Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 6, 2015

S. Previdi, Ed.
C. Filsfils
Cisco Systems, Inc.
B. Field
Comcast
I. Leung
Rogers Communications
May 5, 2015

IPv6 Segment Routing Header (SRH)
draft-previdi-6man-segment-routing-header-06

Abstract

Segment Routing (SR) allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological, or application/service based) while maintaining per-flow state only at the ingress node to the SR domain.

Segment Routing can be applied to the IPv6 data plane with the addition of a new type of Routing Extension Header. This draft describes the Segment Routing Extension Header Type and how it is used by SR capable nodes.

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 http://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 November 6, 2015.

Table of Contents

1. Structure of this document

Section 3 gives an introduction on SR for IPv6 networks.

Section 4 describes the Segment Routing abstract model.

Section 5 defines the Segment Routing Header (SRH) allowing instantiation of SR over IPv6 dataplane.

Section 6 details the procedures of the Segment Routing Header.

2. Segment Routing Documents

Segment Routing terminology is defined in [I-D.ietf-spring-segment-routing].

Segment Routing use cases are described in [I-D.filsfils-spring-segment-routing-use-cases].

Segment Routing IPv6 use cases are described in [I-D.ietf-spring-ipv6-use-cases].

Segment Routing protocol extensions are defined in [I-D.ietf-isis-segment-routing-extensions], and [I-D.psenak-ospf-segment-routing-ospfv3-extension].

The security mechanisms of the Segment Routing Header (SRH) are described in [I-D.vyncke-6man-segment-routing-security].

3. Introduction

Segment Routing (SR), defined in [I-D.ietf-spring-segment-routing], allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological or service/application based) while maintaining per-flow state only at the ingress node to the SR domain. Segments can be derived from different components: IGP, BGP, Services, Contexts, Locators, etc. The list of segment forming the path is called the Segment List and is encoded in the packet header.

SR allows the use of strict and loose source based routing paradigms without requiring any additional signaling protocols in the infrastructure hence delivering an excellent scalability property.

The source based routing model described in [I-D.ietf-spring-segment-routing] is inherited from the ones proposed by [RFC1940] and [RFC2460]. The source based routing model offers the support for explicit routing capability.

3.1. Data Planes supporting Segment Routing

Segment Routing (SR), can be instantiated over MPLS ([I-D.ietf-spring-segment-routing-mpls]) and IPv6. This document defines its instantiation over the IPv6 data-plane based on the use-cases defined in [I-D.ietf-spring-ipv6-use-cases].

Segment Routing for IPv6 (SR-IPv6) is required in networks where MPLS data-plane is not used or, when combined with SR-MPLS, in networks where MPLS is used in the core and IPv6 is used at the edge (home networks, datacenters).

This document defines a new type of Routing Header (originally defined in [RFC2460]) called the Segment Routing Header (SRH) in order to convey the Segment List in the packet header as defined in [I-D.ietf-spring-segment-routing]. Mechanisms through which segment are known and advertised are outside the scope of this document.

3.2. Illustration

In the context of Figure 1 where all the links have the same IGP cost, let us assume that a packet P enters the SR domain at an ingress edge router I and that the operator requests the following requirements for packet P:

The local service S offered by node B must be applied to packet P.

The links AB and CE cannot be used to transport the packet P.

Any node N along the journey of the packet should be able to determine where the packet P entered the SR domain and where it will exit. The intermediate node should be able to determine the paths from the ingress edge router to itself, and from itself to the egress edge router.

Per-flow State for packet P should only be created at the ingress edge router.

The operator can forbid, for security reasons, anyone outside the operator domain to exploit its intra-domain SR capabilities.

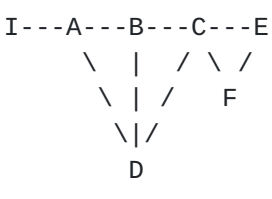


Figure 1: An illustration of SR properties

All these properties may be realized by instructing the ingress SR edge router I to push the following abstract SR header on the packet P.

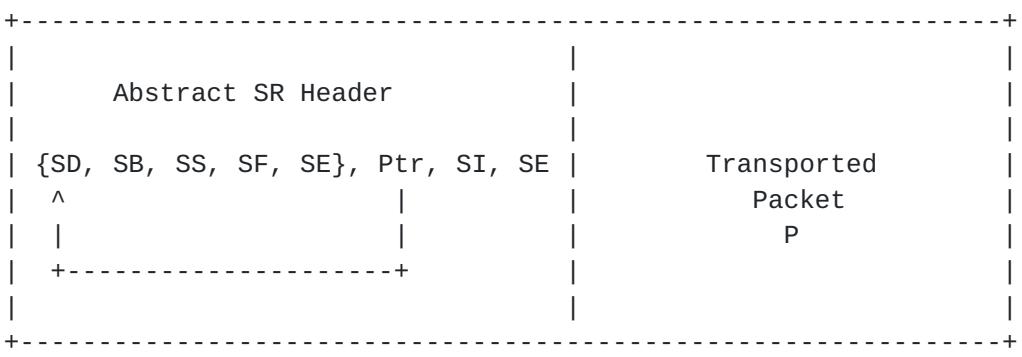


Figure 2: Packet P at node I

The abstract SR header contains a source route encoded as a list of segments {SD, SB, SS, SF, SE}, a pointer (Ptr) and the identification of the ingress and egress SR edge routers (segments SI and SE).

A segment identifies a topological instruction or a service instruction. A segment can either be global or local. The instruction associated with a global segment is recognized and executed by any SR-capable node in the domain. The instruction associated with a local segment is only supported by the specific node that originates it.

Let us assume some IGP (i.e.: ISIS and OSPF) extensions to define a "Node Segment" as a global instruction within the IGP domain to forward a packet along the shortest path to the specified node. Let us further assume that within the SR domain illustrated in Figure 1, segments SI, SD, SB, SE and SF respectively identify IGP node segments to I, D, B, E and F.

Let us assume that node B identifies its local service S with local segment SS.

With all of this in mind, let us describe the journey of the packet P.

The packet P reaches the ingress SR edge router. I pushes the SR header illustrated in Figure 2 and sets the pointer to the first segment of the list (SD).

SD is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to D.

Once at D, the pointer is incremented and the next segment is executed (SB).

SB is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to B.

Once at B, the pointer is incremented and the next segment is executed (SS).

SS is an instruction only recognized by node B which causes the packet to receive service S.

Once the service applied, the next segment is executed (SF) which causes the packet to be forwarded along the shortest path to F.

Once at F, the pointer is incremented and the next segment is executed (SE).

SE is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to E.

E then removes the SR header and the packet continues its journey outside the SR domain.

All of the requirements are met.

First, the packet P has not used links AB and CE: the shortest-path from I to D is I-A-D, the shortest-path from D to B is D-B, the shortest-path from B to F is B-C-F and the shortest-path from F to E is F-E, hence the packet path through the SR domain is I-A-D-B-C-F-E and the links AB and CE have been avoided.

Second, the service S supported by B has been applied on packet P.

Third, any node along the packet path is able to identify the service and topological journey of the packet within the SR domain. For example, node C receives the packet illustrated in Figure 3 and hence is able to infer where the packet entered the SR domain (SI), how it

got up to itself {SD, SB, SS, SE}, where it will exit the SR domain (SE) and how it will do so {SF, SE}.

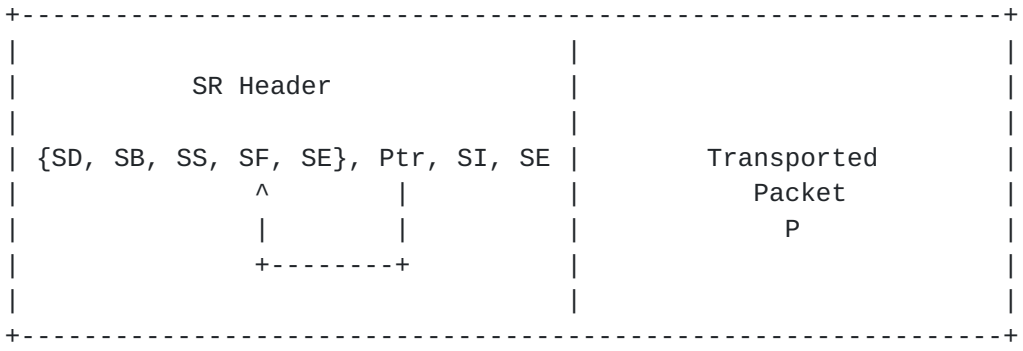


Figure 3: Packet P at node C

Fourth, only node I maintains per-flow state for packet P. The entire program of topological and service instructions to be executed by the SR domain on packet P is encoded by the ingress edge router I in the SR header in the form of a list of segments where each segment identifies a specific instruction. No further per-flow state is required along the packet path. The per-flow state is in the SR header and travels with the packet. Intermediate nodes only hold states related to the IGP global node segments and the local IGP adjacency segments. These segments are not per-flow specific and hence scale very well. Typically, an intermediate node would maintain in the order of 100's to 1000's global node segments and in the order of 10's to 100 of local adjacency segments. Typically the SR IGP forwarding table is expected to be much less than 10000 entries.

Fifth, the SR header is inserted at the entrance to the domain and removed at the exit of the operator domain. For security reasons, the operator can forbid anyone outside its domain to use its intra-domain SR capability.

4. Abstract Routing Model

At the entrance of the SR domain, the ingress SR edge router pushes the SR header on top of the packet. At the exit of the SR domain, the egress SR edge router removes the SR header.

The abstract SR header contains an ordered list of segments, a pointer identifying the next segment to process and the identifications of the ingress and egress SR edge routers on the path of this packet. The pointer identifies the segment that MUST be used by the receiving router to process the packet. This segment is called the active segment.

A property of SR is that the entire source route of the packet, including the identity of the ingress and egress edge routers is always available with the packet. This allows for interesting accounting and service applications.

We define three SR-header operations:

"PUSH": an SR header is pushed on an IP packet, or additional segments are added at the head of the segment list. The pointer is moved to the first entry of the added segments.

"NEXT": the active segment is completed, the pointer is moved to the next segment in the list.

"CONTINUE": the active segment is not completed, the pointer is left unchanged.

In the future, other SR-header management operations may be defined.

As the packet travels through the SR domain, the pointer is incremented through the ordered list of segments and the source route encoded by the SR ingress edge node is executed.

A node processes an incoming packet according to the instruction associated with the active segment.

Any instruction might be associated with a segment: for example, an intra-domain topological strict or loose forwarding instruction, a service instruction, etc.

At minimum, a segment instruction must define two elements: the identity of the next-hop to forward the packet to (this could be the same node or a context within the node) and which SR-header management operation to execute.

Each segment is known in the network through a Segment Identifier (SID). The terms "segment" and "SID" are interchangeable.

4.1. Segment Routing Global Block (SRGB)

In the SR abstract model, a segment is identified by a Segment Routing Identifier (SID). The SR abstract model doesn't mandate a specific format for the SID (IPv6 address or other formats).

In Segment Routing IPv6 the SID is an IPv6 address. Therefore, the SRGB is materialized by the global IPv6 address space which represents the set of IPv6 routable addresses in the SR domain. The following rules apply:

- Each node of the SR domain MUST be configured with the Segment Routing Global Block (SRGB).

- All global segments must be allocated from the SRGB. Any SR capable node MUST be able to process any global segment advertised by any other node within the SR domain.

- Any segment outside the SRGB has a local significance and is called a "local segment". An SR-capable node MUST be able to process the local segments it originates. An SR-capable node MUST NOT support the instruction associated with a local segment originated by a remote node.

4.2. Traffic Engineering with SR

An SR Traffic Engineering policy is composed of two elements: a flow classification and a segment-list to prepend on the packets of the flow.

In SR, this per-flow state only exists at the ingress edge node where the policy is defined and the SR header is pushed.

It is outside the scope of the document to define the process that leads to the instantiation at a node N of an SR Traffic Engineering policy.

[I-D.filsfils-spring-segment-routing-use-cases] illustrates various alternatives:

N is deriving this policy automatically (e.g. FRR).

N is provisioned explicitly by the operator.

N is provisioned by a controller or server (e.g.: SDN Controller).

N is provisioned by the operator with a high-level policy which is mapped into a path thanks to a local CSPF-based computation (e.g. affinity/SRLG exclusion).

N could also be provisioned by other means.

[I-D.filsfils-spring-segment-routing-use-cases] explains why the majority of use-cases require very short segment-lists, hence minimizing the performance impact, if any, of inserting and transporting the segment list.

A SDN controller, which desires to instantiate at node N an SR Traffic Engineering policy, collects the SR capability of node N such as to ensure that the policy meets its capability.

4.3. Segment Routing Database

The Segment routing Database (SRDB) is a set of entries where each entry is identified by a SID. The instruction associated with each entry at least defines the identity of the next-hop to which the packet should be forwarded and what operation should be performed on the SR header (PUSH, CONTINUE, NEXT).

Segment	Next-Hop	SR Header operation
Sk	M	CONTINUE
Sj	N	NEXT
Sl	NAT Srvc	NEXT
Sm	FW srvc	NEXT
Sn	Q	NEXT
etc.	etc.	etc.

Figure 4: SR Database

Each SR-capable node maintains its local SRDB. SRDB entries can either derive from local policy or from protocol segment advertisement.

5. IPv6 Instantiation of Segment Routing

5.1. Segment Identifiers (SIDs) and SRGB

Segment Routing, as described in [I-D.ietf-spring-segment-routing], defines Node-SID and Adjacency-SID. When SR is used over IPv6 data-plane the following applies.

The SRGB is the global IPv6 address space which represents the set of IPv6 routable addresses in the SR domain.

5.1.1. Node-SID

The Node-SID identifies a node. With SR-IPv6 the Node-SID is an IPv6 prefix that the operator configured on the node and that is used as the node identifier. Typically, in case of a router, this is the IPv6 address of the node loopback interface. Therefore, SR-IPv6 does not require any additional SID advertisement for the Node Segment. The Node-SID is in fact the IPv6 address of the node.

Node SIDs are IPv6 addresses part of the SRGB (i.e.: addresses of global scope).

5.1.2. Adjacency-SID

Adjacency-SIDs can be either globally scoped IPv6 addresses or any 128-bit identifier representing the adjacency. Obviously, in the latter case, the scope of the Adjacency-SID is local to the router and any packet with the a such Adjacency-SID would need first to reach the node through the node's Node-SID prior for the node to process the Adjacency-SID. In other wrods, two segments (SIDs) would then be required: the first is the node's Node-SID that brings the packet to the node and the second is the Adjacency-SID that will make the node to forward the packet through the interface the Adjacency-SID is allocated to.

In the SR architecture defined in [I-D.ietf-spring-segment-routing] the Adjacency-SID (or Adj-SID) is the segment identifier associated with the instruction of forwarding the packet through the interface the Adj-SID is assigned to. Adj-SIDs can be either globally scoped IPv6 addresses or any 128-bit identifier representing the adjacency. Obviously, in the latter case, the scope of the Adj-SID is local to the router and any packet with the a such Adj-SID would need first to reach the node through the node's Node-SID prior for the node to process the Adj-SID. In other wrods, two segments (SIDs) would then be required: the first is the node's Node-SID that brings the packet to the node and the second is the Adj-SID that will make the node to forward the packet through the interface the Adj-SID is allocated to. A node may advertise one (or more) Adj-SIDs allocated to the same interface as well as a node can advertise the same Adj-SID for multiple interfaces. Use cases of Adj-SID advertisements are described in [I-D.ietf-spring-segment-routing]The semantic of the Adj-SID is:

> Send out the packet to the interface this Adj-SID is allocated to.

When SR is applied to IPv6, Node-SIDs are a global IPv6 addresses and therefore, an Adj-SID has a global significance (i.e.: the IPv6 address representing the SID is a global address). In other words, a node that advertises the Adj-SID in the form of a global IPv6 address representing the link/adjacency the packet has to be forwarded to, will apply to the Adj-SID a global significance.

Advertisement of Adj-SID may be done using multiple mechanisms among which the ones described in ISIS and OSPF protocol extensions: [I-D.ietf-isis-segment-routing-extensions] and [I-D.psenak-ospf-segment-routing-ospfv3-extension]. The distinction

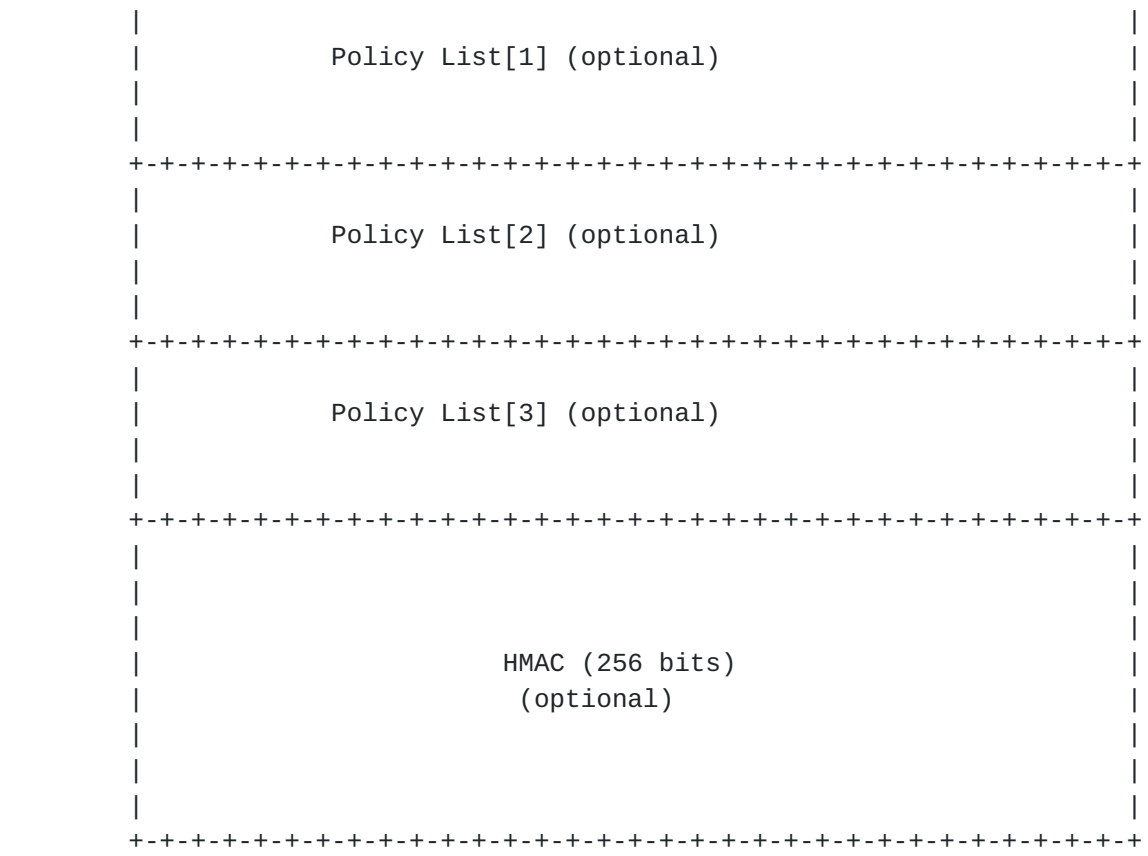
between local and global significance of the Adj-SID is given in the encoding of the Adj-SID advertisement.

5.2. Segment Routing Extension Header (SRH)

A new type of the Routing Header (originally defined in [RFC2460]) is defined: the Segment Routing Header (SRH) which has a new Routing Type, (suggested value 4) to be assigned by IANA.

As an example, if an explicit path is to be constructed across a core network running ISIS or OSPF, the segment list will contain SIDs representing the nodes across the path (loose or strict) which, usually, are the IPv6 loopback interface address of each node. If the path is across service or application entities, the segment list contains the IPv6 addresses of these services or application instances.

The Segment Routing Header (SRH) is defined as follows:



where:

- Next Header: 8-bit selector. Identifies the type of header immediately following the SRH.

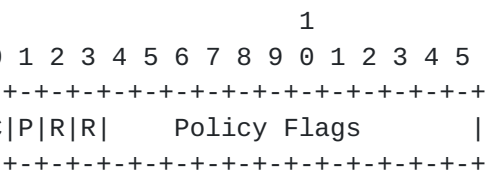
- Hdr Ext Len: 8-bit unsigned integer, is the length of the SRH header in 8-octet units, not including the first 8 octets.

- Routing Type: TBD, to be assigned by IANA (suggested value: 4).

- Segments Left. Defined in [RFC2460], it contains the index, in the Segment List, of the next segment to inspect. Segments Left is decremented at each segment and it is used as an index in the segment list.

- First Segment: offset in the SRH, not including the first 8 octets and expressed in 16-octet units, pointing to the last element of the segment list, which is in fact the first segment of the segment routing path.

- Flags: 16 bits of flags. Following flags are defined:



C-flag: Clean-up flag. Set when the SRH has to be removed from the packet when packet reaches the last segment.

P-flag: Protected flag. Set when the packet has been rerouted through FRR mechanism by a SR endpoint node. See Section 6.3 for more details.

R-flags. Reserved and for future use.

Policy Flags. Define the type of the IPv6 addresses encoded into the Policy List (see below). The following have been defined:

Bits 4-6: determine the type of the first element after the segment list.

Bits 7-9: determine the type of the second element.

Bits 10-12: determine the type of the third element.

Bits 13-15: determine the type of the fourth element.

The following values are used for the type:

0x0: Not present. If value is set to 0x0, it means the element represented by these bits is not present.

0x1: SR Ingress.

0x2: SR Egress.

0x3: Original Source Address.

- HMAC Key ID and HMAC field, and their use are defined in [I-D.vyncke-6man-segment-routing-security].

- Segment List[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 path. I.e., the first element of the segment list (Segment List [0]) contains the last segment of the path while the last segment of the Segment List (Segment List[n])

contains the first segment of the path. The index contained in "Segments Left" identifies the current active segment.

- Policy List. Optional addresses representing specific nodes in the SR path such as:

  SR Ingress: a 128 bit generic identifier representing the ingress in the SR domain (i.e.: it needs not to be a valid IPv6 address).

  SR Egress: a 128 bit generic identifier representing the egress in the SR domain (i.e.: it needs not to be a valid IPv6 address).

  Original Source Address: IPv6 address originally present in the SA field of the packet.

  The segments in the Policy List are encoded after the segment list and they are optional. If none are in the SRH, all bits of the Policy List Flags MUST be set to 0x0.

5.2.1. SRH and RFC2460 behavior

The SRH being a new type of the Routing Header, it also has the same properties:

SHOULD only appear once in the packet.

Only the router whose address is in the DA field of the packet header MUST inspect the SRH.

Therefore, Segment Routing in IPv6 networks implies that the segment identifier (i.e.: the IPv6 address of the segment) is moved into the DA of the packet.

The DA of the packet changes at each segment termination/completion and therefore the original DA of the packet MUST be encoded as the last segment of the path.

As illustrated in Section 3.2, nodes that are within the path of a segment will forward packets based on the DA of the packet without inspecting the SRH. This ensures full interoperability between SR-capable and non-SR-capable nodes.

6. SRH Procedures

In this section we describe the different procedures on the SRH.

6.1. Segment Routing Operations

When Segment Routing is instantiated over the IPv6 data plane the following applies:

- The segment list is encoded in the SRH.
- The active segment is in the destination address of the packet.
- The Segment Routing CONTINUE operation (as described in [I-D.ietf-spring-segment-routing]) is implemented as a regular/ plain IPv6 operation consisting of DA based forwarding.
- The NEXT operation is implemented through the update of the DA with the value represented by the Next Segment field in the SRH.
- The PUSH operation is implemented through the insertion of the SRH or the insertion of additional segments in the SRH segment list.

6.2. Segment Routing Node Functions

SR packets are forwarded to segments endpoints (i.e.: nodes whose address is in the DA field of the packet). The segment endpoint, when receiving a SR packet destined to itself, does:

- Inspect the SRH.
- Determine the next active segment.
- Update the Segments Left field (or, if requested, remove the SRH from the packet).
- Update the DA.
- Send the packet to the next segment.

The procedures applied to the SRH are related to the node function. Following nodes functions are defined:

Ingress SR Node.

Transit Non-SR Node.

Transit SR Intra Segment Node.

SR Endpoint Node.

6.2.1. Ingress SR Node

Ingress Node can be a router at the edge of the SR domain or a SR-capable host. The ingress SR node may obtain the segment list by either:

Local path computation.

Local configuration.

Interaction with an SDN controller delivering the path as a complete SRH.

Any other mechanism (mechanisms through which the path is acquired are outside the scope of this document).

The following are the steps of the creation of the SRH:

Next Header and Hdr Ext Len fields are set according to [RFC2460].

Routing Type field is set as TBD (SRH).

The Segment List is built with the FIRST segment of the path encoded in the LAST element of the Segment List. Subsequent segments are encoded on top of the first segment. Finally, the LAST segment of the path is encoded in the FIRST element of the Segment List. In other words, the Segment List is encoded in the reverse order of the path.

The original DA of the packet is encoded as the last segment of the path (encoded in the first element of the Segment List).

The DA of the packet is set with the value of the first segment (found in the last element of the segment list).

The Segments Left field is set to n-1 where n is the number of elements in the Segment List.

The First Segment field is set to n-1 where n is the number of elements in the Segment List.

The packet is sent out towards the first segment (i.e.: represented in the packet DA).

6.2.1.1. Security at Ingress

The procedures related to the Segment Routing security are detailed in [I-D.vyncke-6man-segment-routing-security].

In the case where the SR domain boundaries are not under control of the network operator (e.g.: when the SR domain edge is in a home network), it is important to authenticate and validate the content of any SRH being received by the network operator. In such case, the security procedure described in [I-D.vyncke-6man-segment-routing-security] is to be used.

The ingress node (e.g.: the host in the home network) requests the SRH from a control system (e.g.: an SDN controller) which delivers the SRH with its HMAC signature on it.

Then, the home network host can send out SR packets (with an SRH on it) that will be validated at the ingress of the network operator infrastructure.

The ingress node of the network operator infrastructure, is configured in order to validate the incoming SRH HMACs in order to allow only packets having correct SRH according to their SA/DA addresses.

6.2.2. Transit Non-SR Capable Node

SR is interoperable with plain IPv6 forwarding. Any non SR-capable node will forward SR packets solely based on the DA. There's no SRH inspection. This ensures full interoperability between SR and non-SR nodes.

6.2.3. SR Intra Segment Transit Node

Only the node whose address is in DA inspects and processes the SRH (according to [RFC2460]). An intra segment transit node is not in the DA and its forwarding is based on DA and its SR-IPv6 FIB.

6.2.4. SR Segment Endpoint Node

The SR segment endpoint node is the node whose address is in the DA. The segment endpoint node inspects the SRH and does:

```
1.   IF DA = myself (segment endpoint)
2.      IF Segments Left > 0 THEN
           decrement Segments Left
           update DA with Segment List[Segments Left]
3.         IF Segments Left == 0 THEN
              IF Clean-up bit is set THEN remove the SRH
4.      ELSE give the packet to next PID (application)
             End of processing.
5.   Forward the packet out
```

6.3. FRR Flag Settings

A node supporting SR and doing Fast Reroute (as described in [I-D.filsfils-spring-segment-routing-use-cases], when rerouting packets through FRR mechanisms, SHOULD inspect the rerouted packet header and look for the SRH. If the SRH is present, the rerouting node SHOULD set the Protected bit on all rerouted packets.

7. SR and Tunneling

Encapsulation can be realized in two different ways with SR-IPv6:

Outer encapsulation.

SRH with SA/DA original addresses.

Outer encapsulation tunneling is the traditional method where an additional IPv6 header is prepended to the packet. The original IPv6 header being encapsulated, everything is preserved and the packet is switched/routed according to the outer header (that could contain a SRH).

SRH allows encoding both original SA and DA, hence an operator may decide to change the SA/DA at ingress and restore them at egress. This can be achieved without outer encapsulation, by changing SA/DA and encoding the original SA in the Policy List and in the original DA in the Segment List.

8. Example Use Case

A more detailed description of use cases are available in [I-D.ietf-spring-ipv6-use-cases]. In this section, a simple SR-IPv6 example is illustrated.

In the topology described in Figure 6 it is assumed an end-to-end SR deployment. Therefore SR is supported by all nodes from A to J.

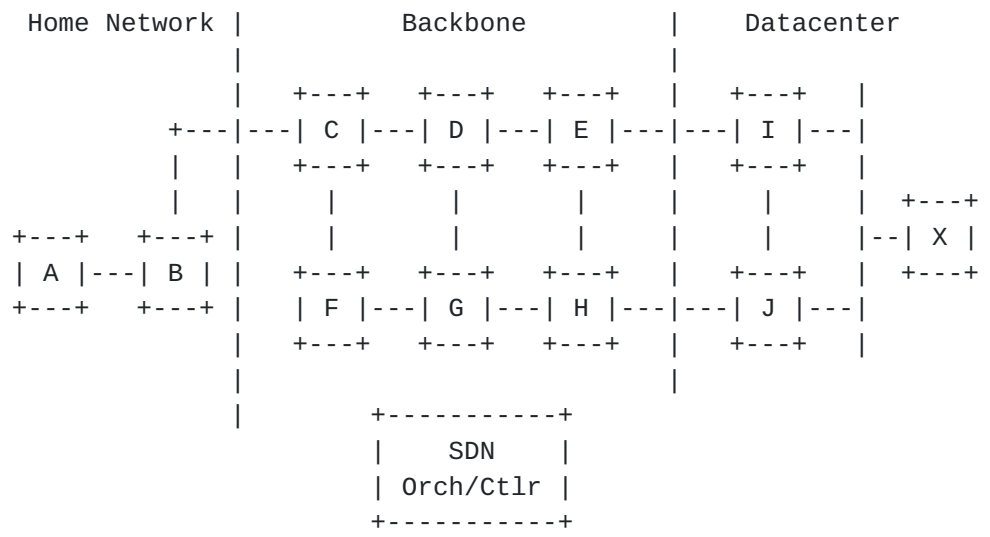


Figure 6: Sample SR topology

The following workflow applies to packets sent by host A and destined to server X.

- Host A sends a request for a path to server X to the SDN controller or orchestration system.

- The SDN controller/orchestrator builds a SRH with:
  - Segment List: C, F, J, X
  - HMAC

  that satisfies the requirements expressed in the request by host A and based on policies applicable to host A.

- Host A receives the SRH and insert it into the packet. The packet has now:
  - SA: A
  - DA: C
  - SRH with
    - SL: X, J, F, C
    - Segments Left: 3 (i.e.: Segment List size - 1)
    - PL: C (ingress), J (egress)

    Note that X is the last segment and C is the first segment (i.e.: the SL is encoded in the reverse path order).
  - HMAC

- When packet arrives in C (first segment), C does:
  - Validate the HMAC of the SRH.
  - Decrement Segments Left by one: 2
  - Update the DA with the next segment found in Segment List[2]. DA is set to F.
  - Forward the packet to F.

- When packet arrives in F (second segment), F does:
  - Decrement Segments Left by one: 1
  - Update the DA with the next segment found in Segment List[1]. DA is set to J.
  - Forward the packet to J.

- Packet travels across G and H nodes which do plain IPv6 forwarding based on DA. No inspection of SRH needs to be done in these nodes. However, any SR capable node is allowed to set the Protected bit in case of FRR protection.

- When packet arrives in J (third segment), J does:
  - Decrement Segments Left by one: 0
  - Update the DA with the next segment found in Segment List[0]. DA is set to X.
  - If the cleanup bit is set, then node J will strip out the SRH from the packet.
  - Forward the packet to X.

The packet arrives in the server that may or may not support SR. The return traffic, from server to host, may be sent using the same procedures.

9. IANA Considerations

TBD

10. Manageability Considerations

TBD

11. Security Considerations

Security mechanisms applied to Segment Routing over IPv6 networks are detailed in [I-D.vyncke-6man-segment-routing-security].

12. Contributors

The authors would like to thank Dave Barach, John Leddy, John Brzozowski, Pierre Francois, Nagendra Kumar, Mark Townsley, Christian Martin, Roberta Maglione, Eric Vyncke, James Connolly, David Lebrun and Fred Baker for their contribution to this document.

13. Acknowledgements

TBD

14. References

14.1. Normative References

[I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-03 (work in progress), October 2014.

[I-D.psenak-ospf-segment-routing-ospfv3-extension]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-psenak-ospf-segment-routing-ospfv3-extension-02 (work in progress), July 2014.

[I-D.vyncke-6man-segment-routing-security]
Vyncke, E., Previdi, S., and D. Lebrun, "IPv6 Segment Routing Security Considerations", draft-vyncke-6man-segment-routing-security-02 (work in progress), February 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

14.2. Informative References

[I-D.filsfils-spring-segment-routing-use-cases]
Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-01 (work in progress), October 2014.

[I-D.ietf-spring-ipv6-use-cases]
Brzozowski, J., Leddy, J., Leung, I., Previdi, S., Townsley, W., Martin, C., Filsfils, C., and R. Maglione, "IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-cases-04 (work in progress), March 2015.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-00 (work in progress), December 2014.

[RFC1940] Estrin, D., Li, T., Rekhter, Y., Varadhan, K., and D. Zappala, "Source Demand Routing: Packet Format and Forwarding Specification (Version 1)", RFC 1940, May 1996.

Authors' Addresses

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Brian Field
Comcast
4100 East Dry Creek Road
Centennial, CO 80122
US

Email: Brian_Field@cable.comcast.com

Ida Leung
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
CA

Email: Ida.Leung@rci.rogers.com