Internet Engineering Task Force (IETF)
Request for Comments: 8754
Category: Standards Track
ISSN: 2070-1721

C. Filsfils, Ed.
D. Dukes, Ed.
Cisco Systems, Inc.
S. Previdi
Huawei
J. Leddy
Individual
S. Matsushima
SoftBank
D. Voyer
Bell Canada
March 2020

IPv6 Segment Routing Header (SRH)

Abstract

Segment Routing can be applied to the IPv6 data plane using a new type of Routing Extension Header called the Segment Routing Header (SRH). This document describes the SRH and how it is used by nodes that are Segment Routing (SR) capable.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8754.

Table of Contents

1. Introduction

Segment Routing (SR) can be applied to the IPv6 data plane using a

new type of routing header called the Segment Routing Header (SRH). This document describes the SRH and how it is used by nodes that are SR capable.

"Segment Routing Architecture" [RFC8402] describes Segment Routing and its instantiation in two data planes: MPLS and IPv6.

The encoding of IPv6 segments in the SRH is defined in this document.

1.1. Terminology

This document uses the terms Segment Routing (SR), SR domain, SR over IPv6 (SRv6), Segment Identifier (SID), SRv6 SID, Active Segment, and SR Policy as defined in [RFC8402].

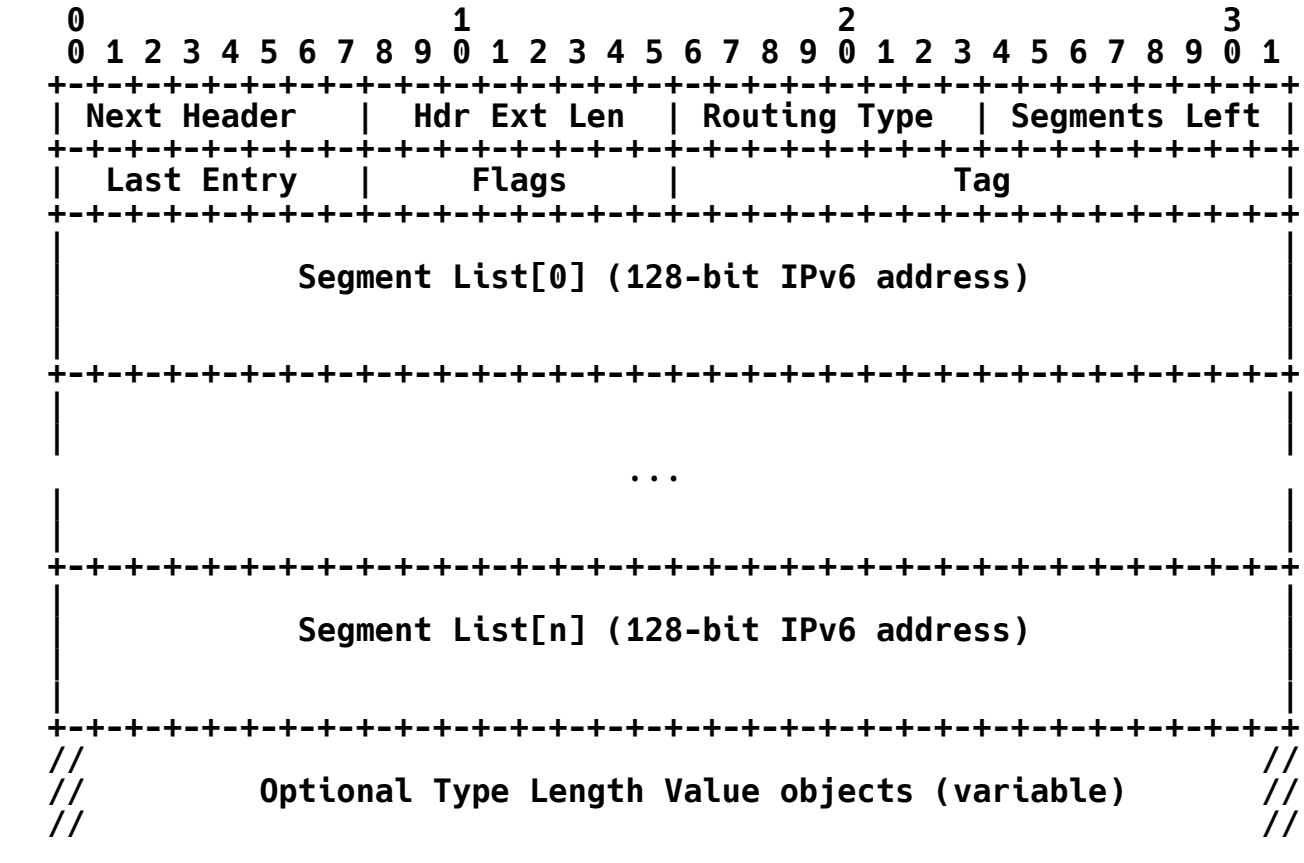
1.2. Requirements Language

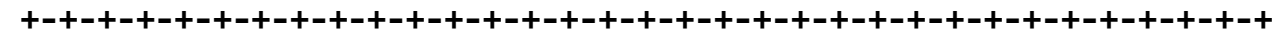
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Segment Routing Header

Routing headers are defined in [RFC8200]. The Segment Routing Header (SRH) has a new Routing Type (4).

The SRH is defined as follows:





where:

Next Header: Defined in [RFC8200], Section 4.4.

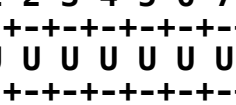
Hdr Ext Len: Defined in [RFC8200], Section 4.4.

Routing Type: 4.

Segments Left: Defined in [RFC8200], Section 4.4.

Last Entry: contains the index (zero based), in the Segment List, of the last element of the Segment List.

Flags: 8 bits of flags. Section 8.1 creates an IANA registry for new flags to be defined. The following flags are defined:



U: Unused and for future use. MUST be 0 on transmission and ignored on receipt.

Tag: Tag a packet as part of a class or group of packets -- e.g., packets sharing the same set of properties. When Tag is not used at the source, it MUST be set to zero on transmission. When Tag is not used during SRH processing, it SHOULD be ignored. Tag is not used when processing the SID defined in Section 4.3.1. It may be used when processing other SIDs that are not defined in this document. The allocation and use of tag is outside the scope of this document.

Segment List[0..n]: 128-bit IPv6 addresses representing the nth segment in the Segment List. The Segment List is encoded starting from the last segment of the SR Policy. That is, the first element of the Segment List (Segment List[0]) contains the last segment of the SR Policy, the second element contains the penultimate segment of the SR Policy, and so on.

TLV: Type Length Value (TLV) is described in Section 2.1.

In the SRH, the Next Header, Hdr Ext Len, Routing Type, and Segments Left fields are defined in Section 4.4 of [RFC8200]. Based on the constraints in that section, Next Header, Header Ext Len, and Routing Type are not mutable while Segments Left is mutable.

The mutability of the TLV value is defined by the most significant bit in the type, as specified in Section 2.1.

Section 4.3 defines the mutability of the remaining fields in the SRH (Flags, Tag, Segment List) in the context of the SID defined in this document.

New SIDs defined in the future MUST specify the mutability properties of the Flags, Tag, and Segment List and indicate how the Hashed Message Authentication Code (HMAC) TLV (Section 2.1.2) verification works. Note that, in effect, these fields are mutable.

Consistent with the SR model, the source of the SRH always knows how to set the Segment List, Flags, Tag, and TLVs of the SRH for use within the SR domain. How it achieves this is outside the scope of this document but may be based on topology, available SIDs and their mutability properties, the SRH mutability requirements of the destination, or any other information.

2.1. SRH TLVs

This section defines TLVs of the Segment Routing Header.

A TLV provides metadata for segment processing. The only TLVs defined in this document are the HMAC (Section 2.1.2) and padding TLVs (Section 2.1.1). While processing the SID defined in Section 4.3.1, all TLVs are ignored unless local configuration indicates otherwise (Section 4.3.1.1.1). Thus, TLV and HMAC support is optional for any implementation; however, an implementation adding or parsing TLVs MUST support PAD TLVs. Other documents may define additional TLVs and processing rules for them.

TLVs are present when the Hdr Ext Len is greater than (Last Entry+1)*2.

While processing TLVs at a segment endpoint, TLVs MUST be fully contained within the SRH as determined by the Hdr Ext Len. Detection of TLVs exceeding the boundary of the SRH Hdr Ext Len results in an ICMP Parameter Problem, Code 0, message to the Source Address, pointing to the Hdr Ext Len field of the SRH, and the packet being discarded.

An implementation MAY limit the number and/or length of TLVs it processes based on local configuration. It MAY limit:

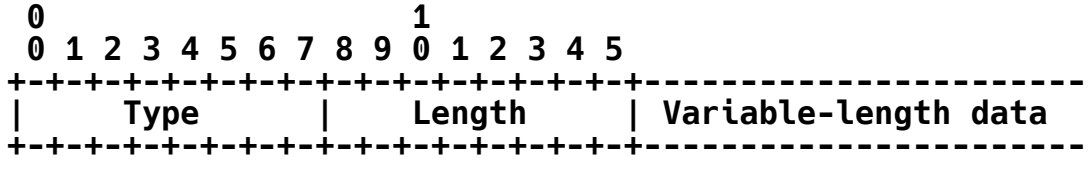
* the number of consecutive Pad1 (Section 2.1.1.1) options to 1. If padding of more than one byte is required, then PadN (Section 2.1.1.2) should be used.

* The length in PadN to 5.

* The maximum number of non-Pad TLVs to be processed.

* The maximum length of all TLVs to be processed.

The implementation MAY stop processing additional TLVs in the SRH when these configured limits are exceeded.



Type: An 8-bit codepoint from the "Segment Routing Header TLVs" [IANA-SRHTLV]. Unrecognized Types MUST be ignored on receipt.

Length: The length of the variable-length data field in bytes.

Variable-length data: data that is specific to the Type.

Type Length Value (TLV) entries contain OPTIONAL information that may be used by the node identified in the Destination Address (DA) of the packet.

Each TLV has its own length, format, and semantic. The codepoint allocated (by IANA) to each TLV Type defines both the format and the semantic of the information carried in the TLV. Multiple TLVs may be encoded in the same SRH.

The highest-order bit of the TLV type (bit 0) specifies whether or not the TLV data of that type can change en route to the packet's final destination:

0: TLV data does not change en route

1: TLV data does change en route

All TLVs specify their alignment requirements using an xn+y format. The xn+y format is defined as per [RFC8200]. The SR source nodes use the xn+y alignment requirements of TLVs and Padding TLVs when constructing an SRH.

The Length field of the TLV is used to skip the TLV while inspecting the SRH in case the node doesn't support or recognize the Type. The Length defines the TLV length in octets, not including the Type and Length fields.

The following TLVs are defined in this document:

Padding TLVs

HMAC TLV

Additional TLVs may be defined in the future.

2.1.1. Padding TLVs

There are two types of Padding TLVs, Pad1 and PadN, and the following applies to both:

Padding TLVs are used for meeting the alignment requirement of the subsequent TLVs.

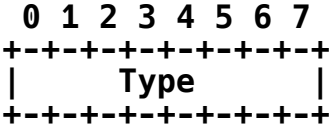
Padding TLVs are used to pad the SRH to a multiple of 8 octets.

Padding TLVs are ignored by a node processing the SRH TLV.

Multiple Padding TLVs MAY be used in one SRH.

2.1.1.1. Pad1

Alignment requirement: none

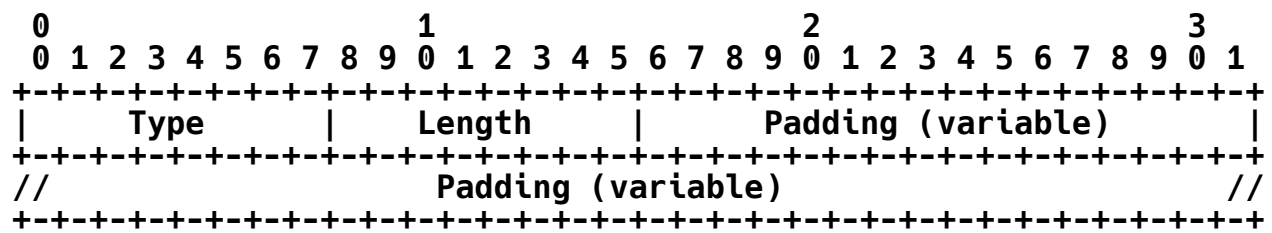


Type: 0

A single Pad1 TLV MUST be used when a single byte of padding is required. A Pad1 TLV MUST NOT be used if more than one consecutive byte of padding is required.

2.1.1.2. PadN

Alignment requirement: none



Type: 4

Length: 0 to 5. The length of the Padding field in bytes.

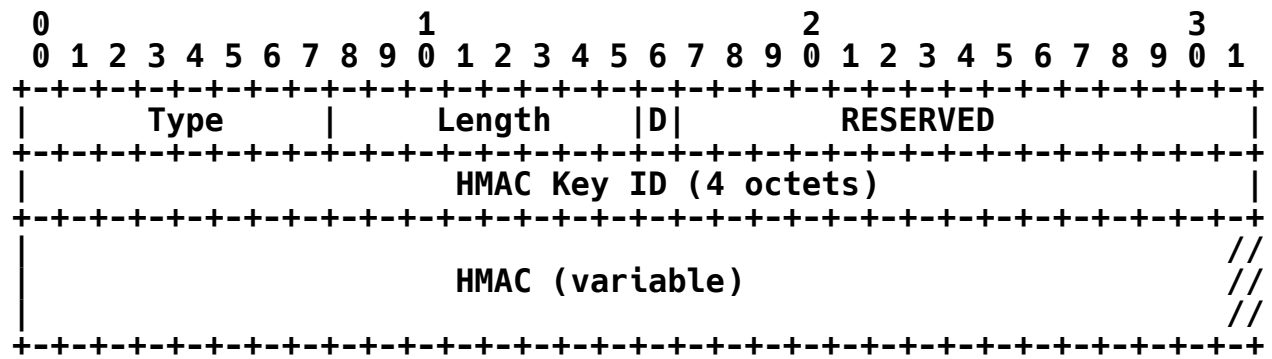
Padding: Padding bits have no semantic. They MUST be set to 0 on transmission and ignored on receipt.

The PadN TLV MUST be used when more than one byte of padding is required.

2.1.2. HMAC TLV

Alignment requirement: 8n

The keyed Hashed Message Authentication Code (HMAC) TLV is OPTIONAL and has the following format:



where:

Type: 5.

Length: The length of the variable-length data in bytes.

D: 1 bit. 1 indicates that the Destination Address verification is disabled due to use of a reduced Segment List (see Section 4.1.1).

RESERVED: 15 bits. MUST be 0 on transmission.

HMAC Key ID: A 4-octet opaque number that uniquely identifies the pre-shared key and algorithm used to generate the HMAC.

HMAC: Keyed HMAC, in multiples of 8 octets, at most 32 octets.

The HMAC TLV is used to verify that the SRH applied to a packet was selected by an authorized party and to ensure that the segment list is not modified after generation. This also allows for verification that the current segment (by virtue of being in the authorized Segment List) is authorized for use. The SR domain ensures that the source node is permitted to use the source address in the packet via ingress filtering mechanisms as defined in BCP 84 [RFC3704] or other strategies as appropriate.

2.1.2.1. HMAC Generation and Verification

Local configuration determines when to check for an HMAC. This local configuration is outside the scope of this document. It may be based on the active segment at an SR Segment endpoint node, the result of an Access Control List (ACL) that considers incoming interface, HMAC Key ID, or other packet fields.

An implementation that supports the generation and verification of the HMAC supports the following default behavior, as defined in the remainder of this section.

The HMAC verification begins by checking that the current segment is equal to the destination address of the IPv6 header. The check is successful when either:

* HMAC D bit is 1 and Segments Left is greater than Last Entry, or

* HMAC Segments Left is less than or equal to Last Entry, and the destination address is equal to Segment List[Segments Left].

The HMAC field is the output of the HMAC computation as defined in [RFC2104], using:

* key: The pre-shared key identified by HMAC Key ID

* HMAC algorithm: Identified by the HMAC Key ID

* Text: A concatenation of the following fields from the IPv6 header and the SRH, as it would be received at the node verifying the

HMAC:

- IPv6 header: Source address (16 octets)
- SRH: Last Entry (1 octet)
- SRH: Flags (1 octet)
- SRH: HMAC 16 bits following Length
- SRH: HMAC Key ID (4 octets)
- SRH: All addresses in the Segment List (variable octets)

The HMAC digest is truncated to 32 octets and placed in the HMAC field of the HMAC TLV.

For HMAC algorithms producing digests less than 32 octets long, the digest is placed in the lowest-order octets of the HMAC field. Subsequent octets MUST be set to zero such that the HMAC length is a multiple of 8 octets.

If HMAC verification is successful, processing proceeds as normal.

If HMAC verification fails, an ICMP error message (parameter problem, error code 0, pointing to the HMAC TLV) SHOULD be generated (but rate limited) and logged, and the packet SHOULD be discarded.

2.1.2.2. HMAC Pre-shared Key Algorithm

The HMAC Key ID field allows for the simultaneous existence of several hash algorithms (SHA-256, SHA3-256 ... or future ones) as well as pre-shared keys.

The HMAC Key ID field is opaque -- i.e., it has neither syntax nor semantic except as an identifier of the right combination of pre-shared key and hash algorithm.

At the HMAC TLV generating and verification nodes, the Key ID uniquely identifies the pre-shared key and HMAC algorithm.

At the HMAC TLV generating node, the Text for the HMAC computation is set to the IPv6 header fields and SRH fields as they would appear at the verification node(s), not necessarily the same as the source node sending a packet with the HMAC TLV.

Pre-Shared key rollover is supported by having two key IDs in use while the HMAC TLV generating node and verifying node converge to a new key.

The HMAC TLV generating node may need to revoke an SRH for which it previously generated an HMAC. Revocation is achieved by allocating a new key and key ID, then rolling over the key ID associated with the SRH to be revoked. The HMAC TLV verifying node drops packets with the revoked SRH.

An implementation supporting HMAC can support multiple hash functions. An implementation supporting HMAC MUST implement SHA-2 [FIPS180-4] in its SHA-256 variant.

The selection of pre-shared key and algorithm and their distribution is outside the scope of this document. Some options may include:

* setting these items in the configuration of the HMAC generating or verifying nodes, either by static configuration or any SDN-oriented approach

* dynamically using a trusted key distribution protocol such as [RFC6407]

While key management is outside the scope of this document, the recommendations of BCP 107 [RFC4107] should be considered when choosing the key management system.

3. SR Nodes

There are different types of nodes that may be involved in segment routing networks: SR source nodes that originate packets with a segment in the destination address of the IPv6 header, transit nodes that forward packets destined to a remote segment, and SR segment endpoint nodes that process a local segment in the destination address of an IPv6 header.

3.1. SR Source Node

A SR source node is any node that originates an IPv6 packet with a segment (i.e., SRv6 SID) in the destination address of the IPv6 header. The packet leaving the SR source node may or may not contain an SRH. This includes either:

* A host originating an IPv6 packet, or

* An SR domain ingress router encapsulating a received packet in an outer IPv6 header, followed by an optional SRH.

It is out of the scope of this document to describe the mechanism through which a segment in the destination address of the IPv6 header and the Segment List in the SRH are derived.

3.2. Transit Node

A transit node is any node forwarding an IPv6 packet where the destination address of that packet is not locally configured as a segment or a local interface. A transit node is not required to be capable of processing a segment or SRH.

3.3. SR Segment Endpoint Node

An SR segment endpoint node is any node receiving an IPv6 packet where the destination address of that packet is locally configured as a segment or local interface.

4. Packet Processing

This section describes SRv6 packet processing at the SR source, Transit, and SR segment endpoint nodes.

4.1. SR Source Node

A source node steers a packet into an SR Policy. If the SR Policy results in a Segment List containing a single segment, and there is no need to add information to the SRH flag or add TLV; the DA is set to the single Segment List entry, and the SRH MAY be omitted.

When needed, the SRH is created as follows:

The Next Header and Hdr Ext Len fields are set as specified in [RFC8200].

The Routing Type field is set to 4.

The DA of the packet is set with the value of the first segment.

The first element of the SRH Segment List is the ultimate segment. The second element is the penultimate segment, and so on.

The Segments Left field is set to n-1, where n is the number of elements in the SR Policy.

The Last Entry field is set to n-1, where n is the number of elements in the SR Policy.

TLVs (including HMAC) may be set according to their specification.

The packet is forwarded toward the packet's Destination Address (the first segment).

4.1.1. Reduced SRH

When a source does not require the entire SID list to be preserved in the SRH, a reduced SRH may be used.

A reduced SRH does not contain the first segment of the related SR Policy (the first segment is the one already in the DA of the IPv6 header), and the Last Entry field is set to n-2, where n is the number of elements in the SR Policy.

4.2. Transit Node

As specified in [RFC8200], the only node allowed to inspect the Routing Extension Header (and therefore the SRH) is the node corresponding to the DA of the packet. Any other transit node MUST NOT inspect the underneath routing header and MUST forward the packet toward the DA according to its IPv6 routing table.

When a SID is in the destination address of an IPv6 header of a packet, it's routed through an IPv6 network as an IPv6 address. SIDs, or the prefix(es) covering SIDs, and their reachability may be

distributed by means outside the scope of this document. For example, [RFC5308] or [RFC5340] may be used to advertise a prefix covering the SIDs on a node.

4.3. SR Segment Endpoint Node

Without constraining the details of an implementation, the SR segment endpoint node creates Forwarding Information Base (FIB) entries for its local SIDs.

When an SRv6-capable node receives an IPv6 packet, it performs a longest-prefix-match lookup on the packet's destination address. This lookup can return any of the following:

* A FIB entry that represents a locally instantiated SRv6 SID
* A FIB entry that represents a local interface, not locally instantiated as an SRv6 SID
* A FIB entry that represents a nonlocal route
* No Match

4.3.1. FIB Entry Is a Locally Instantiated SRv6 SID

This document and section define a single SRv6 SID. Future documents may define additional SRv6 SIDs. In such a case, the entire content of this section will be defined in that document.

If the FIB entry represents a locally instantiated SRv6 SID, process the next header chain of the IPv6 header as defined in Section 4 of [RFC8200]. Section 4.3.1.1 describes how to process an SRH; Section 4.3.1.2 describes how to process an upper-layer header or the absence of a Next Header.

Processing this SID modifies the Segments Left and, if configured to process TLVs, it may modify the "variable-length data" of TLV types that change en route. Therefore, Segments Left is mutable, and TLVs that change en route are mutable. The remainder of the SRH (Flags, Tag, Segment List, and TLVs that do not change en route) are immutable while processing this SID.

4.3.1.1. SRH Processing

```
S01. When an SRH is processed {
S02.   If Segments Left is equal to zero {
S03.     Proceed to process the next header in the packet,
         whose type is identified by the Next Header field in
         the routing header.
S04.   }
S05.   Else {
S06.     If local configuration requires TLV processing {
S07.       Perform TLV processing (see TLV Processing)
S08.     }
S09.     max_last_entry  =  ( Hdr Ext Len /  2 ) - 1
S10.     If  ((Last Entry > max_last_entry) or
```

```
S11.          (Segments Left is greater than (Last Entry+1)) {
S12.        Send an ICMP Parameter Problem, Code 0, message to
            the Source Address, pointing to the Segments Left
            field, and discard the packet.
S13.      }
S14.      Else {
S15.        Decrement Segments Left by 1.
S16.        Copy Segment List[Segments Left] from the SRH to the
            destination address of the IPv6 header.
S17.        If the IPv6 Hop Limit is less than or equal to 1 {
S18.          Send an ICMP Time Exceeded -- Hop Limit Exceeded in
              Transit message to the Source Address and discard
              the packet.
S19.        }
S20.        Else {
S21.          Decrement the Hop Limit by 1
S22.          Resubmit the packet to the IPv6 module for transmission
              to the new destination.
S23.        }
S24.      }
S25.    }
S26. }
```

4.3.1.1.1. TLV Processing

Local configuration determines how TLVs are to be processed when the Active Segment is a local SID defined in this document. The definition of local configuration is outside the scope of this document.

For illustration purposes only, two example local configurations that may be associated with a SID are provided below.

```
Example 1:
For any packet received from interface I2
  Skip TLV processing

Example 2:
For any packet received from interface I1
  If first TLV is HMAC {
    Process the HMAC TLV
  }
  Else {
    Discard the packet
  }
```

4.3.1.2. Upper-Layer Header or No Next Header

When processing the upper-layer header of a packet matching a FIB entry locally instantiated as an SRv6 SID defined in this document:

```
IF (Upper-layer Header is IPv4 or IPv6) and
    local configuration permits {
  Perform IPv6 decapsulation
  Resubmit the decapsulated packet to the IPv4 or IPv6 module
}
```

```
ELSE {
  Send an ICMP parameter problem message to the Source Address and
  discard the packet.  Error code (4) "SR Upper-layer
  Header Error", pointer set to the offset of the upper-layer
  header.
}
```

A unique error code allows an SR source node to recognize an error in SID processing at an endpoint.

4.3.2. FIB Entry Is a Local Interface

If the FIB entry represents a local interface and is not locally instantiated as an SRv6 SID, the SRH is processed as follows:

> If Segments Left is zero, the node must ignore the routing header and proceed to process the next header in the packet, whose type is identified by the Next Header field in the routing header.
>
> If Segments Left is non-zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

4.3.3. FIB Entry Is a Nonlocal Route

Processing is not changed by this document.

4.3.4. FIB Entry Is a No Match

Processing is not changed by this document.

5. Intra-SR-Domain Deployment Model

The use of the SIDs exclusively within the SR domain and solely for packets of the SR domain is an important deployment model.

This enables the SR domain to act as a single routing system.

This section covers:

* securing the SR domain from external attempts to use its SIDs
* using the SR domain as a single system with delegation between components
* handling packets of the SR domain

5.1. Securing the SR Domain

Nodes outside the SR domain are not trusted: they cannot directly use the SIDs of the domain. This is enforced by two levels of access control lists:

1. Any packet entering the SR domain and destined to a SID within the SR domain is dropped. This may be realized with the following logic. Other methods with equivalent outcome are

considered compliant:

* Allocate all the SIDs from a block S/s

* Configure each external interface of each edge node of the domain with an inbound infrastructure access list (IACL) that drops any incoming packet with a destination address in S/s

* Failure to implement this method of ingress filtering exposes the SR domain to source-routing attacks, as described and referenced in [RFC5095]

2. The distributed protection in #1 is complemented with per-node protection, dropping packets to SIDs from source addresses outside the SR domain. This may be realized with the following logic. Other methods with equivalent outcome are considered compliant:

* Assign all interface addresses from prefix A/a

* At node k, all SIDs local to k are assigned from prefix Sk/sk

* Configure each internal interface of each SR node k in the SR domain with an inbound IACL that drops any incoming packet with a destination address in Sk/sk if the source address is not in A/a.

5.2. SR Domain as a Single System with Delegation among Components

All intra-SR-domain packets are of the SR domain. The IPv6 header is originated by a node of the SR domain and is destined to a node of the SR domain.

All interdomain packets are encapsulated for the part of the packet journey that is within the SR domain. The outer IPv6 header is originated by a node of the SR domain and is destined to a node of the SR domain.

As a consequence, any packet within the SR domain is of the SR domain.

The SR domain is a system in which the operator may want to distribute or delegate different operations of the outermost header to different nodes within the system.

An operator of an SR domain may choose to delegate SRH addition to a host node within the SR domain and delegate validation of the contents of any SRH to a more trusted router or switch attached to the host. Consider a top-of-rack switch T connected to host H via interface I. H receives an SRH (SRH1) with a computed HMAC via some SDN method outside the scope of this document. H classifies traffic it sources and adds SRH1 to traffic requiring a specific Service Level Agreement (SLA). T is configured with an IACL on I requiring verification of the SRH for any packet destined to the SID block of the SR domain (S/s). T checks and verifies that SRH1 is valid and contains an HMAC TLV; T then verifies the HMAC.

An operator of the SR domain may choose to have all segments in the SR domain verify the HMAC. This mechanism would verify that the SRH Segment List is not modified while traversing the SR domain.

5.3. MTU Considerations

An SR domain ingress edge node encapsulates packets traversing the SR domain and needs to consider the MTU of the SR domain. Within the SR domain, well-known mitigation techniques are RECOMMENDED, such as deploying a greater MTU value within the SR domain than at the ingress edges.

Encapsulation with an outer IPv6 header and SRH shares the same MTU and fragmentation considerations as IPv6 tunnels described in [RFC2473]. Further investigation on the limitation of various tunneling methods (including IPv6 tunnels) is discussed in [INTAREA-TUNNELS] and SHOULD be considered by operators when considering MTU within the SR domain.

5.4. ICMP Error Processing

ICMP error packets generated within the SR domain are sent to source nodes within the SR domain. The invoking packet in the ICMP error message may contain an SRH. Since the destination address of a packet with an SRH changes as each segment is processed, it may not be the destination used by the socket or application that generated the invoking packet.

For the source of an invoking packet to process the ICMP error message, the ultimate destination address of the IPv6 header may be required. The following logic is used to determine the destination address for use by protocol-error handlers.

* Walk all extension headers of the invoking IPv6 packet to the routing extension header preceding the upper-layer header.

  - If routing header is type 4 Segment Routing Header (SRH)

    o The SID at Segment List[0] may be used as the destination address of the invoking packet.

ICMP errors are then processed by upper-layer transports as defined in [RFC4443].

For IP packets encapsulated in an outer IPv6 header, ICMP error handling is as defined in [RFC2473].

5.5. Load Balancing and ECMP

For any interdomain packet, the SR source node MUST impose a flow label computed based on the inner packet. The computation of the flow label is as recommended in [RFC6438] for the sending Tunnel End Point.

For any intradomain packet, the SR source node SHOULD impose a flow

label computed as described in [RFC6437] to assist ECMP load balancing at transit nodes incapable of computing a 5-tuple beyond the SRH.

At any transit node within an SR domain, the flow label MUST be used as defined in [RFC6438] to calculate the ECMP hash toward the destination address. If a flow label is not used, the transit node would likely hash all packets between a pair of SR Edge nodes to the same link.

At an SR segment endpoint node, the flow label MUST be used as defined in [RFC6438] to calculate any ECMP hash used to forward the processed packet to the next segment.

5.6. Other Deployments

Other deployment models and their implications on security, MTU, HMAC, ICMP error processing, and interaction with other extension headers are outside the scope of this document.

6. Illustrations

This section provides illustrations of SRv6 packet processing at SR source, transit, and SR segment endpoint nodes.

6.1. Abstract Representation of an SRH

For a node k, its IPv6 address is represented as Ak, and its SRv6 SID is represented as Sk.

IPv6 headers are represented as the tuple of (source,destination). For example, a packet with source address A1 and destination address A2 is represented as (A1,A2). The payload of the packet is omitted.

An SR Policy is a list of segments. A list of segments 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.

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

* Source Address SA, Destination Addresses DA, and next header SRH.

* SRH with SID list <S1,S2,S3> with SegmentsLeft = SL.

* Note the difference between the <> and () symbols. <S1,S2,S3> represents a SID list where the leftmost segment is the first segment. In contrast, (S3,S2,S1; SL) represents the same SID list but encoded in the SRH Segment List format where the leftmost segment is the last segment. When referring to an SR Policy in a high-level use case, it is simpler to use the <S1,S2,S3> notation. When referring to an illustration of detailed behavior, the (S3,S2,S1; SL) notation is more convenient.

At its SR Policy headend, the Segment List <S1,S2,S3> results in SRH (S3,S2,S1; SL=2) represented fully as:

```
    Segments Left=2
    Last Entry=2
    Flags=0
    Tag=0
    Segment List[0]=S3
    Segment List[1]=S2
    Segment List[2]=S1
```

6.2. Example Topology

The following topology is used in examples below:

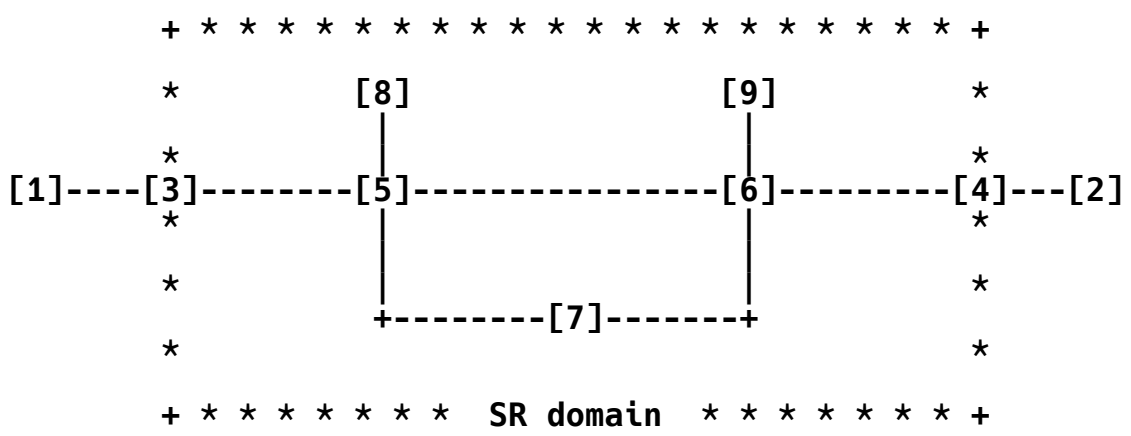


Figure 1

* 3 and 4 are SR domain edge routers

* 5, 6, and 7 are all SR domain routers

* 8 and 9 are hosts within the SR domain

* 1 and 2 are hosts outside the SR domain

* The SR domain implements ingress filtering as per Section 5.1 and no external packet can enter the domain with a destination address equal to a segment of the domain.

6.3. SR Source Node

6.3.1. Intra-SR-Domain Packet

When host 8 sends a packet to host 9 via an SR Policy <S7,A9> the packet is

P1: (A8,S7)(A9,S7; SL=1)

6.3.1.1. Reduced Variant

When host 8 sends a packet to host 9 via an SR Policy <S7,A9> and it wants to use a reduced SRH, the packet is

P2: (A8,S7)(A9; SL=1)

6.3.2. Inter-SR-Domain Packet -- Transit

When host 1 sends a packet to host 2, the packet is

P3: (A1,A2)

The SR domain ingress router 3 receives P3 and steers it to SR domain egress router 4 via an SR Policy <S7,S4>. Router 3 encapsulates the received packet P3 in an outer header with an SRH. The packet is

P4: (A3,S7)(S4,S7; SL=1)(A1,A2)

If the SR Policy contains only one segment (the egress router 4), the ingress router 3 encapsulates P3 into an outer header (A3,S4) without SRH. The packet is

P5: (A3,S4)(A1,A2)

6.3.2.1. Reduced Variant

The SR domain ingress router 3 receives P3 and steers it to SR domain egress router 4 via an SR Policy <S7,S4>. If router 3 wants to use a reduced SRH, it encapsulates the received packet P3 in an outer header with a reduced SRH. The packet is

P6: (A3,S7)(S4; SL=1)(A1,A2)

6.3.3. Inter-SR-Domain Packet -- Internal to External

When host 8 sends a packet to host 1, the packet is encapsulated for the portion of its journey within the SR domain. From 8 to 3 the packet is

P7: (A8,S3)(A8,A1)

In the opposite direction, the packet generated from 1 to 8 is

P8: (A1,A8)

At node 3, P8 is encapsulated for the portion of its journey within the SR domain, with the outer header destined to segment S8. This results in

P9: (A3,S8)(A1,A8)

At node 8, the outer IPv6 header is removed by S8 processing, then processed again when received by A8.

6.4. Transit Node

Node 5 acts as transit node for packet P1 and sends packet

P1: (A8,S7)(A9,S7;SL=1)

on the interface toward node 7.

6.5. SR Segment Endpoint Node

Node 7 receives packet P1 and, using the logic in Section 4.3.1, sends packet

P7: (A8,A9)(A9,S7; SL=0)

on the interface toward router 6.

6.6. Delegation of Function with HMAC Verification

This section describes how a function may be delegated within the SR domain. In the following sections, consider a host 8 connected to a top of rack 5.

6.6.1. SID List Verification

An operator may prefer to apply the SRH at source 8, while 5 verifies that the SID list is valid.

For illustration purposes, an SDN controller provides 8 an SRH terminating at node 9, with Segment List <S5,S7,S6,A9>, and HMAC TLV computed for the SRH. The HMAC key ID and key associated with the HMAC TLV is shared with 5. Node 8 does not know the key. Node 5 is configured with an IACL applied to the interface connected to 8, requiring HMAC verification for any packet destined to S/s.

Node 8 originates packets with the received SRH, including HMAC TLV.

P15: (A8,S5)(A9,S6,S7,S5;SL=3;HMAC)

Node 5 receives and verifies the HMAC for the SRH, then forwards the packet to the next segment

P16: (A8,S7)(A9,S6,S7,S5;SL=2;HMAC)

Node 6 receives

P17: (A8,S6)(A9,S6,S7,S5;SL=1;HMAC)

Node 9 receives

P18: (A8,A9)(A9,S6,S7,S5;SL=0;HMAC)

This use of an HMAC is particularly valuable within an enterprise-based SR domain [SRN].

7. Security Considerations

This section reviews security considerations related to the SRH, given the SRH processing and deployment models discussed in this document.

As described in Section 5, it is necessary to filter packets' ingress to the SR domain, destined to SIDs within the SR domain (i.e., bearing a SID in the destination address). This ingress filtering is

via an IACL at SR domain ingress border nodes. Additional protection is applied via an IACL at each SR Segment Endpoint node, filtering packets not from within the SR domain, destined to SIDs in the SR domain. ACLs are easily supported for small numbers of seldom changing prefixes, making summarization important.

Additionally, ingress filtering of IPv6 source addresses as recommended in BCP 38 [RFC2827] SHOULD be used.

7.1. SR Attacks

An SR domain implements distributed and per-node protection as described in Section 5.1. Additionally, domains deny traffic with spoofed addresses by implementing the recommendations in BCP 84 [RFC3704].

Full implementation of the recommended protection blocks the attacks documented in [RFC5095] from outside the SR domain, including bypassing filtering devices, reaching otherwise-unreachable Internet systems, network topology discovery, bandwidth exhaustion, and defeating anycast.

Failure to implement distributed and per-node protection allows attackers to bypass filtering devices and exposes the SR domain to these attacks.

Compromised nodes within the SR domain may mount the attacks listed above along with other known attacks on IP networks (e.g., DoS/DDoS, topology discovery, man-in-the-middle, traffic interception/ siphoning).

7.2. Service Theft

Service theft is defined as the use of a service offered by the SR domain by a node not authorized to use the service.

Service theft is not a concern within the SR domain, as all SR source nodes and SR segment endpoint nodes within the domain are able to utilize the services of the domain. If a node outside the SR domain learns of segments or a topological service within the SR domain, IACL filtering denies access to those segments.

7.3. Topology Disclosure

The SRH is unencrypted and may contain SIDs of some intermediate SR nodes in the path towards the destination within the SR domain. If packets can be snooped within the SR domain, the SRH may reveal topology, traffic flows, and service usage.

This is applicable within an SR domain, but the disclosure is less relevant as an attacker has other means of learning topology, flows, and service usage.

7.4. ICMP Generation

The generation of ICMPv6 error messages may be used to attempt

denial-of-service attacks by sending an error-causing destination address or SRH in back-to-back packets. An implementation that correctly follows Section 2.4 of [RFC4443] would be protected by the ICMPv6 rate-limiting mechanism.

7.5. Applicability of AH

The SR domain is a trusted domain, as defined in [RFC8402], Sections 2 and 8.2. The SR source is trusted to add an SRH (optionally verified as having been generated by a trusted source via the HMAC TLV in this document), and segments advertised within the domain are trusted to be accurate and advertised by trusted sources via a secure control plane. As such, the SR domain does not rely on the Authentication Header (AH) as defined in [RFC4302] to secure the SRH.

The use of SRH with AH by an SR source node and its processing at an SR segment endpoint node are not defined in this document. Future documents may define use of SRH with AH and its processing.

8. IANA Considerations

This document makes the following registrations in the "Internet Protocol Version 6 (IPv6) Parameters" "Routing Types" subregistry maintained by IANA:

Value	Description	Reference
4	Segment Routing Header (SRH)	This document

Table 1: SRH Registration

This document makes the following registrations in the "Type 4 - Parameter Problem" message of the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry maintained by IANA:

Code	Name
4	SR Upper-layer Header Error

Table 2: SR Upper-layer Header Error Registration

8.1. Segment Routing Header Flags Registry

This document describes a new IANA-managed registry to identify SRH Flags Bits. The registration procedure is "IETF Review" [RFC8126]. The registry name is "Segment Routing Header Flags". Flags are 8 bits.

8.2. Segment Routing Header TLVs Registry

This document describes a new IANA-managed registry to identify SRH

TLVs. The registration procedure is "IETF Review". The registry name is "Segment Routing Header TLVs". A TLV is identified through an unsigned 8-bit codepoint value, with assigned values 0-127 for TLVs that do not change en route and 128-255 for TLVs that may change en route. The following codepoints are defined in this document:

Value	Description	Reference
0	Pad1 TLV	This document
1	Reserved	This document
2	Reserved	This document
3	Reserved	This document
4	PadN TLV	This document
5	HMAC TLV	This document
6	Reserved	This document
124-126	Experimentation and Test	This document
127	Reserved	This document
252-254	Experimentation and Test	This document
255	Reserved	This document

Table 3: Segment Routing Header TLVs Registry

Values 1, 2, 3, and 6 were defined in draft versions of this specification and are Reserved for backwards compatibility with early implementations and should not be reassigned. Values 127 and 255 are Reserved to allow for expansion of the Type field in future specifications, if needed.

9. References

9.1. Normative References

[FIPS180-4]
National Institute of Standards and Technology (NIST), "Secure Hash Standard (SHS)", FIPS PUB 180-4, DOI 10.6028/NIST.FIPS.180-4, August 2015, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

[IANA-SRHTLV]
IANA, "Segment Routing Header TLVs", <https://www.iana.org/assignments/ipv6-parameters/>.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-

Hashing for Message Authentication", RFC 2104,
DOI 10.17487/RFC2104, February 1997,
<https://www.rfc-editor.org/info/rfc2104>.

[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>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in
IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473,
December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering:
Defeating Denial of Service Attacks which employ IP Source
Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827,
May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed
Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March
2004, <https://www.rfc-editor.org/info/rfc3704>.

[RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic
Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107,
June 2005, <https://www.rfc-editor.org/info/rfc4107>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302,
DOI 10.17487/RFC4302, December 2005,
<https://www.rfc-editor.org/info/rfc4302>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation
of Type 0 Routing Headers in IPv6", RFC 5095,
DOI 10.17487/RFC5095, December 2007,
<https://www.rfc-editor.org/info/rfc5095>.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain
of Interpretation", RFC 6407, DOI 10.17487/RFC6407,
October 2011, <https://www.rfc-editor.org/info/rfc6407>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme,
"IPv6 Flow Label Specification", RFC 6437,
DOI 10.17487/RFC6437, November 2011,
<https://www.rfc-editor.org/info/rfc6437>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label
for Equal Cost Multipath Routing and Link Aggregation in
Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011,
<https://www.rfc-editor.org/info/rfc6438>.

[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>.

[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>.

[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>.

9.2. Informative References

[INTAREA-TUNNELS] Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-10, 12 September 2019, <https://tools.ietf.org/html/draft-ietf-intarea-tunnels-10>.

[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>.

[RFC5308] Hopps, C., "Routing IPv6 with IS-IS", RFC 5308, DOI 10.17487/RFC5308, October 2008, <https://www.rfc-editor.org/info/rfc5308>.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <https://www.rfc-editor.org/info/rfc5340>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[SRN] Lebrun, D., Jadin, M., Clad, F., Filsfils, C., and O. Bonaventure, "Software Resolved Networks: Rethinking Enterprise Networks with IPv6 Segment Routing", 2018, <https://inl.info.ucl.ac.be/system/files/sosr18-final15-embedfonts.pdf>.

Acknowledgements

The authors would like to thank Ole Troan, Bob Hinden, Ron Bonica, Fred Baker, Brian Carpenter, Alexandru Petrescu, Punit Kumar Jaiswal, David Lebrun, Benjamin Kaduk, Frank Xialiang, Mirja Kühlewind, Roman Danyliw, Joe Touch, and Magnus Westerlund for their comments to this document.

Contributors

Kamran Raza, Zafar Ali, Brian Field, Daniel Bernier, Ida Leung, Jen Linkova, Ebben Aries, Tomoya Kosugi, Éric Vyncke, David Lebrun, Dirk Steinberg, Robert Raszuk, Dave Barach, John Brzozowski, Pierre Francois, Nagendra Kumar, Mark Townsley, Christian Martin, Roberta Maglione, James Connolly, and Aloys Augustin contributed to the

content of this document.

Authors' Addresses

Clarence Filsfils (editor)
Cisco Systems, Inc.
Brussels
Belgium

Email: cfilsfil@cisco.com

Darren Dukes (editor)
Cisco Systems, Inc.
Ottawa
Canada

Email: ddukes@cisco.com

Stefano Previdi
Huawei
Italy

Email: stefano@previdi.net

John Leddy
Individual
United States of America

Email: john@leddy.net

Satoru Matsushima
SoftBank

Email: satoru.matsushima@g.softbank.co.jp

Daniel Voyer
Bell Canada

Email: daniel.voyer@bell.ca