

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 13, 2020

X. Geng
M. Chen
Huawei
Y. Zhu
China Telecom
March 12, 2020

DetNet SRv6 Data Plane Encapsulation
draft-geng-detnet-dp-sol-srv6-02

Abstract

This document specifies Deterministic Networking data plane operation for SRv6 encapsulated user data.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

Status of This Memo

This Internet-Draft 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 September 13, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology and Conventions	3
2.1.	Terminology	3
2.2.	Conventions	4
3.	SRv6 DetNet Data Plane Overview	4
3.1.	SRv6 DetNet Data Plane Layers	5
3.2.	SRv6 DetNet Data Plane Scenarios	5
4.	SRv6 DetNet Data Plane Solution Considerations	7
5.	SRv6 DetNet Data Plane Solution for Service Sub-layer	8
5.1.	TLV Based SRv6 Data Plane Solution	9
5.1.1.	Encapsulation	9
5.1.2.	Replication SID and Elimination for DetNet	11
5.2.	SID Based SRv6 Data Plane Solution	13
5.2.1.	Encapsulation	13
5.2.2.	Functions	14
5.3.	DetNet SID Based SRv6 Data Plane Solution	15
5.3.1.	Encapsulation	15
5.3.2.	Functions	15
6.	SRv6 DetNet Data Plane Solution for Transport Sub-layer	16
7.	IANA Considerations	16
8.	Security Considerations	16
9.	Acknowledgements	16
10.	Normative References	16
	Authors' Addresses	17

[1.](#) Introduction

Deterministic Networking (DetNet), as described in [\[I-D.ietf-detnet-architecture\]](#) provides a capability to carry specified data flows with extremely low data loss rates and bounded latency within a network domain. DetNet is enabled by a group of technologies, such as resource allocation, service protection and explicit routes.

Segment Routing(SR) leverages the source routing paradigm. An ingress node steers a packet through an ordered list of instructions, called "segments". SR can be applied over IPv6 data plane using the Segment Routing Extension Header (SRH, [\[I-D.ietf-6man-segment-routing-header\]](#)). A segment in segment routing terminology is not limited to a routing/forwarding function.

A segment can be associated to an arbitrary processing of the packet in the node identified by the segment. In other words, an SRv6 Segment can indicate functions that are executed locally in the node where they are defined. SRv6 network Programming [[I-D.filsfils-spring-srv6-network-programming](#)] describe the different segments and functions associated to them.

This document describes how to implement DetNet in an SRv6 enabled domain, including :

- o Source routing, which steers the DetNet flows through the network according to an explicit path with allocated resources;
- o Network programming, which applies instructions (functions) to packets in some special nodes (or even all the nodes) along the path in order to guarantee, e.g., service protection and congestion protection.

DetNet SRv6 encapsulation and new SRv6 functions ([[I-D.filsfils-spring-srv6-network-programming](#)]) for DetNet are defined in this document. Control plane and OAM are not in the scope of this document.

Control plane and OAM are not in the scope of this document.

2. Terminology and Conventions

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](#)].

2.1. Terminology

Terminologies for DetNet go along with the definition in [[I-D.ietf-detnet-architecture](#)] and [[RFC8402](#)]. Other terminologies are defined as follows:

- o NH: The IPv6 next-header field.
- o SID: A Segment Identifier ([[RFC8402](#)]).
- o SRH: The Segment Routing Header ([[I-D.ietf-6man-segment-routing-header](#)]).

2.2. Conventions

Conventions in the document are defined as follows:

- o NH=SRH means that NH is 43 with routing type 4 which is (as defined in [[I-D.ietf-6man-segment-routing-header](#)], the values representing the SRH.
- o A SID list is represented as <S1, S2, S3> where S1 is the first SID to visit, S2 is the second SID to visit and S3 is the last SID to visit along the SR path.
- o SRH[SL] represents the SID pointed by the SL field in the first SRH. In our example, SRH[2] represents S1, SRH[1] represents S2 and SRH[0] represents S3. It has to be noted that [[I-D.ietf-6man-segment-routing-header](#)] defines the segment list encoding in the reverse order of the path. A path represented by <S1,S2,S3>, will be encoded in the SRH as follows:

SegmentList[0]=S3

SegmentList[1]=S2

SegmentList[2]=S1

The reverse encoding has been defined in order to optimise the processing time of the segment list. See [[draft-ietf-6man-segment-routing-header](#)] for more details.

- o (SA,DA) (S3, S2, S1; SL) represents an IPv6 packet with:

IPv6 header with source and destination addresses SA and DA respectively, and next-header set to SRH (i.e.: 43 with type 4) , with a list of segments(SIDs) <S1, S2, S3> with SegmentsLeft = SL

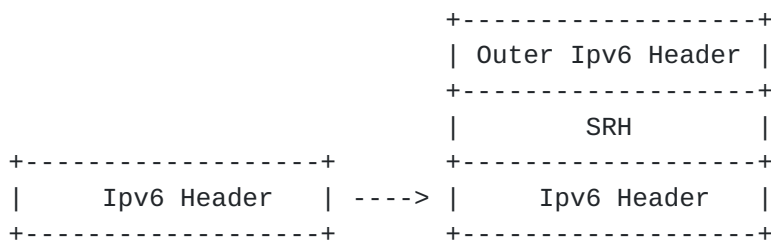
The payload of the packet is not represented

(S3, S2, S1; SL) represents the same SID list as <S1, S2, S3>, but encoded in the SRH format where the rightmost SID in the SRH is the first SID and the leftmost SID in the SRH is the last SID

3. SRv6 DetNet Data Plane Overview

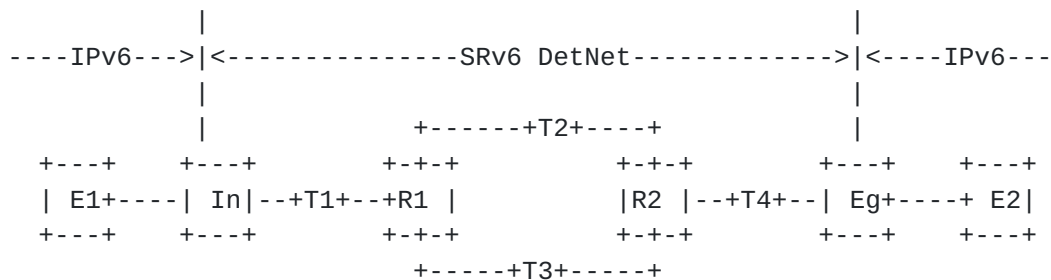
3.1. SRv6 DetNet Data Plane Layers

[[I-D.ietf-detnet-architecture](#)] decomposes the DetNet data plane into two sub-layers: service sub-layer and transport sub-layer. Different from DetNet MPLS data plane solution, which uses DetNet Control Word(d-CW) and S-Label to support service sub-layer and uses T-Label to support transport sub-layer, no explicit sub-layer division exists in SRv6 data plane. A classical SRv6 DetNet data plane solution is showed in the picture below:



The outer IPv6 Header with the SRH is used for carrying DetNet flows. Traffic Engineering is instantiated in the segment list of SRH, and other functions and arguments for service protection (packet replication, elimination and ordering) and congestion control (packet queuing and forwarding) are also defined in the SRH.

3.2. SRv6 DetNet Data Plane Scenarios



The figure above shows that an IPv6 flow is sent out from the end station E1. The packet of the flow is encapsulated in an outer IPv6+SRH header as a DetNet SRv6 packet in the Ingress(In) and transported through an SRv6 DetNet domain. In the Egress(Eg), the outer IPv6 header+SRH of the packet is popped, and the packet is sent to the destination E2.

The figure above shows that an IPv6 flow is sent out from the end station: E1. The packet of the flow is encapsulated as a DetNet SRv6 packet in the Ingress(In) and transported through an SRv6 DetNet domain. In the Egress(Eg), the upper IPv6 header with SRH of the packet is popped, and the packet is transmitted to the destination(E2).

The DetNet packet processing is as follows:

Ingress:

Inserts the SRv6 Policy that will steer the packet from Ingress to the destination

The methods and mechanisms used for defining, instantiating and applying the policy are outside of this document. An example of policies are described in [[I-D.ietf-spring-segment-routing-policy](#)]

Flow Identification and Sequence Number are carried in the SRH optionally.

Relay Node 1(Replication Node):

Replicates the payload and IPv6 Header with the SRH. This is a new function in the context of SRv6 Network Programming which will associate a given SID to a replication instruction in the node originating and advertising the SID. The replication instruction includes:

- * The removal of the existing IPv6+SRH header
- * The encapsulation into a new outer IPv6+SRH header. Each packet (the original and the duplicated) are encapsulated into respectively new outer IPv6+SRH headers.

Binding two different SRv6 Policies respectively to the original packet and the replicated packet, which can steer the packets from Relay Node 1 to Relay Node 2 through two tunnels.

If Flow Identification and Sequence Number are not carried in the SRH in the ingress, add it them in the SRH.

Relay Node 2(Elimination Node):

Eliminates the redundant packets.

Binds a new SRv6 Policy to the survival packet, which steers the packet from Relay Node 2 to Egress.

Egress:

Decapsulates the outer Ipv6 header.

Sends the inter packet to the End Station 2.

The DetNet packet encapsulation is illustrated here below. It has to be noted that, in the example below, the R2 address is a SRH SID associated to a TBD function related to the packet replication the node R1 has to perform. The same (or reverse) apply to node R2 which is in charge of the discard of the duplicated packet. Here also a new function will have a new SID allocated to it and representing the delete of the duplication in R2.

End Station1 output packet: (E1,E2)

Ingress output packet: (In, T1)(R1,T1, SL=2)(E1,E2)

Transit Node1 output packet: (In, R1)(R1,T1,SL=1)(E1,E2)

Relay Node1 output packets : (R1, T2)(R2, T2, SL=2)(E1, E2),
(R1, T3)(R2, T3, SL=2)(E1, E2)

Transit Node2 output packet: (R1, R2)(R2, T2, SL=1)(E1, E2)

Transit Node3 output packet: (R1, R2)(R2, T3, SL=1)(E1, E2)

Relay Node2 output packet: (R2, T4)(Eg, T4, SL=2)(E1, E2)

Transit Node4 output packet: (R2, Eg)(Eg, T4, SL=1)(E1, E2)

Egress out : (E1,E2)

4. SRv6 DetNet Data Plane Solution Considerations

To carry DetNet over SRv6, the following elements are required:

1. A method of identifying the SRv6 payload type;
2. A suitable explicit path to deliver the DetNet flow ;
3. A method of indicating packet processing, such as PREOF(Packet Replication, Elimination and Ordering as defined in [\[I-D.ietf-detnet-architecture\]](#));
4. A method of identifying the DetNet flow;
5. A method of carrying DetNet sequence number;
6. A method of carrying queuing and forwarding indication to do congestion protection;

In this design, DetNet flows are encapsulated in an outer IPv6+SRH header at the Ingress Node. The SR policy identified in the SRH steers the DetNet flow along a selected path. The explicit path followed by a DetNet flow, which protect it from temporary interruptions caused by the convergence of routing, is encoded within the SID list of the SR policy. The network device inside the DetNet domain forwards the packet according to IPv6 Destination Address(DA), and the IPv6 DA is updated with the SID List according to SRv6 forwarding procedures defined in [\[I-D.ietf-6man-segment-routing-header\]](#) and [\[I-D.filsfils-spring-srv6-network-programming\]](#)

With SRv6 network programming, the SID list can also give instruments representing a function to be called at the node in the DetNet domain. Therefore DetNet specific functions defined in [\[I-D.ietf-detnet-architecture\]](#), corresponding to local packet processing in the network, can also be implemented by SRv6. New functions associated with SIDs for DetNet are defined in this document.

This document describes how DetNet flows are encapsulated/identified, and how functions of Packet Replication/Elimination/Ordering are implemented in an SRv6 domain. Congestion protection is also in the scope of this document.

Editor: This version only covers the functions of service protection and the congestion protection considerations will be added in the following versions.

5. SRv6 DetNet Data Plane Solution for Service Sub-layer

This section defines options of SRv6 data plane solution to support DetNet Service Sub-layer.

SRH is as follows, which defined in [\[I-D.ietf-6man-segment-routing-header\]](#)

1. IF NH=SRH & SL>0 THEN
2. extract the DetNet TLV values from the SRH
3. create two new outer IPv6+SRH headers: IPv6-SRH-1 and IPv6-SRH-2
Insert the policy-instructed segment lists in each newly created SRH (SRH-1 and SRH-2). Also, add the extracted DetNet TLVs into SRH-1 and SRH-2.
4. remove the incoming outer IPv6+SRH header.
5. create a duplication of the incoming packet.
6. encapsulate the original packet into the first outer IPv6+SRH header: (IPv6-SRH-1) (original packet)
7. encapsulate the duplicate packet into the second outer IPv6+SRH header: (IPv6-SRH-2) (duplicate packet)
8. set the IPv6 SA as the local address of this node.
9. set the IPv6 DA of IPv6-SRH-1 to the first segment of the SRv6 Policy in of SRH-1 segment list.
10. set the IPv6 DA of IPv6-SRH-2 to the first segment of the SRv6 Policy in of SRH-2 segment list.
11. ELSE
12. drop the packet

5.1.2.2. Elimination SID

Elimination SID indicates the following operations:

- o Packet elimination: forward the first received packets and eliminate the redundant packets.
- o Packet ordering(optional): reorder the packets if the packet arrives out of order

When N receives a packet whose IPv6 DA is S and S is a Elimination SID, N could do:

1. IF NH=SRH & SL>0 & "the packet is not a redundant packet" THEN
2. do not decrement SL nor update the IPv6 DA with SRH[SL]

3. extract the value of DetNet TLVs from the SRH
4. create a new outer IPv6+SRH header
5. insert the policy-instructed segment lists in the newly created SRH and add the retrieved DetNet TLVs in the newly created SRH
6. remove the incoming outer IPv6+SRH header.
7. set the IPv6 DA to the first segment of the SRv6 Policy in the newly created SRH
8. ELSE
9. drop the packet

5.2. SID Based SRv6 Data Plane Solution

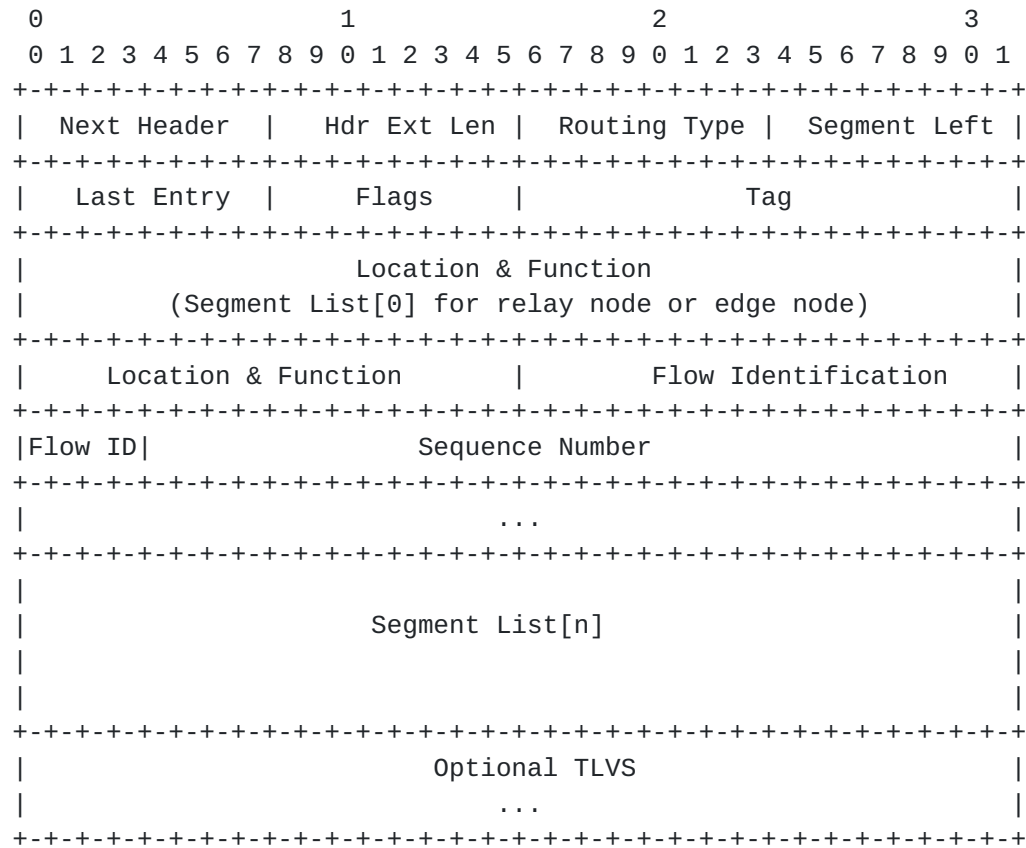
5.2.1. Encapsulation

SRv6 SID can be represented as LOC:FUNCT:ARG::, where:

LOC, means "LOCATION" and defines the node associated with the SID (i.e.: represented by the SID).

FUNCT, means "FUNCTION", and identifies the processing that the node specified in LOC applies to the packet.

ARG, means "ARGUMENTS" and provides the additional arguments for the function. New SID functions for DetNet is defined in [section 5.2.2](#). See [[I-D.filsfils-spring-srv6-network-programming](#)] for details on SRv6 Network Programming. The SRH for DetNet in the outer IPv6 header is illustrated as follows



where:

- o LOCATION&FUNCTION: the 80 most significant bits that are used for routing the packet towards the LOCATION (as defined in [\[I-D.filesfiles-spring-srv6-network-programming\]](#));
- o FLOW IDENTIFICATION: 20 bits, in the DetNet TLVs in the SRH, used for DetNet flow identification in the DetNet relay node;
- o SEQUENCE NUMBER : 28 bits, in the DetNet TLVs, used for dis crime packets in the same DetNet flow;

5.2.2. Functions

New SID functions are defined as follows:

5.2.2.1. End. B.Replication: Packet Replication Function

The function is similar as that has been defined in [section 5.1.2.1](#). The only difference is that instead of retrieving the TLV values, this function retrieves the argument.

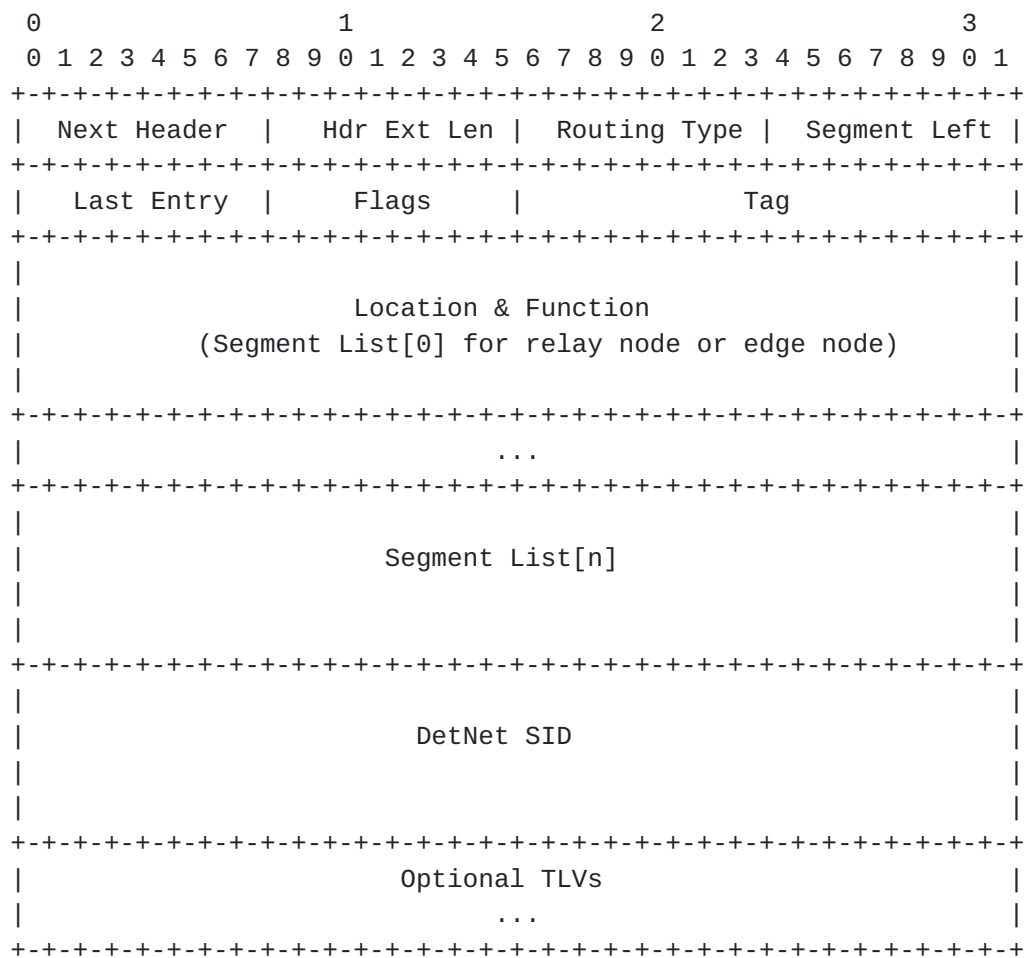
[5.2.2.2.](#) **End. B.** Elimination: Packet Elimination Function

The function is similar as that has been defined in [section 5.1.2.2.](#) The only difference is that instead of retrieving the TLV values, this function retrieves the argument.

[5.3.](#) **DetNet SID Based SRv6 Data Plane Solution**

[5.3.1.](#) **Encapsulation**

A non-forwarding DetNet SID is defined to carry Flow Identification and Sequence Number.



[5.3.2.](#) **Functions**

TBD

6. SRv6 DetNet Data Plane Solution for Transport Sub-layer

TBD

7. IANA Considerations

TBD

8. Security Considerations

TBD

9. Acknowledgements

Thank you for valuable comments from James Guichard and Andrew Mails.

10. Normative References

[I-D.filsfils-spring-srv6-network-programming]

Filsfils, C., Camarillo, P., Leddy, J.,
daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6
Network Programming", [draft-filsfils-spring-srv6-network-programming-07](#) (work in progress), February 2019.

[I-D.ietf-6man-segment-routing-header]

Filsfils, C., Dukes, D., Previdi, S., Leddy, J.,
Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header
(SRH)", [draft-ietf-6man-segment-routing-header-26](#) (work in
progress), October 2019.

[I-D.ietf-detnet-architecture]

Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", [draft-ietf-detnet-architecture-13](#) (work in progress), May 2019.

[I-D.ietf-detnet-dp-sol-mpls]

Korhonen, J. and B. Varga, "DetNet MPLS Data Plane
Encapsulation", [draft-ietf-detnet-dp-sol-mpls-02](#) (work in
progress), March 2019.

[I-D.ietf-spring-segment-routing-policy]

Filsfils, C., Sivabalan, S., Voyer, D., Bogdanov, A., and
P. Mattes, "Segment Routing Policy Architecture", [draft-ietf-spring-segment-routing-policy-06](#) (work in progress),
December 2019.

- [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>>.
- [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>>.

Authors' Addresses

Xuesong Geng
Huawei

Email: gengxuesong@huawei.com

Mach(Guoyi) Chen
Huawei

Email: mach.chen@huawei.com

Yongqing Zhu
China Telecom

Email: zhuyq@gsta.com

