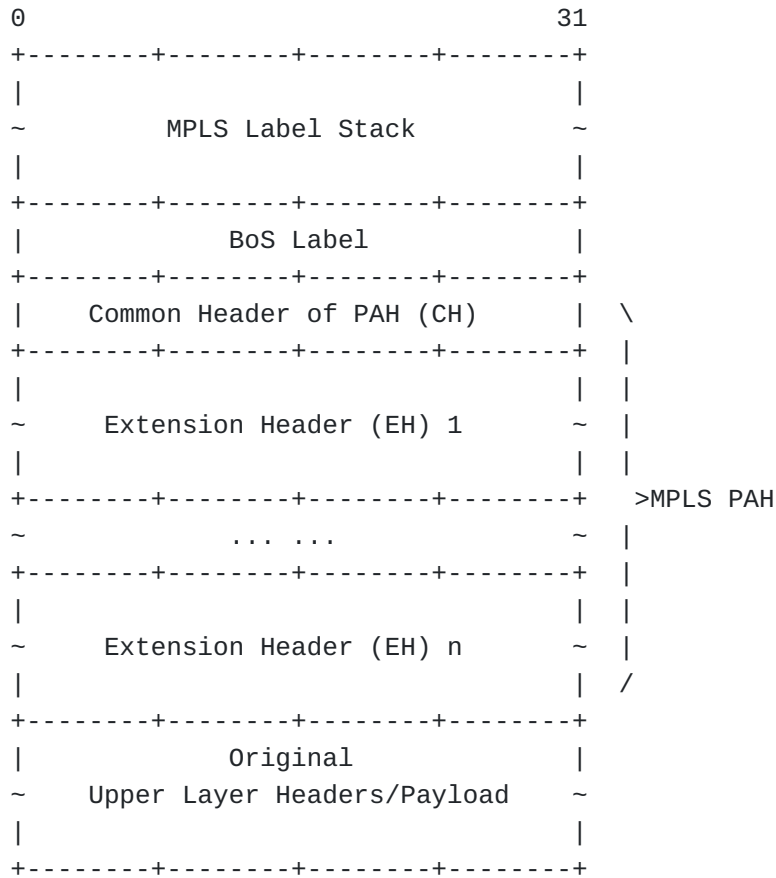
           Figure 1: MPLS with Post Stack Network Action Header

   Following the MPLS label stack is the 4-octet Common Header of PAH
   (CH), which indicates the total number of extension headers in this
   packet, the overall length of the PAH, the type of the original
   upper layer header, and the type of the next extension header. The
   format of the CH is shown in Figure 2.

```
 0            1          2          3
 0123 4567 89012345 67890123 45678901
 +----+----+--------+--------+--------+
 | R  |EHC |  EHTL  |  OUL   |   NH   |
 +----+----+--------+--------+--------+
```
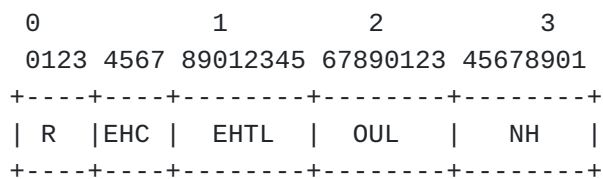
Figure 2: CH Format

The meaning of the fields in a CH is as follows:

R: reserved nibble. The nibble value means to avoid any potential
   conflicting with IP version numbers and other well-defined
   semantics [I-D.kbbma-mpls-1stnibble].

EHC: 4-bit unsigned integer for the Extension Header Counter. This
   field keeps the total number of extension headers included in
   this packet. It does not count the original upper layer headers.
   At most 15 EHs are allowed in one packet.

EHTL: 8-bit unsigned integer for the Extension Header Total Length
   in 4-octet units. This field keeps the total length of the EHs in
   this packet, not including the CH itself.

OUL: 8-bit Original Upper Layer protocol number indicating the
   original upper layer protocol type. It can be set to "UNKNOWN"
   (value TBD) if unknown. Sometimes the MPLS FEC may indicate the
   type of payload. In this case either OUL is redundant or OUL can
   be used to replace the control plane mechanism.

NH: 8-bit indicator for the Next Header. This field identifies the
   type of the MNA in the extension header immediately following the
   CH.

The value of the reserved nibble needs further consideration. The
EHC field can be used to keep track of the number of extension
headers when some headers are inserted or removed at some network
nodes. The EHTL field can help to skip all the EHs in one step if
the original upper layer headers or payload need to be accessed. The
OUL field can help identify the type of the original upper layer
protocol.

The format of an Extension Header (EH) is shown in Figure 3.

```
 0           1           2           3
 01234567 89012345 67890123 45678901
 +--------+--------+--------+-------+
 |  NH    | HLEN   |   EXT(opt)    |
 +--------+--------+--------+-------+
 |                                 |
 ~  MNA Instruction/Ancillary Data  ~
 |                                 |
 +--------+--------+----------------+
```
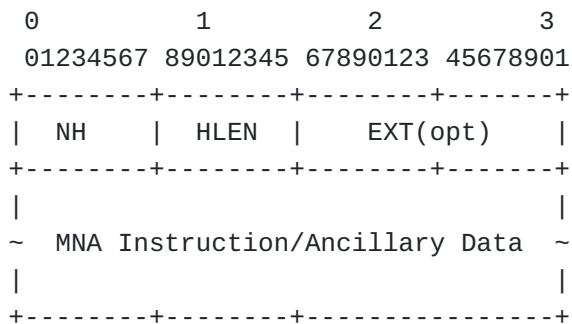
Figure 3: EH Format

The meaning of the fields in an EH is as follows:

NH: 8-bit indicator for the Next Header type. This field identifies
the type of the MNA in the EH immediately following this EH.

HLEN: 8-bit unsigned integer for the Extension Header Length in 4-
octet units, not including the first 4 octets.

EXT: 16-bit optional type extension. To save the Next Header
numbers and extend the number space, it is possible to use one
"Next Header" code to cover a set of sub-types. For example, IOAM
has several different options, such as trace and DEX. It is too
expensive to assign an EH type for each of it. In this case, it
is better to have a single EH type value for IOAM, and use the
EXT to specify the option types. This field is optional and only
specified for some specific MNA types. This field can also be
used to encode other information.

MNA Instruction/Ancillary Data: A variable length field for the
specification of an MNA. This field may need to be padded to make
the EH 4-octet aligned.

The extension headers as well as the first original upper layer
protocol header are chained together through the NH field in CH and
EHs. The encoding of NH can share the same value registry for IPv4/
IPv6 protocol numbers. Values for new MNA types (i.e., NH number)
shall be assigned by IANA from the same registry as for the ipv4 and
ipv6 protocol numbers (https://www.iana.org/assignments/protocol-
numbers/protocol-numbers.xhtml).

Specifically, the NH field of the last EH in a chain can have some
special values, which shall be assigned by IANA as well:

NONE (No Next Header): Indicates that there is no other header and
payload after this EH. This can be used to transport packets with
only extension header(s), for example, the control packets for
control or the probe packets for measurements. Note that value 59
was reserved for "IPv6 No Next Header" indicator. It may be
possible for MPLS EH to share this value (Note: need to work with
6MAN).

UNKNOWN (Unknown Next Header): Indicates that the type of the
header after this header is unknown. This is intended to be
compatible with the original MPLS design in which the upper layer
protocol type is unknown from the MPLS header alone.

MPLS: Indicates that the next protocol header is still of MPLS type
and another MPLS label stack follows.

These NH values can only appear in the last EH in an PAH. Note that
the original upper layer protocol can be of type "MPLS", which
implies that a packet may contain multiple logically independent
label stacks separated by PAH. Having more than one independent
label stack is not new. For example, A Bier header could separate
the transport/bier labels and the payload labels; An MPLS Pseudo
Wire (PW) network could be implemented on the top of another
infrastructure MPLS network. In such cases, we have the flexibility
to apply different services to different label stacks.

3. Scope of MPLS Extension Headers

Basically, MPLS EHs have two application scopes based on the nature
of the contained MNA: HBH and E2E. E2E means that the EH is only
supposed to be inserted/removed and processed at the MPLS tunnel end
points where the MPLS header is inserted or removed. The EHs that
need to be processed on path nodes within the MPLS tunnel are of the
HBH type. However, any node in the tunnel can be configured to
ignore an HBH EH, even if it is capable of processing it.

If there are two types of EHs in a packet, the HBH EHs must take
precedence over the E2E EHs.

Making a distinction of the EH types and ordering the EHs in a
packet help improve the forwarding performance. For example, if a
node within an MPLS tunnel finds only E2E EHs in a packet, it can
avoid scanning the EH list.

The scope of an EH (i.e., HBH or E2E) is an intrinsic property of
the contained MNA. In other words, such information can be inferred
from the NH value.

4. Operation on MPLS PAH

A suitable indication for the presence of PAH is ensured before
adding the first EH X to an MPLS packet. Then the PAH is inserted
after the MPLS label stack. In the CH of the PAH, EHC is set to 1,
EHTL is set to the length of X in 4-octet units, OUL is set to a
proper value, and NH is set to the header type value of X. At last,
X is inserted after the CH, in which NH and HLEN are set
accordingly. Note that if this operation happens at a PE device, the
upper layer protocol is known before the MPLS encapsulation, so its
value can be saved in the OUL and NH field if desired. Otherwise,
the NH field is filled with the value of "UNKNOWN".

When an EH Y needs to be added to an MPLS packet which already
contains the PAH, the EHC and EHTL in the CH are updated accordingly
(i.e., EHC is incremented by 1 and EHTL is incremented by the size
of Y in 4-octet units). Then a proper location for Y in the EH chain
is located. Y is inserted at this location. The NH field of Y is

copied from the previous EH's NH field (or from the CH's NH field, if Y is the first EH in the chain). The previous EH's NH value, or, if Y is the first EH in the chain, the CH's NH value, is set to the NH value of Y.

Deleting an EH simply reverses the above operation. If the deleted EH is the last one, the PAH indicator and the PAH can also be removed.

When processing an MPLS packet with multiple extension headers in an PAH, the node needs to parse through the entire EH chain and process the EH one by one (but not necessarily in the parsing order). The node should ignore any EH that is not recognized or is configured as "Do not Processing" by the control plane.

The EH can be categorized into HBH or E2E. Since EHs are ordered based on their type (i.e., HBH EHs are located before E2E EHs), a node can avoid some unnecessary EH scan.

5. Use Cases

In this section, we show how PAH can be used to support several new network applications.

In-situ OAM: In-situ OAM (IOAM) records flow OAM information within user packets while the packets traverse a network. The instruction and collected data are kept in an IOAM header [RFC9197]. When applying IOAM in an MPLS network, the IOAM header can be encapsulated in an extension header within an PAH.

Network Telemetry and Measurement: A network telemetry and instruction header can be carried as an extension header in PAH to instruct a node what type of network measurements should be done. For example, the method described in [RFC8321] can be implemented in MPLS networks since the EH provides a natural way to color MPLS packets.

Network Security: Security related functions often require user packets to carry some instruction and ancillary data. In a DoS limiting network architecture, a "packet passport" header is used to embed packet authentication information for each node to verify.

Segment Routing and Network Programming: MPLS extension header in PAH can support the implementation of a new flavor of the MPLS-based segment routing, with better performance and richer functionalities. The details will be described in another draft.

With PAH, multiple in-network applications can be chained together as extension headers. For example, IOAM and SFC can be applied at

the same time to support network OAM and service function chaining.
A node can stop scanning the extension header chain if all the known
headers it can process have been located. For example, if IOAM is
the first EH in a chain and a node is configured to process IOAM
only, it can stop searching the EH chain when the IOAM EH is found.

Details on some of these use cases and discussions on some other use
cases are covered in [I-D.ietf-mpls-mna-usecases].

6. Security Considerations

The major security concerns may come from the MNAs that encapsulated
in the PAH. So we need to be careful to admit actions and take
measures to avoid the security threats such as information leak or
DoS attack.

7. IANA Considerations

This document requests IANA to assign two new Internet Protocol
Numbers from the "Protocol Numbers" Registry to indicate "No Next
Header" and "Unknown Next Header".

This document does not create any other new registries. New
registries for protocol numbers and type extension numbers should be
requested by each MNA use case document.

8. Contributors

The other contributors of this document are listed as follows.

   *James Guichard

   *Stewart Bryant

   *Andrew Malis

9. Acknowledgments

We thank Tarek Saad and the other members of MPLS ODT for helping
improve this document.

10. References

10.1. 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>.

10.2. Informative References

[I-D.ietf-ippm-ioam-ipv6-options] Bhandari, S. and F. Brockners,
           "In-situ OAM IPv6 Options", Work in Progress, Internet-
           Draft, draft-ietf-ippm-ioam-ipv6-options-08, 16 June
           2022, <https://www.ietf.org/archive/id/draft-ietf-ippm-
           ioam-ipv6-options-08.txt>.

[I-D.ietf-mpls-mna-fwk] Andersson, L., Bryant, S., Bocci, M., and T.
           Li, "MPLS Network Actions Framework", Work in Progress,
           Internet-Draft, draft-ietf-mpls-mna-fwk-00, 26 July 2022,
           <https://www.ietf.org/archive/id/draft-ietf-mpls-mna-
           fwk-00.txt>.

[I-D.ietf-mpls-mna-requirements] Bocci, M., Bryant, S., and J.
           Drake, "Requirements for MPLS Network Action Indicators
           and MPLS Ancillary Data", Work in Progress, Internet-
           Draft, draft-ietf-mpls-mna-requirements-03, 19 August
           2022, <https://www.ietf.org/archive/id/draft-ietf-mpls-
           mna-requirements-03.txt>.

[I-D.ietf-mpls-mna-usecases] Saad, T., Makhijani, K., Song, H., and
           G. Mirsky, "Use Cases for MPLS Network Action Indicators
           and MPLS Ancillary Data", Work in Progress, Internet-
           Draft, draft-ietf-mpls-mna-usecases-00, 19 May 2022,
           <https://www.ietf.org/archive/id/draft-ietf-mpls-mna-
           usecases-00.txt>.

[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-06, 9 June 2022, <https://
           www.ietf.org/archive/id/draft-ietf-spring-sr-service-
           programming-06.txt>.

[I-D.kbbma-mpls-1stnibble] Kompella, K., Bryant, S., Bocci, M.,
           Mirsky, G., and L. O. (. Andersson, "IANA Registry for
           the First Nibble Following a Label Stack", Work in
           Progress, Internet-Draft, draft-kbbma-mpls-1stnibble-02,

10 July 2022, <https://www.ietf.org/archive/id/draft-kbbma-mpls-1stnibble-02.txt>.

[RFC5586]   Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed.,
            "MPLS Generic Associated Channel", RFC 5586, DOI
            10.17487/RFC5586, June 2009, <https://www.rfc-editor.org/
            info/rfc5586>.

[RFC7665]   Halpern, J., Ed. and C. Pignataro, Ed., "Service Function
            Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/
            RFC7665, October 2015, <https://www.rfc-editor.org/info/
            rfc7665>.

[RFC8200]   Deering, S. and R. Hinden, "Internet Protocol, Version 6
            (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/
            RFC8200, July 2017, <https://www.rfc-editor.org/info/
            rfc8200>.

[RFC8321]
            Fioccola, G., Ed., Capello, A., Cociglio, M.,
            Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T.
            Mizrahi, "Alternate-Marking Method for Passive and Hybrid
            Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321,
            January 2018, <https://www.rfc-editor.org/info/rfc8321>.

[RFC8596]   Malis, A., Bryant, S., Halpern, J., and W. Henderickx,
            "MPLS Transport Encapsulation for the Service Function
            Chaining (SFC) Network Service Header (NSH)", RFC 8596,
            DOI 10.17487/RFC8596, June 2019, <https://www.rfc-
            editor.org/info/rfc8596>.

[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>.

[RFC9197]   Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi,
            Ed., "Data Fields for In Situ Operations, Administration,
            and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197,
            May 2022, <https://www.rfc-editor.org/info/rfc9197>.

Authors' Addresses

Haoyu Song (editor)

Futurewei Technologies
Santa Clara,
United States of America

Email: haoyu.song@futurewei.com

Zhenbin Li
Huawei
Beijing
P.R. China

Email: lizhenbin@huawei.com

Tianran Zhou
Huawei
Beijing
P.R. China

Email: zhoutianran@huawei.com

Loa Andersson
Bronze Dragon Consulting
Stockholm
Sweden

Email: loa@pi.nu

Zhaohui Zhang
Juniper Networks
Boston,
United States of America

Email: zzhang@juniper.net

Rakesh Gandhi
Cisco Systems
Canada

Email: rgandhi@cisco.com

Jaganbabu Rajamanickam
Cisco Systems
Canada

Email: jrajaman@cisco.com

Jisu Bhattacharya
Cisco Systems

Email: jisu@cisco.com