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

23 October 2023

Benchmarking Methodology for MPLS Segment Routing
draft-vfv-bmwg-srmpls-bench-meth-08

Abstract

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

Status of This Memo

Copyright Notice

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] and [RFC5695] to SR network.

The SR architecture can be instantiated on two data-plane: SR over MPLS (SR-MPLS) and SR over IPv6 (SRv6). This document is limited to SR-MPLS.

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.

The MPLS label stack in scope of this document has a minimum of two entries, e.g. two SIDs. But 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 is expected that future documents may cover the benchmarking of SR-MPLS applications such as Layer 3 VPN (L3VPN) [RFC4364], EVPN [RFC7432], Fast ReRoute [I-D.ietf-rtgwg-segment-routing-ti-lfa], etc.

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

[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.

The purpose of this document is to describe a methodology specific to the benchmarking of Segment Routing. The methodology described is a complement for [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. Hence SR-MPLS Segment List typically contains more labels. The classical forwarding functions available for MPLS networks allow implementing the SR operations.

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

The PUSH operation corresponds to the Label Push function, according to the MPLS label pushing rules specified in [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 before and/or after the popping depends on the instruction associated with the active SID on the received packet prior to the popping. It is equivalent to Penultimate Hop Popping (PHP).

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. It is equivalent to Ultimate Hop Popping (UHP).

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 an IGP signaling extension to advertise their own prefix SIDs. 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 in order 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 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-segment-routing-te-policy], or via manual configuration on the router. PCEP and BGP signaling of SR Policies can be the case of a controller-based deployment.

3. Test Methodology

3.1. 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-MPLS nodes. It is OPTIONAL to choose non-equal proportion for upstream and downstream traffic for some specific aggregation nodes.

The RECOMMENDED topology for SR-MPLS 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.

```
                     +----------+
           +---------|          |<---------+
           | +-------|  Tester  |<-------+ |
           | | +-----|          |<-----+ | |
           | | |     +----------+      | | |
           | | |                       | | |
           | | |      +--------+       | | |
           | | +----->|        |-------+ | |
           | +------->|  DUT   |---------+ |
           +--------->|        |-----------+
                      +--------+
```

Figure 1: Test environment for SR-MPLS 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.

Interface numbers involved in the tests and their oversubscription ratio MUST be reported. SIDs represent only prefix and adjacency segments. In general, MPLS labels at the bottom of the stack may be used to encode services (L2/L3 VPNs) but it is out of the scope of this document.

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.

3.2. Label Distribution Support

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: the BGP peer segment and the BGP Prefix segment.

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

* IS-IS Extensions for Segment Routing [RFC8667]

* OSPF Extensions for Segment Routing [RFC8665]

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

* Segment Routing Policy Architecture [RFC9256].

A routing protocol (OSPF or ISIS or BGP) SHOULD be used for the construction of the simplest stack of 1 SID. It is RECOMMENDED that SR policy should be used for the construction of a stack with 2 SIDs. It is possible to test longer SID lists if there is an interest.

It is RECOMMENDED that the top SID on the list (outer label) should be an adjacency type to emulate the traffic engineering scenario. In all cases, SID stack configuration SHOULD happen before packet forwarding would be started. Control plane convergence speed is not the subject of the present tests.

The label distribution method and SR policy construction method used MUST be reported according to Section 4.

3.3. Frame Formats and Sizes

The tests for SR-MPLS will use Frame characteristics similarly to section 4.1.5 of [RFC5695], except the need for a bigger MTU to accommodate many MPLS labels.

It is to be noted that [RFC5695] requires exactly a single entry in the MPLS label stack. For the scope of this document, this is not enough to simulate a typical SR-MPLS SID list. MPLS label values used in any test case MUST be outside the reserved label value (0-15). The number of entries in the label stack 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.

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 1-2% error is expected. The most popular layer 2 technology for SR is Ethernet, it does not have stuffing.

Section 4.1.5 of [RFC5695] observes that the presence of an MPLS label has the effect of increasing the maximum frame payload size [RFC3032] so that "the resulting Layer 2 frame is Z octets more than the conventional maximum frame payload size, where Z = 4 x number of entries in the label stack".

As already stated, the SID list in scope of this document is composed of two SIDs. Accordingly, it is RECOMMENDED to set the media MTU value to the effective maximum frame payload size [RFC3032], which equals 2 * Z octets + conventional maximum frame payload size. It is expected that such a change in the media MTU value only impacts the effective Maximum Frame Payload Size for MPLS packets, but not for IP packets. The depth of the label stack is set to Z = 4 x 2 = 8 octets.

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

```
<-------------------------72-1526B------------------------->
<---18B---><--4B--><--4B--><-----------46-1500-------------->
+---------+-------+-------+---------+----------------------+
|         | MPLS  | MPLS  |         |         |            |
| Layer 2 | Label | Label | Layer 3 | Layer 4 | High layers |
+---------+-------+-------+---------+----------------------+
```

Figure 2: Ethernet Frame Structure

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).

RECOMMENDED frame sizes are the following:

* Ethernet Minimal: 64+n*4 (n=2)

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

* Ethernet Typical: 1518+n*4 (n=2)

* DUT Maximum: 9000 (or any claimed maximum)

where n is the number of labels (SID Depth).

Note that n*4 octets are added in the previous calculations to accommodate MPLS labels needed for respective tests. The typical frame size values are listed above for the DUT minimal wire speed and maximum, but 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.

3.4. Protocol Addresses

IANA reserved an IPv6 address block 2001:0002::/48 ([RFC4773]) for use with IPv6 benchmark testing and block 198.18.0.0/15 ([RFC3330]) for IPv4 benchmark testing. 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.

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 to have many flows with address randomization for acceptable hash-based load balancing that is implemented in all forwarding engines. 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.

3.5. 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.

In general, the test portion of each trial SHOULD be no less than 250 seconds, which is a reasonable value based on common hold time values. But a test can also adapt to the real setup and select a different value if default configuration has been changed. The test portion of each trial can be chosen at least 10 seconds longer than the hold time to verify that the DUT can maintain a stable control plane when the data-forwarding plane is under stress.

3.6. Traffic Verification

Traffic verification is following section 10 of [RFC2544] and section 4.1.8 of [RFC5695].

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 duplicate 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 SR-MPLS tests assumes headers modifications. All packets SHOULD be checked of the correct headers 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 "to verify that the packets received by the test tool carry the expected MPLS label".

3.7. 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 test. 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-MPLS.

4. Reporting Format

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

Reporting parameter preserved from [RFC5695]:

* Throughput in bytes per second and frames per second

* Frame sizes in Octets (see Section 3.3)

* 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-MPLS Forwarding Operations (PUSH/ NEXT/ CONTINUE).

* Label Distribution protocol and IGP are the same in the context of SR-MPLS. Hence, it is called "label distribution".

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 MPLS Label Stack and the type of SIDs used (Global/Local).

* 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 buffers 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.

5. SR-MPLS Forwarding Benchmarking Tests

In general, tests are compliant with [RFC2544] but the important correction discussed in section 6 of [RFC5695] 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. But 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 on figure 3. Then results correlation and recalculation of the real performance would be an additional burden.

```
                 +----------+
         +-------|  Tester1 |<-------+
         | +-----|          |<-----+ |
         | |     +----------+      | |
         | |                       | |
         | |      +--------+       | |
         | +----->|        |-------+ |
         +------->|  DUT   |---------+
         +------->|        |---------+
         | +----->|        |-------+ |
         | |      +--------+       | |
         | |                       | |
         | |     +----------+      | |
         | +-----|  Tester2 |<-----+ |
         +-------|          |<-------+
                 +----------+
```

Figure 3: 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 (No-Drop Rate - NDR). 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 in order to measure the resulting maximum number. It can also be tested 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.

5.1. Throughput

This section contains a description of the tests that are related to the characterization of a DUT's SR-MPLS 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. It is separately discussed only for throughput tests as an example.

5.1.1. Throughput for SR-MPLS PUSH

Objective: To obtain the DUT's Throughput during the PUSH forwarding operation. 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].

Procedure: Similar to [RFC5695] extended to test SID list longer than 1 SID (more than 2 are RECOMMENDED). 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 SIDs on respective sides, as per Section 3.4, and must use one option for SID stack construction, as per Section 3.2, on its receive and transmit interfaces towards the DUT.

Reporting Format: A table with all parameters specified in Section 4.

5.1.2. Throughput for SR-MPLS NEXT

Objective: To obtain the DUT's Throughput during the NEXT forwarding operation. 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].

Procedure: Similar to [RFC5695] extended to test SID list longer than 1 SID (more than 2 are RECOMMENDED). 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 SIDs on respective sides, as per Section 3.4, and must use one option for SID stack construction, as per Section 3.2, on its receive and transmit interfaces towards the DUT.

Reporting Format: A table with all parameters specified in Section 4.

5.1.3. Throughput for SR-MPLS CONTINUE

Objective: To obtain the DUT's Throughput during the CONTINUE forwarding operation. 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.

Procedure: Similar to [RFC5695] extended to test SID list longer than 1 SID (more than 2 are RECOMMENDED). 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 SIDs on respective sides, as per Section 3.4, and must use one option for SID stack construction, as per Section 3.2, on its receive and transmit interfaces towards the DUT.

Reporting Format: A table with all parameters specified in Section 4.

5.2. Buffers 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 3.7.

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

Procedure: Should be inherited from [RFC9004] with 2 SIDs RECOMMENDED (many SIDs are possible). Despite the simple general idea for filling the buffer until 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 4.

5.3. 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-MPLS forwarding operations (PUSH, NEXT, CONTINUE). It is RECOMMENDED to test all three test types discussed in Section 5.1.

Procedure: Similar to Section 5.1. 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 4.

5.4. 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-MPLS 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 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 10% load step above the wire-speed throughput measured in Section 5.1. It is RECOMMENDED to test all three test types discussed in Section 5.1.

Procedure: Similar to Section 5.1.

Reporting Format: A table with all parameters specified in Section 4.

5.5. System Recovery

Objective: To characterize the speed at which a DUT recovers from an overload condition for each of the SR-MPLS forwarding operations. It is RECOMMENDED to test all three test types discussed in Section 5.1.

Procedure: Similar to section 6.4 of [RFC5695] or section 26.5 of [RFC2544]. Send a stream of frames at a rate 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 a number of times and the average of the recorded values being reported.

Reporting Format: A table with all parameters specified in Section 4.

5.6. Reset

All type of reset tests are OPTIONAL.

Objective: To characterize the speed at which a DUT recovers from a device or software reset for each of the SR-MPLS 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, or 3) power interruption. All resets may be partial, i.e. only for a particular part of hardware (line card) or software (module). Especial 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 test types discussed in Section 5.1, 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 4.

6. 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.

7. IANA Considerations

This document has no IANA actions.

8. Acknowledgements

The authors would like to thank Al Morton, Gabor Lencse, Boris Khasanov for the precious comments and suggestions.

9. References

9.1. 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>.

[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>.

9.2. 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-segment-routing-te-policy]
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-segment-routing-te-policy-25, 26 September 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-idr-segment-routing-te-policy-25>.

[I-D.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", Work in Progress, Internet-Draft, draft-ietf-rtgwg-segment-routing-ti-lfa-11, 30 June 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-segment-routing-ti-lfa-11>.

[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>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[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>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <https://www.rfc-editor.org/info/rfc7432>.

[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>.

[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>.

Authors' Addresses

Giuseppe Fioccola
Huawei Technologies
Palazzo Verrocchio, Centro Direzionale Milano 2
20054 Segrate (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
Palazzo Verrocchio, Centro Direzionale Milano 2
20054 Segrate (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