BMWG
Internet-Draft
Intended status: Informational
Expires: 23 April 2026

G. Fioccola
E. Vasilenko
P. Volpato
Huawei Technologies
L. Contreras
Telefonica
B. Decraene
Orange

20 October 2025

Benchmarking Methodology for Segment Routing
draft-ietf-bmwg-sr-bench-meth-05

Abstract

This document defines a methodology for benchmarking Segment Routing (SR) performance for Segment Routing over IPv6 (SRv6) and MPLS (SR-MPLS). It builds upon RFC 2544, RFC 5180, RFC 5695 and RFC 8402.

Table of Contents

1. Introduction

Segment Routing (SR), defined in [RFC8402], leverages the source routing paradigm. The headend node steers a packet through an SR Policy [RFC9256], instantiated as an ordered list of segments. A segment, referred to by its Segment Identifier (SID), can have a semantic local to an SR node or global within an SR domain. SR supports per-class explicit routing while maintaining per-class state only at the ingress nodes to the SR domain.

However, there is no standard method defined to compare and contrast the foundational SR packet forwarding capabilities of network devices. This document aims to extend the efforts of [RFC1242], [RFC2544], [RFC5180] and [RFC5695] to the SR network.

The SR architecture can be instantiated on two data-planes: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6). SRv6 has a variant with compressed SID ([RFC9800]).

SR can be directly applied to the Multiprotocol Label Switching (MPLS) architecture [RFC8660]. A segment is encoded as an MPLS label. An SR Policy is instantiated as a stack of labels.

SR can be applied to the IPv6 architecture with a new type of routing header called the SR Header (SRH) [RFC8754]. An instruction is associated with a segment and encoded as an IPv6 address. An SRv6 segment is also called an SRv6 SID. An SR Policy is instantiated as an ordered list of SRv6 SIDs in the routing header. The active segment is indicated by the Destination Address (DA) of the packet. A few compressed SIDs may be directly populated into the DA according to [RFC9800].

SR involves 3 types of forwarding plane operations (PUSH/ NEXT/ CONTINUE) as further described in Section 2 and Section 3. SR Segment List for PUSH operation is typically constructed by the source node with an SR Policy, see [RFC9256].

The SID stack in the scope of this document has a minimum of two entries, e.g. two SIDs. However, it is RECOMMENDED that the tests described in the next sections can be applied to label stacks with more than two SIDs. The reason for having a minimum of two SIDs, hence two labels, is to simulate a SID list, e.g. to simulate the explicit steering of a packet flow through different paths/nodes. It SHOULD be tested until the maximum SID depth is supported or claimed by the equipment. In this way, it is possible to identify the performance impact of a large SID list, ideally, all SID depths between two SIDs and the maximum SID depth can be tested. It is RECOMMENDED to test a big enough SID list to fill at least one

compressed SID container (i.e. all 128 bits) for the chosen compressed SID size including one additional SID of any type (the last one may be not compressed).

This document is limited to underlay, like Headend encapsulations (H.Encaps.xxx), segment Endpoints (End, End.X), Endpoints with decapsulations (End.Dxxx) and Binding (End.Bxxx) for SRv6. Compressed SID [RFC9800] is also considered in this document.

[RFC5695] describes a methodology specific to the benchmarking of MPLS forwarding devices, by considering the most common MPLS packet forwarding scenarios and corresponding performance measurements.

[RFC5180] provides benchmarking methodology recommendations that address IPv6-specific aspects, such as evaluating the forwarding performance of traffic containing extension headers.

The purpose of this document is to describe a methodology specific to the benchmarking of Segment Routing. The methodology described is a complement for [RFC5180] and [RFC5695].

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119], RFC 8174 [RFC8174].

2. SR-MPLS Forwarding

SR leverages the source routing paradigm. In MPLS, the ordered list of segments is encoded as a stack of MPLS labels. An SR Policy is instantiated through the MPLS Label Stack: the Segment IDs (SIDs) of a Segment List are inserted as MPLS Labels. The classical forwarding functions available for MPLS networks allow implementing the SR operations. However, SR-MPLS Segment List typically contains more labels.

The operations applied by the SR-MPLS forwarding plane are PUSH, NEXT, and CONTINUE.

The SR PUSH operation corresponds to the MPLS Label Push function [RFC3032]. It consists of pushing one or more MPLS labels on top of an incoming packet then sending it out of a particular physical or virtual interface towards a particular next hop.

The NEXT operation corresponds to the Label Pop function, which consists of removing the topmost label. The action associated with the popping depends on the instruction associated with the active SID on the received packet before the popping.

The CONTINUE operation corresponds to the Label Swap function, according to the MPLS label-swapping rules in [RFC3031]. It consists of associating an incoming label with an outgoing interface and outgoing label and forwarding the packet to the outgoing interface.

The encapsulation of an IP packet into an SR-MPLS packet is performed at the edge of an SR-MPLS domain, reusing the MPLS Forwarding Equivalent Class (FEC) concept. A Forwarding Equivalent Class (FEC) can be associated with an SR Policy ([RFC9256]). When pushing labels onto a packet’s label stack, the Time-to-Live (TTL) field and the Traffic Class (TC) field of each label stack entry must also be set.

All SR nodes in the SR domain use a signaling mechanism to advertise their prefix SIDs, as also detailed in Section 4.2. After receiving the advertised prefix SIDs, each SR node calculates the prefix SIDs to the advertisers. The prefix SID advertisement can be an absolute value advertisement or an index value advertisement. In this regard, the mapping of Segments to MPLS Labels (SIDs) is an important process in the SR-MPLS data plane. Each router can advertise its own available label space to be used for Global Segments called Segment Routing Global Block (SRGB) and an identical range of labels (SRGB) should be used in all routers to simplify services and operations. In the SR domain Global Segments can be identified by an index, which has to be re-mapped into a label, or by an absolute value. This is relevant for the nodes that perform the NEXT operation to the segments, because the label for the next segments needs to be crafted accordingly.

[RFC9256] specifies the concepts of SR Policy and steering into an SR Policy. The header of a packet steered in an SR Policy is augmented with the ordered list of segments associated with that SR Policy. SR Policy state is instantiated only on the headend node, which steers a flow into an SR Policy. Intermediate and endpoint nodes do not require any per policy state to be maintained. SR Policies can be instantiated on the headend dynamically and on-demand basis. SR policy may be installed by PCEP [RFC8664], BGP [I-D.ietf-idr-sr-policy-safi], [I-D.ietf-idr-bgp-sr-segtypes-ext], or via manual configuration on the router. PCEP and BGP signaling of SR Policies can be the case of a controller-based deployment.

3. SRv6 Forwarding

SR leverages the source routing paradigm. In SRv6, a SID is allocated as an IPv6 address. For the IPv6 data plane, a new type of IPv6 Routing Extension Header, called Segment Routing Header (SRH) has been defined [RFC8754]. The SRH contains the Segment List as an ordered list of IPv6 addresses: each address in the list is a SID. Hence SRv6 Segment list typically contains more than two SIDs. A dedicated field, referred to as Segments Left, is used to maintain the pointer to the active SID of the Segment List.

Three different categories of nodes may be involved in segment routing networks.

The SR source node is the headend node and steers a packet into an SR Policy. It can be a host originating an IPv6 packet or an SR domain ingress router encapsulating a received packet into an outer IPv6 packet and insert the SRH in the outer IPv6 header. It sets the first SID of the SR Policy as the IPv6 Destination Address of the packet.

The SR transit node forwards packets destined for a remote segment as a normal IPv6 packet based on the IPv6 destination address, because the IPv6 destination address does not locally match with a segment. According to [RFC8200] the only node allowed to inspect the Routing Extension Header (and therefore the SRH) is the node corresponding to the destination address of the packet.

The SR segment endpoint node receives packets whose IPv6 destination address is locally configured as a segment. It creates Forwarding Information Base (FIB) entries for its local SIDs. For each SR packet, it inspects the SRH, may prepare some actions (like forwarding through a particular interface), and then replaces the IPv6 destination address with the new active segment.

The operations applied by the SRv6 packet processing are different at the SR source, transit, and SR segment endpoint nodes.

The processing of the SR source node corresponds to the sequence of creation of an IPv6 packet with an SRH, composed of SIDs stored in reverse order, and setting of the IPv6 Destination Address as the first SID of the SR Policy. It can be performed by encapsulating a packet into an outer IPv6 packet with an SRH.

The processing of the SR segment endpoint node corresponds to the detection of the new active segment, which is the next segment in the Segment List and the related modification of the IPv6 destination address of the outer IPv6 header. Then packets are forwarded on the basis of the IPv6 forwarding table.

The processing of the SR transit node corresponds to the normal forwarding of the packets containing the SR header. In SRv6, the transit nodes do not need to be SRv6 aware, as every IPv6 router can act as an SRv6 transit node since any IPv6 node will maintain a plain IPv6 FIB entry for any prefix, no matter if the prefix represents a segment or not.

[RFC9256] specifies the concepts of SR Policy and steering into an SR Policy. The header of a packet steered in an SR Policy is augmented with the ordered list of segments associated with that SR Policy. SR Policy state is instantiated only on the headend node, which steers a flow into an SR Policy. Intermediate and endpoint nodes do not require any state to be maintained. SR Policies can be instantiated on the headend dynamically and on-demand basis. SR policy may be installed by PCEP [RFC8664], BGP [I-D.ietf-idr-sr-policy-safi], [I-D.ietf-idr-bgp-sr-segtypes-ext], or via manual configuration on the router. PCEP and BGP signaling of SR Policies can be the case of a controller-based deployment.

In addition to the basic SRv6 packet processing, the SRv6 Network Programming model [RFC8986] describes a set of functions that can be associated to segments and executed in a given SRv6 node.

Examples of such functions are described in [RFC8986], but, in practice, any behavior, and function can be associated with a local SID in a node, to apply any special processing on the packet. The definition of a standardized set of segment routing functions facilitates the deployment of SR domains with interoperable equipment from multiple vendors.

According to [RFC8986], 128 bit SID can be logically split into three fields and interpreted as LOCATOR:FUNCTION:ARGUMENTS (in short LOC:FUNCT:ARG) where LOC includes the L most significant bits, FUNCT the following F bits and ARG the remaining A bits, where L+F+A=128. The LOC corresponds to an IPv6 prefix (for example with a length of 48, 56, or 64 bits) that can be distributed by the routing protocols and provides the reachability of a node that hosts some functions. All the different functions residing in a node have a different FUNCT code, so that their SIDs will be different. The ARG bits are used to provide information (arguments) to a function. From the routing point of view, the solution is scalable, as a single prefix is distributed for a node, which implements a potentially large number of functions and related arguments.

LOCATOR consists of Locator-Block and Locator-Node. Locator-Block is common for all SIDs in the domain, it could be omitted for subsequent SIDs. Moreover, ARGUMENTS may not be needed for many types of SIDs. Then it is possible to compress some number of Locator-Nodes and/or Functions into the ARGUMENTS space of the initial SID as explained in [RFC9800]. It is assumed that initially the full SID list is contracted then it is compressed by one of two flavors (NEXT-C-SID or REPLACE-C-SID) if desired.

4. Test Methodology

4.1. Test Setup

The test setup in general is compliant with section 6 of [RFC2544] but augmented by the methodology specified in section 4 of [RFC5695] using many interfaces. It is needed to test the packet forwarding engine that may have different performance based on the number of interfaces served. The Device Under Test (DUT) may have oversubscribed interfaces, then traffic for such interfaces should be proportionally decreased according to the specific DUT oversubscription ratio. All interfaces served by a particular packet forwarding engine should be loaded in reverse proportion to the claimed oversubscription ratio. Tests SHOULD be done with bidirectional traffic that better reflects the real environment for SR nodes. It is OPTIONAL to choose a non-equal proportion for upstream and downstream traffic for some specific aggregation nodes.

The RECOMMENDED topology for SR Forwarding Benchmarking should be the same used for MPLS benchmarking, as described in section 4 of [RFC5695]. A simplified view is reported below for reference.

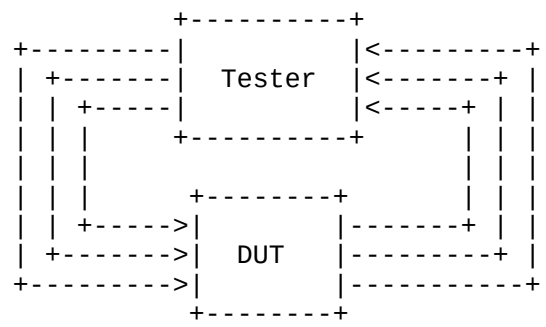

Figure 1: Test environment for SRv6 Forwarding Benchmarking

Differently from [RFC5695], this document prefers the use of the term "interface" instead of "port" as an interface may be either virtual or physical. Also, ports may be confused with TCP/UDP terms.

The RECOMMENDED topology for SR Forwarding Benchmarking should be the same as MPLS and it is described in section 4 of [RFC5695]. Interface numbers involved in the tests and their oversubscription ratio MUST be reported. This document is benchmarking only "source routing". Hence, SIDs represent only prefix and adjacency segments, that may be carried in IGP extensions. For the case of SRv6, SIDs represent only Headend encapsulation (H.Encaps.xxx) or segment Endpoint (End, End.X). In general, Services (L2/L3 VPNs and much more) are typically encoded by the last SID in the stack, but it is out of the scope of this document.

It is OPTIONAL to test SRH in combination with any other extension headers (fragmentation, hop-by-hop, destination options, etc.) but in all tests, the SRH header should be present for the test to be relevant for SRv6. It is RECOMMENDED to follow section 5.3 of [RFC5180] to introduce other extension headers in proportions 1%, 10%, 50% that may better reflect real use cases.

Segment Routing may also be implemented as a software network function in an NFV Infrastructure and, in this case, additional considerations should be done. [ETSI-GR-NFV-TST-007] describes test guidelines for NFV capabilities that require interactions between the components implementing NFV functionality.

Special capabilities SHOULD NOT exist in the DUT/SUT specifically for benchmarking purposes.

4.2. Control Plane Support

SRv6 and SR-MPLS have different terminology that is inherited from [RFC8402] for SR-MPLS and extended by [RFC8986] for SRv6.

As specified in [RFC8402], in the context of an IGP-based distributed control plane, two topological segments are defined: the IGP-Adjacency segment and the IGP-Prefix segment; while in the context of a BGP-based distributed control plane, two topological segments are defined: as the BGP peer segment and the BGP Prefix segment.

As specified in [RFC8986], topological segments have the structure that consists of Locator and Endpoint behavior (H.Encaps, End, End.X, etc), the latter may have a few different flavors (PSP, USP, USD). Different combinations of behavior and flavor are recommended for every test.

It is RECOMMENDED that the DUT and test tool support at least one option for SID stack construction:

* IS-IS Extensions to Support Segment Routing, [RFC8667] for SR-MPLS and [RFC9352] for SRv6

* OSPFv2 Extensions to Support Segment Routing, [RFC8665] for SR-MPLS.

* OSPFv3 Extensions to Support Segment Routing, [RFC8666] for SR-MPLS and [RFC9513] for SRv6

* Segment Routing Prefix Segment Identifier Extensions for BGP [RFC8669]

* Segment Routing Policy Architecture [RFC9256].

A routing protocol (OSPF or IS-IS) SHOULD be used for the construction of the first SRH SID. It is RECOMMENDED to test SR policy with a SID depth between two SIDs and the maximum SID depth supported.

The long SID list may be needed for extensive traffic engineering or other scenarios. The data plane needs to be compliant with the SRv6 control plane requirements (sections 4 of [RFC9513] and [RFC9352] and section 2 of [RFC9514]) to disclose the maximum SID list supported for encapsulation, decapsulation, and SRH deletion in transit. The SID list should not be tested for operations beyond the announced capabilities of OSPF or ISIS on the DUT, but, if there is an interest, it may be tested how the DUT reacts in this situation.

It is RECOMMENDED that the top SID on the list should emulate the traffic engineering scenario. In all cases, SID stack configuration SHOULD happen before packet forwarding is started. Control plane convergence speed is not the subject of the present tests.

It is important to point out that the control plane is independent of the SID list compression method used, if any.

The SID list construction method and SR policy construction method used MUST be reported according to Section 5.

4.3. Frame Formats and Sizes

SR tests will use Frame characteristics similar to section 4.1.5 of [RFC5695], except the need for a bigger MTU to accommodate SRH or MPLS SID stack.

It is assumed that MTU is big enough to accommodate all frame sizes proposed below. Fragmentation is not an option for Transit Segment Endpoint tests because it is prohibited in transit by [RFC8200] section 4.5: unlike IPv4, fragmentation in IPv6 is performed only by source nodes, not by routers along a packet’s delivery path. Fragmentation of IPv4 packet is not considered for Source Edge Node as this is not possible and hence not done for MPLS service so it is likely not implemented for SRv6 services.

It is to be noted that [RFC5695] requires exactly a single entry in the MPLS label stack in an MPLS packet that is not enough to simulate a typical SR SID list. The number of entries in SRH MUST be reported.

According to section 4.1.4.2 of [RFC5695], the payload is RECOMMENDED to have an IP packet (IPv6 or IPv4 with UDP or TCP) to better represent the real environment. The minimal Ethernet payload (46B) could not accommodate the whole IPv6 stack (not enough room for TCP or UDP), hence only IPv4 is possible to use if the test for minimal Ethernet payload is needed. It is possible to choose the bigger payload size for the IPv6-only environment. For the headend node, the frame size of the incoming interface(s) does not include SRH, therefore the outgoing interface(s) must support the increased frame size due to the creation of the SRH and outer IPv6 attachment.

It is assumed that the test would be for Ethernet media only. Other media is possible (see section 4.1.5.2 of [RFC5695] for the POS example). Some layer 2 technologies (like POS/PPP) have bit- or byte- stuffing then [RFC4814] may help to calculate real performance more accurately or else a 1-2% error is expected. The most popular layer 2 technology for SR is Ethernet, it does not have stuffing.

RECOMMENDED frame sizes are presented below. Any other frame sizes may be added if suspected of abnormal behavior. For example, some architectures may allocate buffer memory in big fixed chunks that may drop performance if frame sizes are chosen just a few octets more than the fixed chunk size (the second chunk would have a very low memory utilization).

The resulting Ethernet frame structure is depicted in the next figures.

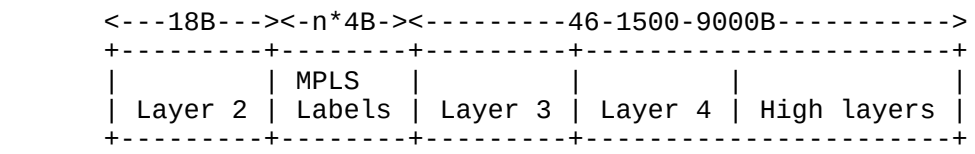

Figure 2: Ethernet Frame Structure for SR-MPLS

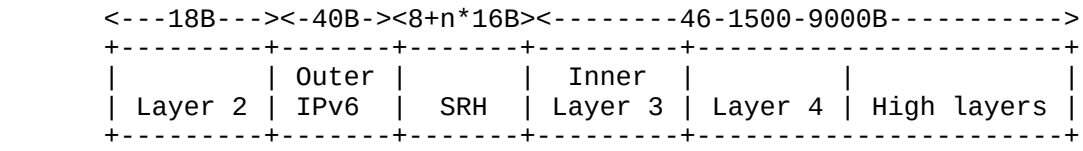

Figure 3: Ethernet Frame Structure for SRv6

RECOMMENDED payload sizes (encapsulated packet with L3 headers and above) are the following:

* Ethernet Minimal: 46

* DUT Minimal Wire Speed: typically 128-256 (it depends on the DUT specification)

* Ethernet Typical: 1500

* DUT Maximum: 9000 (or any claimed maximum)

Note that n*4 octets should be added in the previous calculations for SR-MPLS tests to accommodate MPLS labels needed for respective tests. While 40+8+n*16 bytes should be added for SRv6 tests, where

40 octets are added for the outer (tunnel) IPv6 header

8 octets are added for the SRH header itself

n is the number of SIDs multiplied by 16 octet SID size, one SID may have a few compressed SIDs.

The typical frame size values are listed above for the DUT minimal wire speed and maximum, they can be modified according to the DUT characteristics. The minimum wire speed frame size can be considered based on the DUT specification but, in some cases, many tests may be needed in the search for the real minimum wire speed frame size. VLAN tag may additionally increase the frame size. VLAN tag tests are OPTIONAL.

4.4. Protocol Addresses

IANA reserved an IPv6 address block 2001:0002::/48 ([RFC4773]) for use with IPv6 benchmark testing (see section 8 of [RFC5180]) and block 198.18.0.0/15 ([RFC3330]) for IPv4 benchmark testing. Source and destination addresses for the test streams SHOULD belong to the IPv6 range assigned by IANA. The type of infrastructure protocol (IPv6 vs IPv4) that should be used for IGP and BGP in the tests should be chosen according to the test purpose and requirements. It is not principal what Locator blocks would be chosen for tests. It may be /52, /56, /64, or even bigger. It is possible to test a few different Locator blocks if there is a need.

As it is discussed in section 3.1, there is a need to load the whole forwarding engine (on all interfaces). [RFC4814] discusses the importance of having many flows with address randomization for acceptable hash-based load balancing that is implemented in all forwarding engines. Note that IPv6 flow label randomization must be used, according to [RFC6438] and [RFC8754]. In the context of this document, it may also be relevant for SIDs, because SIDs may be used for hash to choose the next link (depending on DUT default or desired configuration). It is important to check what exactly is used for the hash load balancing algorithm on the DUT to keep these numbers sufficiently random and at volume. It is very often that IP addresses and transport protocol ports are used instead of SIDs for SR-MPLS.

4.5. Trial Duration

The test portion of each trial must take into account the respective protocol configuration. IGP protocols typically have a shorter hold time, while some BGP default configurations may be up to 180 seconds. It is needed to check the default hold time of the DUT for the respective protocol used.

The test portion of each trial SHOULD be at least 10 seconds longer than the hold time for respective protocol configuration to verify that the DUT is able to maintain a stable control plane when the data-forwarding plane is under stress. IGP protocols typically have shorter hold time, some BGP default configuration may be up to 180 seconds. It is needed to check the default hold time of the DUT for the respective protocol used.

4.6. Traffic Verification

Traffic verification is following section 10 of [RFC2544] and section 4.1.8 of [RFC5695]. The text is copied here for your convenience.

As stated in section 10 of [RFC2544], "the test equipment SHOULD discard any frames received during a test run that are not actual forwarded test frames. For example, keep-alive and routing update frames SHOULD NOT be included in the count of received frames. In all cases, sent traffic MUST be accounted for, whether it was received on the wrong interface, the correct interface, or not received at all. In all cases, the test equipment SHOULD verify the length of the received frames and check that they match the expected length.

Preferably, the test equipment SHOULD include sequence numbers (or signature) in the transmitted frames and check for these numbers on the received frames. If this is done, the reported results SHOULD include in addition to the number of frames dropped, the number of frames that were received out of order, the number of duplicated frames received and the number of gaps in the received frame numbering sequence".

Many test tools may, by default, only verify that they have received the embedded signature on the receive side. However, some SRv6 tests assume headers modifications (push or pop the MPLS label stack, add or delete SRH, replace destination address, adjust "segments left"). All packets SHOULD be checked for the correct header values on the receiving side.

In addition, section 4.1.8 of [RFC5695] requires that "the presence or absence of the MPLS label stack, every field value inside the label stack, if present, ethertype (0x8847 or 0x8848 versus 0x0800 or 0x86DD), frame sequencing, and frame check sequence (FCS) MUST be verified in the received frame". This is "to verify that the packets received by the test tool carry the expected MPLS label".

4.7. Buffer tests

Back-to-back frame test was initially discussed in section 26.4 [RFC2544] and later improved in [RFC9004] which is considered the comprehensive reference for back-to-back frame tests. Modern forwarding engines are typically flexible in the buffer distribution between different interfaces. Hence, like for all other benchmarking tests, it is important to stress the forwarding engine on all interfaces. It should be necessary to perform throughput tests first because only frame sizes that stress DUT below wire-speed can be used for back-to-back tests. Buffers would be filled with the rate equal to the difference between the theoretical maximum frame rate (wire-speed) and DUT measured throughput for the respective frame size.

The test time could be much shorter than recommended in [RFC9004] because typical SR DUT is hardware-based with claimed buffers between 30ms to 100ms. It is better to consult with the vendor to find a good starting search point. If DUT is software-based then [RFC9004] recommendation for 2-30 seconds is applied.

Queuing SHOULD NOT have weighted random early detection (WRED) or any other mechanism that may start dropping packets before the buffer is filled. Queuing SHOULD be configured for the tail drop which is, typically, a non-default configuration. Back-to-back frame test is rather complex and expensive (50 runs for every frame size). Hence, it is OPTIONAL for SR.

5. Reporting Format

There are a few parameters that must be changed in section 5 of [RFC5695] for SR tests.

Reporting parameter preserved from [RFC5695]:

* Throughput in bytes per second and frames per second
* Frame sizes in Octets (see Section 4.3)
* Interface speed (10/50/100/400/800/etc GE)
* Interface encapsulation (Ethernet or Ethernet VLAN)
* Interface media type (probably Ethernet)

Parameters changed from [RFC5695]:

* SR Forwarding Operations (PUSH/ NEXT/ CONTINUE).

* Label Distribution protocol and IGP are the same in the context of SR. Hence, it can be called "Label distribution methods" for SR-MPLS or "Locator and Endpoint behaviors methods" for SRv6.

New parameters that MUST be reported are:

* Interface numbers involved for ingress and egress in the tests and their respective oversubscription ratio.

* Upstream/downstream traffic proportion (equal bidirectional or some other split).

* Number of Segments considered in the SID list.

* Number of Segment Lists considered for the same Candidate Path.

* Number of Candidate Paths considered for an SR Policy.

* Number of SR Policies considered for a DUT.

* Compression method used: None, NEXT-C-SID, REPLACE-C-SID and compressed SID size.

* Behavior (H.Encaps, etc.) and Flavor (PSP, USP, USD) used for SRv6 tests (according to [RFC8986]).

* SR Policy construction method (PCEP, BGP, manual configuration).

* Type of the payload (IPv6/IPv4, UDP/TCP).

* Time to recover from the overload state

* Time to recover from the reset state and reset type (particular module in reset)

* Tested buffer size in frames with respective frame size (for the optional back-to-back test); it is possible to record calculated buffer time for wire-speed throughput in milliseconds.

Some parameters may be the same for all tests (like Media type or Ethernet encapsulation) then it may be reported one time.

6. SR Forwarding Benchmarking Tests

In general, tests are compliant with [RFC2544] but the important correction discussed in section 6 of [RFC2544] is applied: interfaces chosen for every test MUST stress all interfaces served by one forwarding engine. It is better to check the DUT specification for the relationship between interfaces and the forwarding engine to minimize the number of interfaces involved. However, it is possible to understand the worst case by looking at the throughput and latency from the trial tests. If any doubt exists about how full is the offered load for the forwarding engine then it is better to stress all interfaces of the line card or all interfaces for the whole router with a centralized forwarding engine. A partial load on the forwarding engine would show optimistic results. Controllable traffic distribution between many interfaces (as specified in section 4 of [RFC5695]) would need separate SID announcements for separate interfaces.

The performance of modern packet forwarding engines may be huge that may need to involve many testers to sufficiently load the DUT as presented in Figure 4. Then results correlation and recalculation of the real performance would be an additional burden.

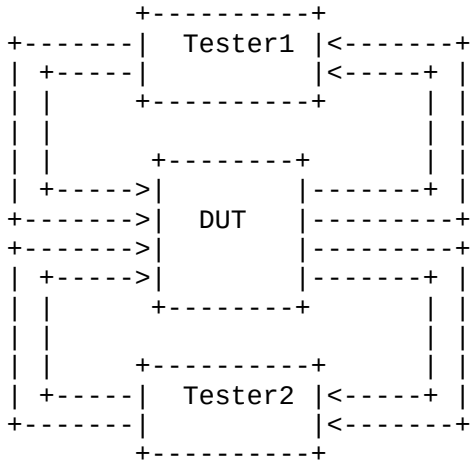

Figure 4: Many testers

As specified in section 6 of [RFC5695], the traffic is sent from test tool Tx interface(s) to the DUT at a constant load for a fixed-time interval, and is received from the DUT on test tool Rx interface(s). If any frame loss is detected, then a new iteration is needed where the offered load is decreased and the sender will transmit again. An iterative search algorithm MUST be used to determine the maximum

offered frame rate with a zero frame loss (Non Drop Rate). Each iteration should involve varying the offered load of the traffic, while keeping the other parameters (test duration, number of interfaces, number of addresses, frame size, etc.) constant, until the maximum rate at which none of the offered frames are dropped is determined.

The test can be repeated with a varying number of Segments pushed on ingress to measure the resulting maximum number. It can also be tested for the maximum number of Segments that are correctly load-balanced in transit by only changing the Nth label in the stack and detect when load-balancing fails.

Therefore, the two main parameters that can be evaluated are:

Maximum offered frame rate,

Maximum number of Segments that can be pushed and hashed by the SR node for load-balancing.

The test could be done to test more construction methods and consequently report the results as specified in Section 5. In addition, it could be possible to test ECMP (Equal-Cost Multi-Path) behavior of an SR Policy. An SR Policy with one active CP (Candidate Path) but with variable SL (Segments Left), SIDs, weights can be tested to check the overall performance and ECMP limits. All the related parameters must be reported as specified in Section 5.

Note that the test can also be done in the case of Compressed SRv6 Segment List Encoding [RFC9800].

6.1. Throughput

This section contains a description of the tests that are related to the characterization of a DUT’s SR traffic forwarding throughput.

The list of segments for SR-MPLS is represented as a stack of MPLS labels. There are three distinct operations to be tested: PUSH, NEXT and CONTINUE. These correspond to the three forwarding operations of an MPLS packet: PUSH (or LSP Ingress), POP (or LSP Egress), or SWAP.

The list of segments for SRv6 is represented as a list of IPv6 addresses, included in the SRH. Three distinct types of nodes are involved in segment routing networks that may represent four different cases.

Note that the different operations are separately discussed only for throughput tests, but they are equally applicable to the other tests below.

6.1.1. Throughput of a Source Edge Node

Objective: To obtain the DUT's Throughput during the packet processing of a Source Node, which is the PUSH forwarding operation. It is when the Source SR node, which corresponds to the headend node, encapsulates a received packet into SR-MPLS or SRv6.

In the case of SR-MPLS, the SID list is PUSHed to the MPLS label stack. It is similar to label Push or LSP Ingress forwarding operation, as per section 6.1.1 of [RFC5695] and section 26.1 of [RFC2544].

In the case of SRv6, the received packet is encapsulated in an IPv6 outer header including the SR Header (SRH) as a Routing Extension Header. The Segment List in the SRH is composed of SIDs and the Source SR node sets the first SID of the SR Policy as the IPv6 Destination Address of the packet. The RECOMMENDED headend behavior is H.Encaps, in case of interest for another behavior (H.Encaps.Red or H.Encaps.L2 or H.Encaps.L2.Red) it is OPTIONAL to test it with proper reporting. Additionally, the router could be configured for NEXT-C-SID or REPLACE-C-SID compression.

Procedure: Similar to section 6.1 of [RFC5695] or section 26.1 of [RFC2544] with extension to test SID list longer than 1 SID (more than 2 are RECOMMENDED). The SID list can be from 1 to N SIDs. N could be specified a priori or measured as part of the test. The test tool must advertise and learn the IP prefix(es) and SID(s) on respective sides, as per Section 4.4, and must use one option for the SID stack construction, as per Section 4.2, on its receive and transmit interfaces towards the DUT.

Reporting Format: A table with all parameters specified in Section 5.

6.1.2. Throughput of a Transit Segment Endpoint Node

Objective: To obtain the DUT's Throughput during the packet processing of a Segment Endpoint Node, which is the CONTINUE forwarding operation. It is when the SR Segment Endpoint node receives packets whose SID is locally configured as a segment.

In the case of SR-MPLS, it is equivalent to MPLS Label Swap or Ultimate Hop Popping (UHP), as per section 6.1.2 of [RFC5695] and section 26.1 of [RFC2544]. Non-reserved MPLS label values MUST be used.

In the case of SRv6, the SR Segment Endpoint node inspects the SR header: it detects the new active segment, i.e. the next segment in the Segment List, or index in the least significant bit for REPLACE-C-SID, modifies the IPv6 destination address of the outer IPv6 header, and forwards the packet based on the IPv6 forwarding table. The RECOMMENDED endpoint behavior is End.X, in case of interest for another behavior (End, End.T, End.BM, End.B6.Encaps, End.B6.Encaps.Red) it is OPTIONAL to test it with proper reporting. SRH SL is assumed to be bigger than zero for this test. Moreover, it is assumed that DUT would not need to delete headers (no PSP, USD, or USP). Additionally, the router could be configured for NEXT-C-SID or REPLACE-C-SID compression.

Procedure: Similar to section 6.1 of [RFC5695] or section 26.1 of [RFC2544] with extension to test SID list longer than 1 SID (more than 2 are RECOMMENDED). The SID list can be from 1 to N SIDs. N should be specified a priori or measured as part of the test. The test tool must advertise and learn the IP prefix(es) and SID(s) on respective sides, as per Section 4.4, and must use one option for the SID stack construction, as per Section 4.2, on its receive and transmit interfaces towards the DUT.

Reporting Format: A table with all parameters specified in Section 5.

6.1.3. Throughput of a Destination Edge Node

Objective: To obtain the DUT’s Throughput during the packet processing of a Segment Endpoint Node that needs decapsulation, which is the NEXT forwarding operation.

In the case of SR-MPLS, it is equivalent to MPLS Label Pop or Penultimate Hop Popping (PHP), as per section 6.1.3 of [RFC5695] and section 26.1 of [RFC2544].

In the case of SRv6, it is when the SR Segment Endpoint node receives packets whose IPv6 destination address is locally configured as a segment and SL in the SRH header is zero. The SR Segment Endpoint node decapsulates the packet, and forwards the packet based on the respective forwarding table (of the inner packet). The RECOMMENDED endpoint decapsulation behavior is End with the USD flavor, in case of interest for another flavor (PSP, USP) it is OPTIONAL to test it with proper reporting. Additionally, the router could be configured for NEXT-C-SID or REPLACE-C-SID compression.

Procedure: Similar to section 6.1 of [RFC5695] or section 26.1 of [RFC2544] with extension to test SID list longer than 1 SID (more than 2 are RECOMMENDED). The SID list can be from 1 to N SIDs. N

should be specified a priori or measured as part of the test. The test tool must advertise and learn the IP prefix(es) and SID(s) on respective sides, as per Section 4.4, and must use one option for the SID stack construction, as per Section 4.2, on its receive and transmit interfaces towards the DUT.

Reporting Format: A table with all parameters specified in Section 5.

6.1.4. Throughput of an Ordinary Transit Node

Objective: To obtain the DUT's Throughput during the packet processing of a Transit Node. It is when a Transit node forwards the packet containing the SR header as a normal IPv6 packet because the IPv6 destination address does not locally match with a segment. This test is possible only for SRv6, SR-MPLS requires all transit nodes to support MPLS.

Procedure: Similar to section 6.1 of [RFC5695] or section 26.1 of [RFC2544] with extension to test SID list longer than 1 SID (more than 2 are RECOMMENDED). The SID list can be from 1 to N SIDs. N should be specified a priori or measured as part of the test. The test tool must advertise and learn the IP prefix(es) and SID(s) on respective sides, as per Section 4.4, and must use one option for the SID stack construction, as per Section 4.2, on its receive and transmit interfaces towards the DUT.

Reporting Format: A table with all parameters specified in Section 5.

6.2. Buffer size

Back-to-back frame test is OPTIONAL and SHOULD be performed only after throughput tests because it SHOULD use only frame sizes that DUT is not capable to forward wire-speed, as explained in Section 4.7.

Objective: To determine the buffer size as defined in section 6 of [RFC9004] for each of the SR forwarding operations.

Procedure: Should be inherited from [RFC9004] with a SID list longer than 1 SID (more than 2 are RECOMMENDED). Despite the simple general idea for filling the buffer until the tail drop, [RFC9004] has many details for procedure, precautions, and calculations that would be too lengthy to copy here.

Reporting Format: A table with all parameters specified in Section 5.

6.3. Latency

Objective: To determine the latency as defined in section 6.2 of [RFC5695] and section 26.2 of [RFC2544] for each of the SR forwarding operations (PUSH, NEXT, CONTINUE). It is RECOMMENDED to test all three (for SR-MPLS) or four (for SRv6) test types discussed in Section 6.1.

Procedure: Similar to Section 6.1. It is OPTIONAL to improve the procedure according to section 7.2 of [RFC8219] with calculations for typical and worst-case latency.

Reporting Format: A table with all parameters specified in Section 5.

6.4. Frame Loss

Objective: To determine the frame-loss rate (as defined in section 6.3 of [RFC5695] and section 26.3 of [RFC2544]) for each of the SR forwarding operations of a DUT throughout the entire range of input data rates and frame sizes. The primary idea is to see what would be the frame loss under the overload conditions. It may be that the overloaded forwarding engine would forward less traffic than in the situation close to the overload. Throughput may drop below the possible maximum. As per section 26.3 of [RFC2544], it is RECOMMENDED to have the data for all tested frame sizes with a 10% load step above the wire-speed throughput measured in Section 6.1. It is RECOMMENDED to test all three (for SR-MPLS) or four (for SRv6) test types discussed in Section 6.1.

Procedure: Similar to Section 6.1.

Reporting Format: A table with all parameters specified in Section 5.

6.5. System Recovery

Objective: To characterize the speed at which a DUT recovers from an overload condition for each of the SR forwarding operations. It is RECOMMENDED to test all three (for SR-MPLS) or four (for SRv6) test types discussed in Section 6.1.

Procedure: Similar to section 6.4 of [RFC5695] or section 26.5 of [RFC2544]. Send a stream of frames at a rate of 110% of the recorded throughput rate or the maximum rate for the media, whichever is lower, for at least 60 seconds. At Timestamp A reduce the frame rate to 50% of the above rate and record the time of the last frame lost (Timestamp B). The system recovery time is determined by subtracting Timestamp B from Timestamp A. The test SHOULD be repeated several times and the average of the recorded values being reported.

Reporting Format: A table with all parameters specified in Section 5.

6.6. Reset

Objective: To characterize the speed at which a DUT recovers from a hardware or software reset for each of the SR forwarding operations. According to section 1.3 of [RFC6201] it is possible to measure frame loss or time stamps (depending on the test tool capability). According to section 4 of [RFC6201] reset could be:

1) hardware,

2) software,

3) power interruption.

All resets may be partial, i.e. only for a particular part of hardware (line card) or software (module). Special interest may be to test redundant power supplies or routing engines to make sure that reset does not affect the traffic. Hardware reset may be soft (command for reset) or hard (physical removal and insertion of the module). These types of reset SHOULD be treated as different. It is OPTIONAL to test all three (for SR-MPLS) or four (for SRv6) test types discussed in Section 6.1, typically they would give the same result.

Procedure: It is inherited from [RFC6201] (see it for more details). It is simple in essence: create the traffic, initiate a reset, measure the time for the traffic lost.

Reporting Format: A table with all parameters specified in Section 5.

All type of reset tests are OPTIONAL.

6.7. Scaling

Objective: To check the scaling capabilities of a DUT as it is an Ingress PE where an SR Policy is configured.

Procedure:

1) Testing of the baseline. Configure a single SR Policy with just one CP and different Segment Lists with a number of SIDs as baseline (e.g. 3 SIDs); verify that the SR Policy is installed; prepare traffic flow and initiate it; then verify that traffic flows successfully. The expected result is that there is no packet loss.

2) Testing the SID scale per SL. The setup is the same as the previous test but SL is equal to Maximum SID Depth (MSD). All the other steps are same as the previous test.

3) Testing SL scale. The setup is the same as the previous test except that X Segment Lists are created, and the test is repeated for each value of X (e.g. X=10,15,20,etc). All SLs are in the same CP. On the first iteration each Segment List has a number of SIDs as baseline, each SL have the same weight (ECMP testing). On the second iteration the Segment List length is equal to MSD. The scope is to verify that traffic flows between CEs and ECMP is working. The test can be repeated with different weights per each SL, testing weighted ECMP (wECMP). The result is to find out the maximum supported SLs number and ECMP/wECMP works fine on that maximum SL scale, with no traffic drops.

4) Testing CP scale in one SR Policy. Y CPs are created in one SR Policy where Y=10,15,20,etc (each per different test run), set higher Discriminator for one of CPs. Verify that all CPs are configured but that only one is Active (with higher Discriminator). Use the minimal SL length, then the maximum SL length, according to the previous test. Verify that traffic flows, with no drops and correct ECMP/wECMP. The result is that the Active CP is working correctly with any SLs, ECMP/wECMP works as expected, and no traffic drops.

5) Testing SR Policies scale (can be combined with composite SR Policy testing as sub-case, if supported). Create Z SR Policies, where Z=10,15,20,etc, then apply CPs per each SR Policy, from one CP to the maximum tested amount of CPs from the previous test, create different color communities for steering traffic into those policies towards one or many Egress PEs (endpoints), start traffic flows per each SR Policy (matching all SLs/CP variances above), verify traffic flows, absence of drops, correct ECMP/wECMP. The maximum number of supported SR Policies (max Z). If composite SR Policy is supported combine all created SR Policies in one composite, then make verification.

Reporting Format: A table with all parameters specified in Section 5.

7. Security Considerations

Benchmarking methodologies are limited to technology characterization in a laboratory environment, with dedicated address space and constraints. Special capabilities SHOULD NOT exist in the DUT/SUT specifically for benchmarking purposes. Any implications for network security arising from the DUT/SUT SHOULD be identical in the lab and production networks. The benchmarking network topology is an

independent test setup and MUST NOT be connected to devices that may forward the test traffic into a production network or misroute traffic to the test management network.

There are no specific security considerations within the scope of this document.

8. IANA Considerations

This document has no IANA requests.

9. Acknowledgements

The authors would like to thank Al Morton, Gabor Lencse, Boris Khasanov, Gyan Mishra, Carsten Rossenhoevel, Maciek Konstantynowicz for the precious comments and suggestions.

10. References

10.1. Normative References

[RFC1242] Bradner, S., "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, DOI 10.17487/RFC1242, July 1991, <https://www.rfc-editor.org/info/rfc1242>.

[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>.

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <https://www.rfc-editor.org/info/rfc2544>.

[RFC3330] IANA, "Special-Use IPv4 Addresses", RFC 3330, DOI 10.17487/RFC3330, September 2002, <https://www.rfc-editor.org/info/rfc3330>.

[RFC4773] Huston, G., "Administration of the IANA Special Purpose IPv6 Address Block", RFC 4773, DOI 10.17487/RFC4773, December 2006, <https://www.rfc-editor.org/info/rfc4773>.

[RFC4814] Newman, D. and T. Player, "Hash and Stuffing: Overlooked Factors in Network Device Benchmarking", RFC 4814, DOI 10.17487/RFC4814, March 2007, <https://www.rfc-editor.org/info/rfc4814>.

[RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, DOI 10.17487/RFC5180, May 2008, <https://www.rfc-editor.org/info/rfc5180>.

[RFC5695] Akhter, A., Asati, R., and C. Pignataro, "MPLS Forwarding Benchmarking Methodology for IP Flows", RFC 5695, DOI 10.17487/RFC5695, November 2009, <https://www.rfc-editor.org/info/rfc5695>.

[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>.

[RFC8219] Georgescu, M., Pislaru, L., and G. Lencse, "Benchmarking Methodology for IPv6 Transition Technologies", RFC 8219, DOI 10.17487/RFC8219, August 2017, <https://www.rfc-editor.org/info/rfc8219>.

[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>.

[RFC8660] Bashandy, A., Ed., Filsfils, C., Ed., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with the MPLS Data Plane", RFC 8660, DOI 10.17487/RFC8660, December 2019, <https://www.rfc-editor.org/info/rfc8660>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

10.2. Informative References

[ETSI-GR-NFV-TST-007]
ETSI, "ETSI GR NFV-TST 007: Network Functions Virtualisation (NFV) Release 3; Testing; Guidelines on Interoperability Testing for MANO", 2020, <https://www.etsi.org/deliver/etsi_gr/NFV-TST/001_099/007/03.01.01_60/gr_NFV-TST007v030101p.pdf>.

[I-D.ietf-idr-bgp-sr-segtypes-ext]
Talaulikar, K., Filsfils, C., Previdi, S., Mattes, P., and D. Jain, "Segment Routing Segment Types Extensions for BGP SR Policy", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-sr-segtypes-ext-08, 20 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-bgp-sr-segtypes-ext-08>.

[I-D.ietf-idr-sr-policy-safi]
Previdi, S., Filsfils, C., Talaulikar, K., Mattes, P., and D. Jain, "Advertising Segment Routing Policies in BGP", Work in Progress, Internet-Draft, draft-ietf-idr-sr-policy-safi-13, 6 February 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-sr-policy-safi-13>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <https://www.rfc-editor.org/info/rfc3032>.

[RFC6201] Asati, R., Pignataro, C., Calabria, F., and C. Olvera, "Device Reset Characterization", RFC 6201, DOI 10.17487/RFC6201, March 2011, <https://www.rfc-editor.org/info/rfc6201>.

[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>.

[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>.

[RFC8664] Sivabalan, S., Filsfils, C., Tantsura, J., Henderickx, W., and J. Hardwick, "Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing", RFC 8664, DOI 10.17487/RFC8664, December 2019, <https://www.rfc-editor.org/info/rfc8664>.

[RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <https://www.rfc-editor.org/info/rfc8665>.

[RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <https://www.rfc-editor.org/info/rfc8666>.

[RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <https://www.rfc-editor.org/info/rfc8667>.

[RFC8669] Previdi, S., Filsfils, C., Lindem, A., Ed., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix Segment Identifier Extensions for BGP", RFC 8669, DOI 10.17487/RFC8669, December 2019, <https://www.rfc-editor.org/info/rfc8669>.

[RFC9004] Morton, A., "Updates for the Back-to-Back Frame Benchmark in RFC 2544", RFC 9004, DOI 10.17487/RFC9004, May 2021, <https://www.rfc-editor.org/info/rfc9004>.

[RFC9256] Filsfils, C., Talaulikar, K., Ed., Voyer, D., Bogdanov, A., and P. Mattes, "Segment Routing Policy Architecture", RFC 9256, DOI 10.17487/RFC9256, July 2022, <https://www.rfc-editor.org/info/rfc9256>.

[RFC9352] Psenak, P., Ed., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane", RFC 9352, DOI 10.17487/RFC9352, February 2023, <https://www.rfc-editor.org/info/rfc9352>.

[RFC9513] Li, Z., Hu, Z., Talaulikar, K., Ed., and P. Psenak, "OSPFv3 Extensions for Segment Routing over IPv6 (SRv6)", RFC 9513, DOI 10.17487/RFC9513, December 2023, <https://www.rfc-editor.org/info/rfc9513>.

[RFC9514] Dawra, G., Filsfils, C., Talaulikar, K., Ed., Chen, M., Bernier, D., and B. Decraene, "Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing over IPv6 (SRv6)", RFC 9514, DOI 10.17487/RFC9514, December 2023, <https://www.rfc-editor.org/info/rfc9514>.

[RFC9800] Cheng, W., Ed., Filsfils, C., Li, Z., Decraene, B., and F. Clad, Ed., "Compressed SRv6 Segment List Encoding", RFC 9800, DOI 10.17487/RFC9800, June 2025, <https://www.rfc-editor.org/info/rfc9800>.

Authors' Addresses

Giuseppe Fioccola
Huawei Technologies
Viale Martesana, 12
20055 Vimodrone (Milan)
Italy
Email: giuseppe.fioccola@huawei.com

Eduard Vasilenko
Huawei Technologies
17/4 Krylatskaya str.
Moscow
Email: vasilenko.eduard@huawei.com

Paolo Volpato
Huawei Technologies
Viale Martesana, 12
20055 Vimodrone (Milan)
Italy
Email: paolo.volpato@huawei.com

Luis Miguel Contreras Murillo
Telefonica
Spain
Email: luismiguel.contrerasmurillo@telefonica.com

Bruno Decraene
Orange
France
Email: bruno.decraene@orange.com