SFC WG
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2018

T. Ao
ZTE Corporation
G. Mirsky
ZTE Corp.
Z. Chen
China Telecom

October 27, 2017

SFC OAM for path consistency
draft-ao-sfc-oam-path-consistency-01

Abstract

Service Function Chain(SFC) defines an ordered set of service functions(SFs) to be applied to packets and/or frames and/or flows selected as a result of classification. SFC Operation, Administration and Maintenance can monitor the continuity of the SFC, i.e., that all elements of the SFC are reachable to each other in the downstream direction. But SFC OAM must support verification that the order of traversing these SFs corresponds to the state defined by the SFC control plane or orchestrator, the metric referred in this document as the path consistency of the SFC. This document defines a new SFC OAM method to support SFC consistency, i.e. verification that all elements of the given SFC are being traversed in the expected order.

Table of Contents

1. Introduction

Service Function Chain (SFC) is a chain with a series of ordered Service Functions(SFs). Service Function Path (SFP) is a path of a SFC. SFC is described in detail in the SFC architecture document [RFC7665]. The SFs in the SFC are ordered and only when traffic is processed by one SF then it should be processed by the next SF, otherwise errors may occur. Sometimes, a SF needs to use the metadata from its upstream SF process. That's why it's very important for the operator to make sure that the order of traversing the SFs is exactly as defined by the control plane or the

orchestrator. This document refers to the correspondence between the state of control plane and the SFP itself as the SFP consistency.

This document defines the method to check the path consistency of the SFP. It is an extension of the SFC Echo-request/Echo-reply specified in the [I-D.wang-sfc-multi-layer-oam].

2. Conventions used in this document

2.1. Terminology

SFC(Service Function Chain): An ordered set of some abstract SFs.

SFF: Service Function Forwarder

SF: Service Function

OAM: Operation, Administration and Maintenance

SFP: Service Function Path

COAM(Consistency OAM): OAM that can be used to check path consistency.

2.2. Requirements Language

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

3. Consistency OAM: Theory of Operation

Consistency OAM uses two functions: COAM Request and COAM Reply. The SFF, that is ingress of the SFP, transmits COAM Request packet. Every intermediate SFF that receives the COAM Request MUST perform the following actions:

- collect information of traversed by the COAM Request packet SFs and send it to the ingress SFF as COAM Reply packet over IP network [I-D.wang-sfc-multi-layer-oam];

- forward the COAM Request to next downstream SFF if the one exists.

As result, the ingress SFF collects information about all traversed SFFs and SFs, information of the actual path the COAM packet has traveled, so that we can verify the path consistency of the SFC. The

mechanism for the SFP consistency verification is outside the scope of this document.

3.1. COAM packet

Consistency OAM introduces two new types of messages to the SFC Echo request/reply operation [I-D.wang-sfc-multi-layer-oam] with the following values Section 5.1:

- TBA1 - COAM Request
- TBA2 - COAM Reply

An SFF, upon receiving the Consistency OAM Request, MUST include the corresponding SFs information, Section 3.2, into the Value field of the COAM Reply packet.

The COAM packet is displayed in Figure 1.

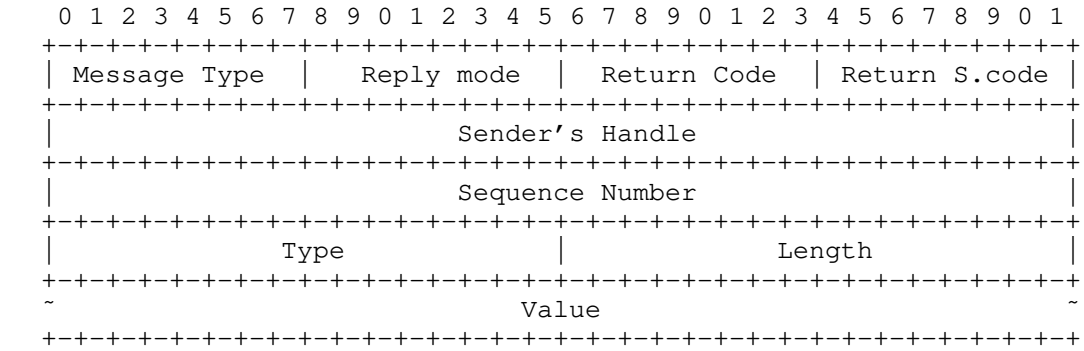


Figure 1: COAM Packet Header

3.2. SF Sub-TLV

Every SFF receiving COAM Request packet MUST include the SF characteristic data into the COAM Reply packet. The per SF data included in COAM Reply packet as SF Information sub-TLV that is displayed in Figure 2.

After the COAM traversed the SFP, all the information of the SFs on the SFP are collected in the TLVs with COAM Reply.

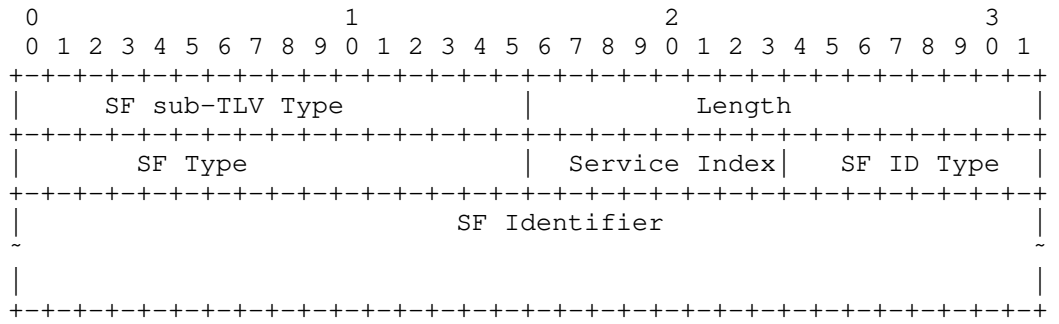


Figure 2: Service Function sub-TLV

SF sub-TLV Type: is two octets long field. It indicates that the TLV is a SF TLV which contains the information of one SF.

Length: is two octets long field. The value of the field is the length of the data following the Length field counted in octets.

SF Type: is two octets long field. It is defined in [I-D.ietf-bess-nsh-bgp-control-plane] and indicates the type of SF, e.g., Firewall, Deep Packet Inspection, WAN optimization controller, etc.

Service Index: indicates the SF's position on the SFP.

SF ID Type: is one octet long field with values defined as Section 5.4.

SF Identifier: An identifier of the SF. The length of the SF Identifier depends on the type of the SF ID Type. For example, if the SF Identifier is its IPv4 address, the SF Identifier should be 32 bits.

4. Security Considerations

Security considerations discussed in [I-D.ietf-sfc-nsh] apply to this document.

In addition, since Service Function sub-TLV discloses information about the RSP the spoofed COAM Request packet may be used to obtain network information, it is RECOMMENDED that implementations provide a means of checking the source addresses of COAM Request messages, specified in SFC Source TLV [I-D.wang-sfc-multi-layer-oam], against an access list before accepting the message.

5. IANA Considerations

5.1. COAM Message Types

IANA is requested to assign values from its Message Types sub-registry in SFC Echo Request/Echo Reply Message Types registry as follows:

Value	Description	Reference
TBA1	SFP Consistency Echo Request	This document
TBA2	SFP Consistency Echo Reply	This document

Table 1: SFP Consistency Echo Request/Echo Reply Message Types

5.2. SFF Information Record TLV Type

IANA is requested to assign new type value from SFC OAM TLV Type registry as follows:

Value	Description	Reference
TBA3	SFF Information Record Type	This document

Table 2: SFF-Information Record

5.3. SF Information Sub-TLV Type

IANA is requested to assign new type value from SFC OAM TLV Type registry as follows:

Value	Description	Reference
TBA4	SF Information	This document

Table 3: SF-Information Sub-TLV Type

5.4. SF Identifier Types

IANA is requested create in the registry SF Types the new sub-registry SF Identifier Types. All code points in the range 1 through 191 in this registry shall be allocated according to the "IETF

Review" procedure as specified in [RFC8126] and assign values as follows:

Value	Description	Reference
0	Reserved	This document
TBA6	IPv4	This document
TBA7	IPv6	This document
TBA8	MAC	This document
TBA8+1-191	Unassigned	IETF Review
192-251	Unassigned	First Come First Served
252-254	Unassigned	Private Use
255	Reserved	This document

Table 4: SF Identifier Type

6. Acknowledgements

Thanks to John Drake for his review and the reference to the work on BGP Control Plane for NSH SFC.

7. References

7.1. Normative References

[I-D.ietf-bess-nsh-bgp-control-plane]
Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for NSH SFC", draft-ietf-bess-nsh-bgp-control-plane-01 (work in progress), September 2017.

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-27 (work in progress), October 2017.

[I-D.wang-sfc-multi-layer-oam]
Mirsky, G., Meng, W., Khasnabish, B., and C. Wang, "Multi-Layer Active OAM for Service Function Chains in Networks", draft-wang-sfc-multi-layer-oam-10 (work in progress), September 2017.

[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>.

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

[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>.

7.2. Informational References

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Ting Ao
ZTE Corporation
No.889, BiBo Road
Shanghai 201203
China

Phone: +86 21 68897642
Email: ao.ting@zte.com.cn

Greg Mirsky
ZTE Corp.
1900 McCarthy Blvd. #205
Milpitas, CA 95035
USA

Email: gregimirsky@gmail.com

Zhonghua Chen
China Telecom
No.1835, South PuDong Road
Shanghai 201203
China

Phone: +86 18918588897
Email: 18918588897@189.cn

SFC WG
Internet-Draft
Intended status: Standards Track
Expires: June 30, 2019

T. Ao
ZTE Corporation
G. Mirsky
ZTE Corp.
Z. Chen
China Telecom
K. Leung
Cisco

Dec 27, 2018

SFC OAM for path consistency
draft-ao-sfc-oam-path-consistency-04

Abstract

Service Function Chain (SFC) defines an ordered set of service functions (SFs) to be applied to packets and/or frames and/or flows selected as a result of classification. SFC Operation, Administration and Maintenance can monitor the continuity of the SFC, i.e., that all elements of the SFC are reachable to each other in the downstream direction. But SFC OAM must support verification that the order of traversing these SFs corresponds to the state defined by the SFC control plane or orchestrator, the metric referred in this document as the path consistency of the SFC. This document defines a new SFC OAM method to support SFC consistency check, i.e. verification that all elements of the given SFC are being traversed in the expected order.

Table of Contents

1. Introduction

Service Function Chain (SFC) is a chain with a series of ordered Service Functions (SFs). Service Function Path (SFP) is a path of a SFC. SFC is described in detail in the SFC architecture document [RFC7665]. The SFs in the SFC are ordered and only when one SF processes traffic then it can be processed by the next SF. Otherwise errors may occur. Sometimes, a SF needs to use the metadata from its

upstream SF process. That's why it's very important for the operator to make sure that the order of traversing the SFs is exactly as defined by the control plane or the orchestrator. This document refers to the correspondence between the state of the control plane and the SFP itself as the SFP consistency.

This document defines the method to check the path consistency of the SFP. It is an extension of the SFC Echo-request/Echo-reply specified in the [I-D.ietf-sfc-multi-layer-oam].

2. Conventions used in this document

2.1. Terminology

SFC(Service Function Chain): An ordered set of some abstract SFs.

SFF: Service Function Forwarder

SF: Service Function

OAM: Operation, Administration and Maintenance

SFP: Service Function Path

COAM(Consistency OAM): OAM that can be used to check the consistency of the Service Function Path.

2.2. Requirements Language

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

3. Consistency OAM: Theory of Operation

Consistency OAM uses two functions: COAM Request and COAM Reply. The SFF, that is ingress of the SFP, transmits COAM Request packet. Every intermediate SFF that receives the COAM Request MUST perform the following actions:

- Collect information of traversed by the COAM Request packet SFs and send it to the ingress SFF as COAM Reply packet over IP network [I-D.ietf-sfc-multi-layer-oam];

- Forward the COAM Request to next downstream SFF if the one exists.

As result, the ingress SFF collects information about all traversed SFFs and SFs, information of the actual path the COAM packet has traveled, so that we can verify the path consistency of the SFC. The mechanism for the SFP consistency verification is outside the scope of this document.

3.1. COAM packet

Consistency OAM introduces two new types of messages to the SFC Echo request/reply operation [I-D.ietf-sfc-multi-layer-oam] with the following values Section 5.1:

- TBA1 - COAM Request
- TBA2 - COAM Reply

Upon receiving the Consistency OAM(COAM) Request, the SFF MUST respond with the COAM Reply. The SFF MUST include the SFs information, as as described in Section 3.3 and Section 3.2.

The COAM packet is displayed in Figure 1.

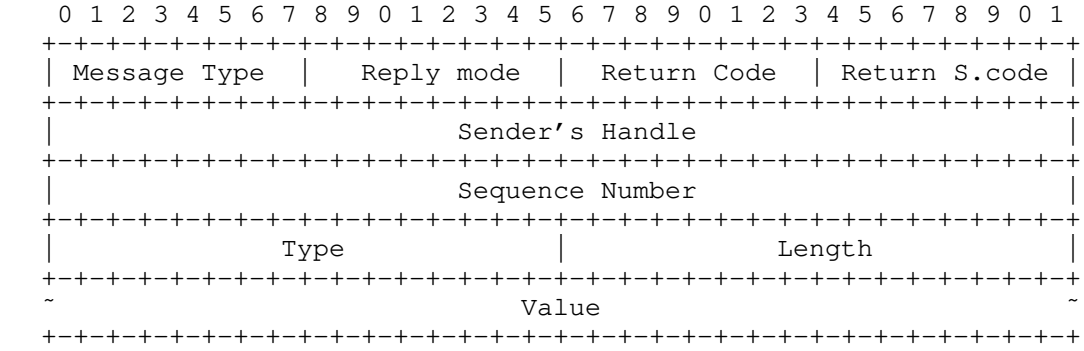


Figure 1: COAM Packet Header

3.2. SFF Information Record TLV

For COAM Request, the SFF MUST include the Information of SFs into the SF Information Record TLV in the COAM Reply message. Every SFF send back one COAM Reply Message with all the SFs that are attaching to the SFF along the SFP indicated by the COAM Request.

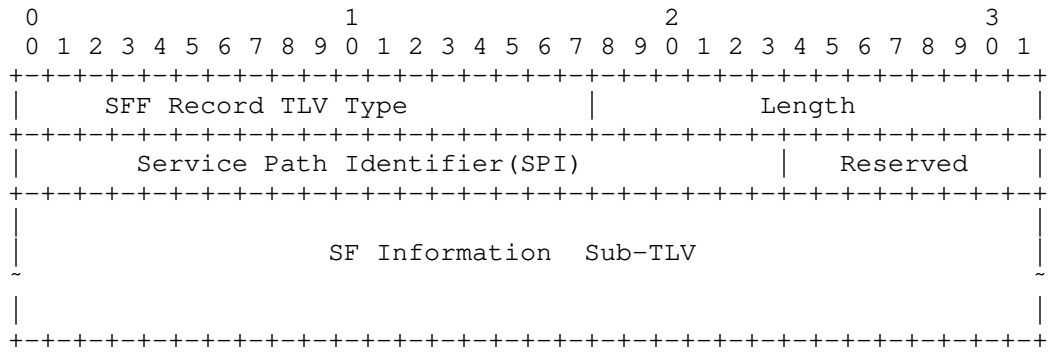


Figure 2: SFF Information Record TLV

Service Path Identifier(SPI): The identifier of SFP to which all the SFs in this TLV belong.

SF Information Sub-TLV: The Sub-TLV as defined in Figure 3.

3.3. SF Information Sub-TLV

Every SFF receiving COAM Request packet MUST include the SF characteristic data into the COAM Reply packet. The per SF data included in COAM Reply packet as SF Information sub-TLV that is displayed in Figure 3.

After the COAM traversed the SFP, all the information of the SFs on the SFP are collected from the TLVs with COAM Reply.

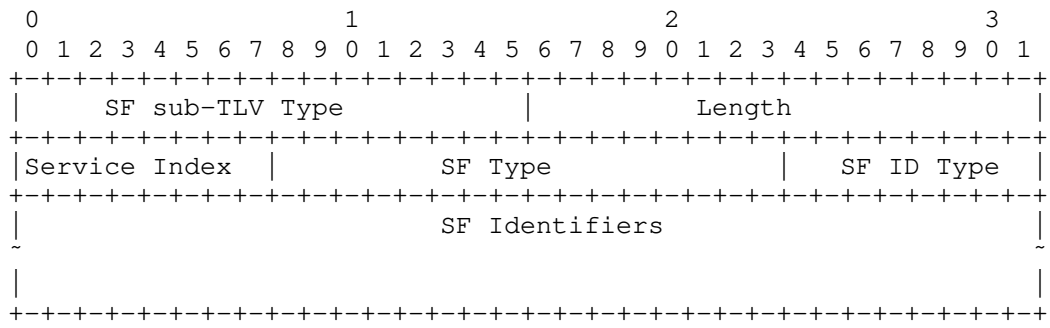


Figure 3: Service Function information sub-TLV

SF sub-TLV Type: Two octets long field. It indicates that the TLV is a SF TLV which contains the information of one SF.

Length: Two octets long field. The value of the field is the length of the data following the Length field counted in octets.

Service Index: Indicates the SF's position on the SFP.

SF Type: Two octets long field. It is defined in [I-D.ietf-bess-nsh-bgp-control-plane] and indicates the type of SF, e.g., Firewall, Deep Packet Inspection, WAN optimization controller, etc.

Reserved: For future use. MUST be zeroed on transmission and MUST be ignored on receipt.

SF ID Type: One octet long field with values defined as Section 5.4.

SF Identifier: An identifier of the SF. The length of the SF Identifier depends on the type of the SF ID Type. For example, if the SF Identifier is its IPv4 address, the SF Identifier should be 32 bits. SF ID Type and SF Identifier may be a list, indicating the list of the SFs are in load balance group.

3.4. SF Information Sub-TLV Construction

Each SFF in the SFP MUST send one and only one COAM Reply corresponding to the COAM Request. If there is only one SF attached to the SFF in such SFP, only one SF information sub-TLV is included in the on COAM Reply. If there are several SFs attached to the SFF in the SFP, SF Information Sub-TLV MUST be constructed as described below in either Section 3.4.1 and Section 3.4.2.

3.4.1. Multiple SFs as hops of SFP

Multiple SFs attached to one SFF are the hops of the SFP, the service indexes of these SFs are different. Service function types of these SFs could be different or be the same. Information about all SFs MAY be included in the COAM Reply message. Information about each SF MUST be listed as separate SF Information Sub-TLVs in the COAM Reply message.

An example of the COAM procedure for this case is shown in Figure 4. The Service Function Path(SPI=x) is SF1->SF2->SF4->SF3. The SF1,SF2 and SF3 are attached to SFF1, and SF4 is attached to SFF2. The COAM Request message is sent to the SFFs in the sequence of the SFP(SFF1->SFF2->SFF1). Every SFF(SFF1,SFF2) replies with the information of SFs belonging to the SFP. The SF information Sub-TLV in Figure 3 contains information for each SF(SF1, SF2,SF3 and SF4).

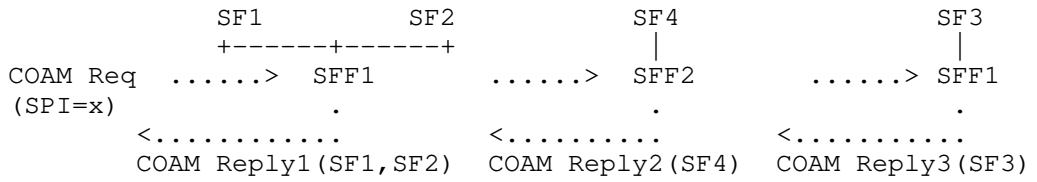


Figure 4: Example 1 for COAM Reply with multiple SFs

3.4.2. Multiple SFs for load balance

Multiple SFs may be attached to one SFF to balance the load, in other words, that means that the particular traffic flow will transmit only one of these SFs . These SFs have the same Service Function Type and Service Index. For this case, the SF identifiers and SF ID Type of all these SFs will be listed in the SF Identifiers field and SF ID Type in a single SF information sub-TLV of COAM Reply message. The number of these SFs can be calculated according to SF ID Type and the value of Length field of the sub-TLV.

An example of the COAM procedure for this case is shown in Figure 4. The Service Function Path(SPI=x) is SF1a/SF1b->SF2a/SF2b. The Service Functions SF1a and SF1b are attached to SFF1 which are load balance for each other, and the Service Functions SF2a and SF2b are attached to SFF2 which are load balance for each other as well. The COAM Request message is sent to the SFFs in the sequence of the SFP (i.e. SFF1->SFF2). Every SFF(SFF1,SFF2) replies with the information of SFs belonging to the SFP. The SF information Sub-TLV in Figure 3 contains information for all SFs at that hop.

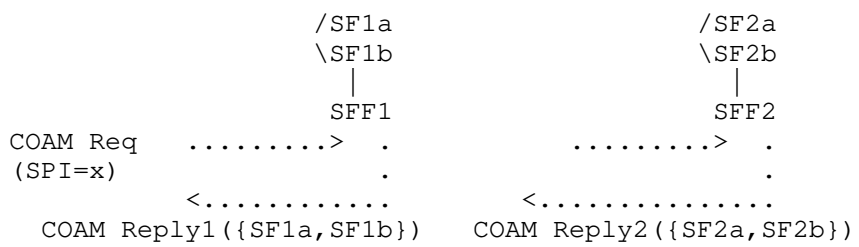


Figure 5: Example 2 for COAM Reply with multiple SFs

4. Security Considerations

Security considerations discussed in [RFC8300] apply to this document.

Also, since Service Function sub-TLV discloses information about the SFP the spoofed COAM Request packet may be used to obtain network information, it is RECOMMENDED that implementations provide a means of checking the source addresses of COAM Request messages, specified in SFC Source TLV [I-D.ietf-sfc-multi-layer-oam], against an access list before accepting the message.

5. IANA Considerations

5.1. COAM Message Types

IANA is requested to assign values from its Message Types sub-registry in SFC Echo Request/Echo Reply Message Types registry as follows:

Value	Description	Reference
TBA1	SFP Consistency Echo Request	This document
TBA2	SFP Consistency Echo Reply	This document

Table 1: SFP Consistency Echo Request/Echo Reply Message Types

5.2. SFF Information Record TLV Type

IANA is requested to assign new type value from SFC OAM TLV Type registry as follows:

Value	Description	Reference
TBA3	SFF Information Record Type	This document

Table 2: SFF-Information Record

5.3. SF Information Sub-TLV Type

IANA is requested to assign new type value from SFC OAM TLV Type registry as follows:

Value	Description	Reference
TBA4	SF Information	This document

Table 3: SF-Information Sub-TLV Type

5.4. SF Identifier Types

IANA is requested create in the registry SF Types the new sub-registry SF Identifier Types. All code points in the range 1 through 191 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126] and assign values as follows:

Value	Description	Reference
0	Reserved	This document
TBA6	IPv4	This document
TBA7	IPv6	This document
TBA8	MAC	This document
TBA8+1-191	Unassigned	IETF Review
192-251	Unassigned	First Come First Served
252-254	Unassigned	Private Use
255	Reserved	This document

Table 4: SF Identifier Type

6. Acknowledgements

Thanks to John Drake for his review and the reference to the work on BGP Control Plane for NSH SFC.

Thanks to Joel M. Halpern for their suggestion about the load balance scenario.

7. References

7.1. Normative References

[I-D.ietf-bess-nsh-bgp-control-plane]
Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for NSH SFC", draft-ietf-bess-nsh-bgp-control-plane-04 (work in progress), July 2018.

[I-D.ietf-sfc-multi-layer-oam]
Mirsky, G., Meng, W., Khasnabish, B., and C. Wang, "Active OAM for Service Function Chains in Networks", draft-ietf-sfc-multi-layer-oam-00 (work in progress), November 2018.

[I-D.ietf-sfc-nsh-tlv]
Quinn, P., Elzur, U., and S. Majee, "Network Service Header TLVs", draft-ietf-sfc-nsh-tlv-00 (work in progress), January 2018.

[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>.

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

[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>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

7.2. Informational References

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Ting Ao
ZTE Corporation
No.889, BiBo Road
Shanghai 201203
China

Phone: +86 21 68897642
Email: ao.ting@zte.com.cn

Greg Mirsky
ZTE Corp.
1900 McCarthy Blvd. #205
Milpitas, CA 95035
USA

Email: gregimirsky@gmail.com

Zhonghua Chen
China Telecom
No.1835, South PuDong Road
Shanghai 201203
China

Phone: +86 18918588897
Email: 18918588897@189.cn

Kent Leung
Cisco

Email: kleung@cisco.com

SFC WG
Internet-Draft
Intended status: Standards Track
Expires: April 30, 2018

T. Ao
ZTE Corporation
G. Mirsky
ZTE Corp.
Z. Chen
China Telecom

October 27, 2017

Controlled Return Path for Service Function Chain (SFC) OAM
draft-ao-sfc-oam-return-path-specified-01

Abstract

This document defines extensions to the Service Function Chain (SFC) Operation, Administration and Maintenance (OAM) that enable control of the Echo Reply return path by specifying it as Reverse Service Function Path. Enforcing the specific return path can be used to verify bidirectional connectivity of SFC and increase robustness of SFC OAM.

Table of Contents

1. Introduction

While Service Function Chain (SFC) Echo Request, defined in [I-D.wang-sfc-multi-layer-oam], always traverses the SFC it directed to, the corresponding Echo Reply is sent over IP network [I-D.wang-sfc-multi-layer-oam]. There are scenarios when it is beneficial to direct the responder to use path other than the IP network. This document defines extensions to the Service Function Chain (SFC) Operation, Administration and Maintenance (OAM) that enable control of the Echo Reply return path by specifying it as Reply Service Function Path. This document defines a new Type-Length-Value (TLV), Reply Service Function Path TLV, for Reply via Specified Path mode of SFC Echo Reply (Section 4).

The Reply Service Function Path TLV provides efficient mechanism to test bidirectional and hybrid SFCs, as these were defined in Section 2.2 [RFC7665], that allows an operator to test both directions of the bidirectional or hybrid SFP with a single SFC Echo Request/Echo Reply operation.

2. Conventions used in this document

2.1. Terminology

SF - Service Function

SFF - Service Function Forwarder

SFC - Service Function Chain, an ordered set of some abstract SFs.

SFP - Service Function Path

SPI - Service Path Index

OAM - Operation, Administration, and Maintenance

2.2. Requirements Language

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

3. Extension

Following reply modes had been defined in [I-D.wang-sfc-multi-layer-oam]:

- Do Not Reply
- Reply via an IPv4/IPv6 UDP Packet
- Reply via Application Level Control Channel
- Reply via Specified Path

The Reply via Specified Path mode is intended to enforce use of the particular return path specified in the included TLV. This mode may help to verify bidirectional continuity or increase robustness of the monitoring of the SFC by selecting more stable path. In case of SFC, the sender of Echo Request instructs the egress SFF to send Echo Reply message along the SFP specified in the SFC Reply Path TLV Section 4.

4. SFC Reply Path TLV

The SFC Reply Path TLV carries the information that sufficiently identifies the return SFP that the SFC Echo Reply message is expected to follow. The format of SFC Reply Path TLV is display in Figure 1.

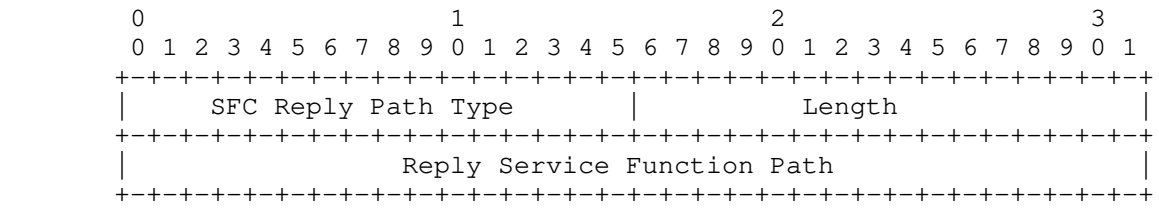


Figure 1: SFC Reply TLV Format

where:

- Reply Path TLV Type: is 2 octets long, indicates the TLV that contains a information about the SFC Reply path.
- Length: is 2 octets long, MUST be equal to 4
- Reply Service Function Path is used to describe the return path that an SFC Echo Reply is requested to follow.

The format of the Reply Service Function Path field displayed in Figure 2

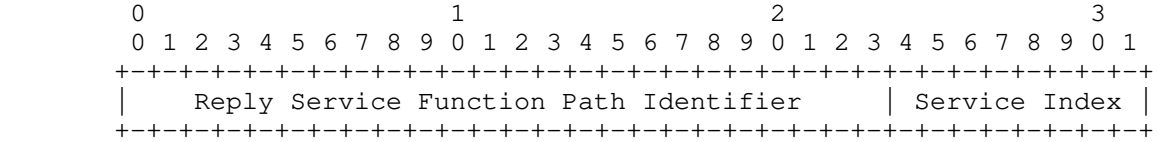


Figure 2: Reply Service Function Path Field Format

where:

- Reply Service Path Identifier: is SFP identifier for the path that the SFC Echo Reply message is requested to be sent over.
- Service Index: used for forwarding in the reply SFP.

5. Theory of Operation

[RFC7110] defined mechanism to control return path for MPLS LSP Echo Reply. In case of SFC, the return path is a SFP along which SFC Echo Reply message MUST be transmitted. Hence, the SFC Reply Path TLV included in the SFC Echo Request message MUST sufficiently identify the SFP that the sender of the Echo Request message expects the receiver to use for the corresponding SFC Echo Reply.

When sending an Echo Request the sender MUST set the value of Reply Mode field to "Reply via Specified Path", defined in [I-D.wang-sfc-multi-layer-oam], and MUST include SFC Reply Path TLV. The SFC Reply Path TLV includes identifier of the reverse SFP and an appropriate Service Index.

Echo Reply is expected to be sent by the egress SFF of the SFP being tested or by the SFF at which SFC TTL expires as defined [I-D.ietf-sfc-nsh]. Processing described below equally applies in both cases and referred as responding SFF.

If the Echo Request message with SFC Reply Path TLV, received by the responding SFF, has Reply Mode value of "Reply via Specified Path" but no SFC Reply Path TLV is present, then the responding SFF MUST send Echo Reply with Return Code set to "Reply Path TLV is missing" value (TBA2). If the responding SFF cannot find requested SFP it MUST send Echo Reply with Return Code set to "Reply SFP was not found" and include the SFC Reply Path TLV from the Echo Request message.

5.1. Case of Bi-directional SFC

Ability to specify the return path to be used for Echo Reply is very useful in bi-directional SFC. For bi-directional SFC, since the last SFF of the forward SFP may not co-locate with classifier of the reverse SFP,it is assumed that last SFF doesn’t know the reply path of a SFC. So even for bi-directional SFC, a reverse SFP also need to be indicated in reply path TLV in echo request message.

6. Security Considerations

Security considerations discussed in [I-D.ietf-sfc-nsh] apply to this document..

In addition, the SFC Return Path extension, defined in this document, may be used for potential "proxying" attacks. For example, an echo request initiator may specify a return path that has a destination different from that of the initiator. But normally, such attacks will not happen in an SFC domain where the initiators and receivers

belong to the same domain, as specified in [RFC7665]. Even if the attack happens, in order to prevent using the SFC Return Path extension for proxying any possible attacks, the return path SFP SHOULD have destination to the sender of the echo request, identified in SFC Source TLV [I-D.wang-sfc-multi-layer-oam]. The receiver may drop the echo request when it cannot determine whether the return path SFP has the destination to the initiator. That means, when sending echo request, the sender SHOULD choose a proper source address according the specified return path SFP to help the receiver to make the decision.

7. IANA Considerations

7.1. SFC Return Path Type

IANA is requested to assign from its SFC Echo Request/Echo Reply TLV registry new type as following:

Value	Description	Reference
TBA1	SFC Reply Path Type	This document

Table 1: SFC Return Path Type

7.2. New Return Codes

IANA is requested to assign new return codes from the SFC Echo Request/Echo Reply Return Codes registry as following:

Value	Description	Reference
TBA2	Reply Path TLV is missing	This document
TBA3	Reply SFP was not found	This document

Table 2: SFC Echo Reply Return Codes

8. References

8.1. Normative References

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-27 (work in progress), October 2017.

[I-D.wang-sfc-multi-layer-oam]
Mirsky, G., Meng, W., Khasnabish, B., and C. Wang, "Multi-Layer Active OAM for Service Function Chains in Networks", draft-wang-sfc-multi-layer-oam-10 (work in progress), September 2017.

[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>.

[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>.

8.2. Informative References

[RFC7110] Chen, M., Cao, W., Ning, S., Jounay, F., and S. Delord, "Return Path Specified Label Switched Path (LSP) Ping", RFC 7110, DOI 10.17487/RFC7110, January 2014, <https://www.rfc-editor.org/info/rfc7110>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Ting Ao
ZTE Corporation
No.889, BiBo Road
Shanghai 201203
China

Phone: +86 21 68897642
Email: ao.ting@zte.com.cn

Greg Mirsky
ZTE Corp.
1900 McCarthy Blvd. #205
Milpitas, CA 95035
USA

Email: gregimirsky@gmail.com

Zhonghua Chen
China Telecom
No.1835, South PuDong Road
Shanghai 201203
China

Phone: +86 18918588897
Email: 18918588897@189.cn

SFC WG
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2019

T. Ao
ZTE Corporation
G. Mirsky
ZTE Corp.
Z. Chen
China Telecom

October 16, 2018

Controlled Return Path for Service Function Chain (SFC) OAM
draft-ao-sfc-oam-return-path-specified-02

Abstract

This document defines extensions to the Service Function Chain (SFC) Operation, Administration and Maintenance (OAM) that enable control of the Echo Reply return path by specifying it as Reverse Service Function Path. Enforcing the specific return path can be used to verify bidirectional connectivity of SFC and increase the robustness of SFC OAM.

Table of Contents

1. Introduction

While Service Function Chain (SFC) Echo Request, defined in [I-D.wang-sfc-multi-layer-oam], always traverses the SFC it directed to, the corresponding Echo Reply is sent over IP network [I-D.wang-sfc-multi-layer-oam]. There are scenarios when it is beneficial to direct the responder to use a path other than the IP network. This document defines extensions to the Service Function Chain (SFC) Operation, Administration and Maintenance (OAM) that enable control of the Echo Reply return path by specifying it as Reply Service Function Path. This document defines a new Type-Length-Value (TLV), Reply Service Function Path TLV, for Reply via Specified Path mode of SFC Echo Reply (Section 4).

The Reply Service Function Path TLV can provide an efficient mechanism to test SFCs,such as bidirectional and hybrid SFC, as these were defined in Section 2.2 [RFC7665], For example, it allows an operator to test both directions of the bidirectional or hybrid SFP with a single SFC Echo Request/Echo Reply operation.

2. Conventions used in this document

2.1. Terminology

SF - Service Function

SFF - Service Function Forwarder

SFC - Service Function Chain, an ordered set of some abstract SFs.

SFP - Service Function Path

SPI - Service Path Index

OAM - Operation, Administration, and Maintenance

2.2. Requirements Language

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

3. Extension

Following reply modes had been defined in [I-D.wang-sfc-multi-layer-oam]:

- Do Not Reply
- Reply via an IPv4/IPv6 UDP Packet
- Reply via Application Level Control Channel
- Reply via Specified Path

The Reply via Specified Path mode is intended to enforce the use of the particular return path specified in the included TLV. This mode may help to verify bidirectional continuity or increase the robustness of the monitoring of the SFC by selecting a more stable path. In the case of SFC, the sender of Echo Request instructs the destination SFF to send Echo Reply message along the SFP specified in the SFC Reply Path TLV Section 4.

4. SFC Reply Path TLV

The SFC Reply Path TLV carries the information that sufficiently identifies the return SFP that the SFC Echo Reply message is expected to follow. The format of SFC Reply Path TLV is shown in Figure 1.

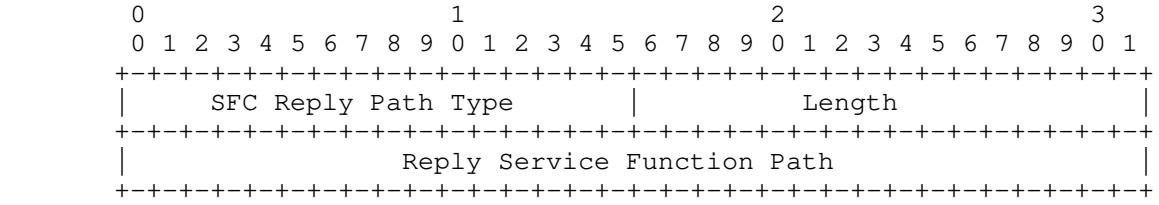


Figure 1: SFC Reply TLV Format

where:

- Reply Path TLV Type: is two octets long, indicates the TLV that contains information about the SFC Reply path.
- Length: is two octets long, MUST be equal to 4
- Reply Service Function Path is used to describe the return path that an SFC Echo Reply is requested to follow.

The format of the Reply Service Function Path field displayed in Figure 2

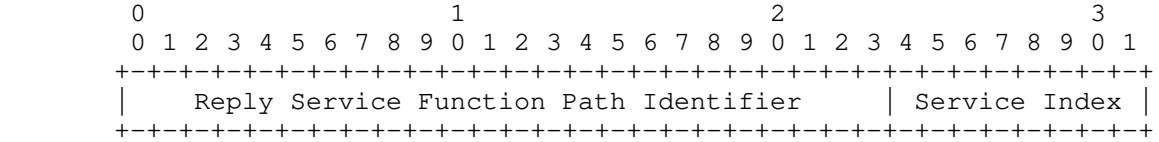


Figure 2: Reply Service Function Path Field Format

where:

- Reply Service Function Path Identifier: SFP identifier for the path that the SFC Echo Reply message is requested to be sent over.
- Service Index: used for forwarding in the reply SFP.

5. Theory of Operation

[RFC7110] defined mechanism to control return path for MPLS LSP Echo Reply. In case of SFC, the return path is a SFP along which SFC Echo Reply message MUST be transmitted. Hence, the SFC Reply Path TLV included in the SFC Echo Request message MUST sufficiently identify the SFP that the sender of the Echo Request message expects the receiver to use for the corresponding SFC Echo Reply.

When sending an Echo Request, the sender MUST set the value of Reply Mode field to "Reply via Specified Path", defined in [I-D.wang-sfc-multi-layer-oam], and if the specified path is SFC path, the Request MUST include SFC Reply Path TLV. The SFC Reply Path TLV includes identifier of the reverse SFP and an appropriate Service Index.

Echo Reply is expected to be sent by the destination SFF of the SFP being tested or by the SFF at which SFC TTL expires as defined [I-D.ietf-sfc-nsh]. The processing described below equally applies in both cases and referred to as responding SFF.

If the Echo Request message with SFC Reply Path TLV, received by the responding SFF, has Reply Mode value of "Reply via Specified Path" but no SFC Reply Path TLV is present, then the responding SFF MUST send Echo Reply with Return Code set to "Reply Path TLV is missing" value (TBA2). If the responding SFF cannot find requested SFP it MUST send Echo Reply with Return Code set to "Reply SFP was not found" and include the SFC Reply Path TLV from the Echo Request message.

5.1. Bi-directional SFC Case

Ability to specify the return path to be used for Echo Reply is handy in bi-directional SFC. For bi-directional SFC, since the last SFF of the forward SFP may not co-locate with a classifier of the reverse SFP,it is assumed that the last SFF doesn’t know the reply path of a SFC. So even for bi-directional SFC, a reverse SFP also need to be indicated in reply path TLV in echo request message.

6. Security Considerations

Security considerations discussed in [I-D.ietf-sfc-nsh] apply to this document.

In addition, the SFC Return Path extension, defined in this document, can be used for potential "proxying" attacks. For example, an echo request initiator may specify a return path that has a destination different from that of the initiator. But usually, such attacks will

not happen in an SFC domain where the initiators and receivers belong to the same domain, as specified in [RFC7665]. Even if the attack occurs, in order to prevent using the SFC Return Path extension for proxying any possible attacks, the return path SFP SHOULD have a path to reach the sender of the echo request, identified in SFC Source TLV [I-D.wang-sfc-multi-layer-oam]. The receiver MAY drop the echo request when it cannot determine whether the return path SFP has the route to the initiator. That means, when sending echo request, the sender SHOULD choose a proper source address according to specified return path SFP to help the receiver to make the decision.

7. IANA Considerations

7.1. SFC Return Path Type

IANA is requested to assign from its SFC Echo Request/Echo Reply TLV registry new type as follows:

Value	Description	Reference
TBA1	SFC Reply Path Type	This document

Table 1: SFC Return Path Type

7.2. New Return Codes

IANA is requested to assign new return codes from the SFC Echo Request/Echo Reply Return Codes registry as following:

Value	Description	Reference
TBA2	Reply Path TLV is missing	This document
TBA3	Reply SFP was not found	This document

Table 2: SFC Echo Reply Return Codes

8. References

8.1. Normative References

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-28 (work in progress), November 2017.

[I-D.wang-sfc-multi-layer-oam]
Mirsky, G., Meng, W., Khasnabish, B., and C. Wang, "Active OAM for Service Function Chains in Networks", draft-wang-sfc-multi-layer-oam-12 (work in progress), October 2018.

[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>.

[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>.

8.2. Informative References

[RFC7110] Chen, M., Cao, W., Ning, S., Jounay, F., and S. Delord, "Return Path Specified Label Switched Path (LSP) Ping", RFC 7110, DOI 10.17487/RFC7110, January 2014, <https://www.rfc-editor.org/info/rfc7110>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Ting Ao
ZTE Corporation
No.889, BiBo Road
Shanghai 201203
China

Phone: +86 21 68897642
Email: ao.ting@zte.com.cn

Greg Mirsky
ZTE Corp.
1900 McCarthy Blvd. #205
Milpitas, CA 95035
USA

Email: gregimirsky@gmail.com

Zhonghua Chen
China Telecom
No.1835, South PuDong Road
Shanghai 201203
China

Phone: +86 18918588897
Email: 18918588897@189.cn

SFC WG
Internet-Draft
Intended status: Informational
Expires: May 3, 2018

T. Ao
ZTE Corporation
G. Mirsky
ZTE Corp.

October 30, 2017

Analysis of the SFC scalability
draft-ao-sfc-scalability-analysis-03

Abstract

SFC is an ordered set of service function, should be scalable to meet broad range of requirements. The scalability of SFC can be interpreted as ability of the SFC to accommodate one or more SFs joining the SFC , or leaving the SFC without significant impact to SFC performance.

This document presents four aspects on SFC scalability, and provide analysis of the data plane and the control plane to implement the scalable SFC.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2018.

Copyright Notice

Table of Contents

1. Introduction

Service Function Chain (SFC) is the chain with a series of ordered Service Functions(SF). The SFC maybe changed because of load balance , failure, or other management requirement. We call it SFC scalability. The SFC being scalable means that the Service Functions can be added or removed from the path of this SFC without impact on other SFCs and minimal impact in the SFC being modified. With this capability, SFC is more flexible and elastic to adapt all kinds of requirements.

In this document, we will present four use cases on SFC scale-out and scale-in, and analysis some requirements to support SFC scalability.

2. Terminology

SFC(Service Function Chain): An ordered set of some abstract SFs.

SFC Scale-out: One or more SFs are added into the path of the SFC for the sake of load balance, protection or other new services requirement.

SFC Scale-in: One or more SFs are removed from the path of the SFC for the sake of the SFs are by-passed or the SFs are failed.

3. Four Use cases for scale-out/scale-in

Following describes four use cases to illustrate the scalability of the SFC.

3.1. Join

This is SFC horizontal scale-out use case. One or more new SFs must be added to a certain SFC for the traffic that has been classified to require application of new SF(s). This case is the reverse scenario to the by-pass. In this case one or more SFs that were by-passed need to be re-inserted into the SFC. And the SFC itself can be characterized as being scaled out.

There are two sub-cases of an SF joining the SFC. One when both the SF and corresponding SFF are new to the SFC. The second is when the SF attaches to an existing SFF. In the first scenario, control plane needs to notify the upstream SFF to modify its next hop to point to the new SFF and configure the new SFF's forwarding information. In the second scenario control plane needs to configure the existing SFF's forwarding information. In this scenario, SFF forwards the packets not only according to the SFPID but also according to the metadata in the SFC header.

3.2. Redundancy

3.2.1. SF Redundancy

This is an example of SFC vertical scale-out use case. One or more SFs are added into the SFC to meet the redundancy or load balance requirements for some certain SFs. This case is different from the Join case (section 3.1) in which the SF in this case is the same with one of the SF that is on the path of the SFC. The new SF have the same function with the existing SF, so that the new SF is added into the SFC to protect the existing corresponding SF and to load balance the existing corresponding SF. Figure 1 is the illustration about SF redundancy. In this figure, SF2' is the redundency of SF2, so that when SF2 is down, SF2' can keep working.

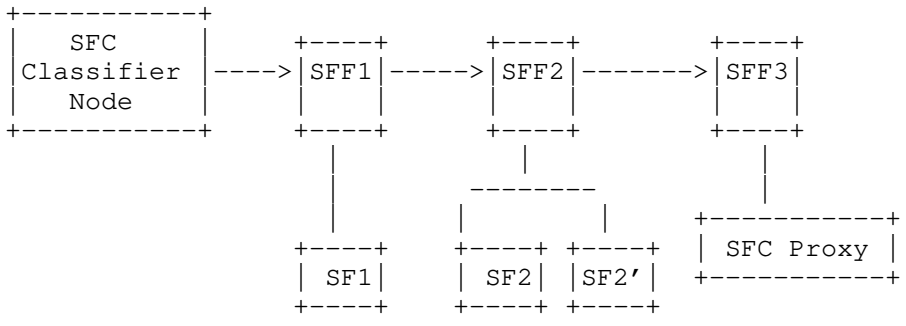


Figure 1

In this case, control plane need to notify the upstream SFF that a new SF joins the SFC as a redundancy SF for protection or load balance, and its next hop should be a protection group or ECMP group. For the purpose of load balance to ensure proper forwarding, the Flow Id field MUST be presented in the NSH as expression of entropy so that SFF can select an SF from the group according to the Flow Id. In the above figure, SFF2 knows that it is connecting a group of SFs and when it foward the packet, it would use Flow id in NSH.

3.2.2. SFC Redundancy

This is also an example of SFC vertical scal-out use case, namely Reduncancy. In this case, SFC is scaled out to two SFP paths. One SFP is redundant to another SFP, and the two SFPs are for protection or load balance. They belongs to a SFC, but have different SFP. The two SFPs are forming a group. Figure 2 is the illustration about the SFC redundancy. In this figure, we can see that SF1', SF2', SFC proxy' are the backup of the SF1, SF2, SFC Proxy seperately. The two SFPs are a group for the Classifier. All these nodes can be joint at some nodes and can be disjoint as well. In the figure 2, all the nodes are disjoint.

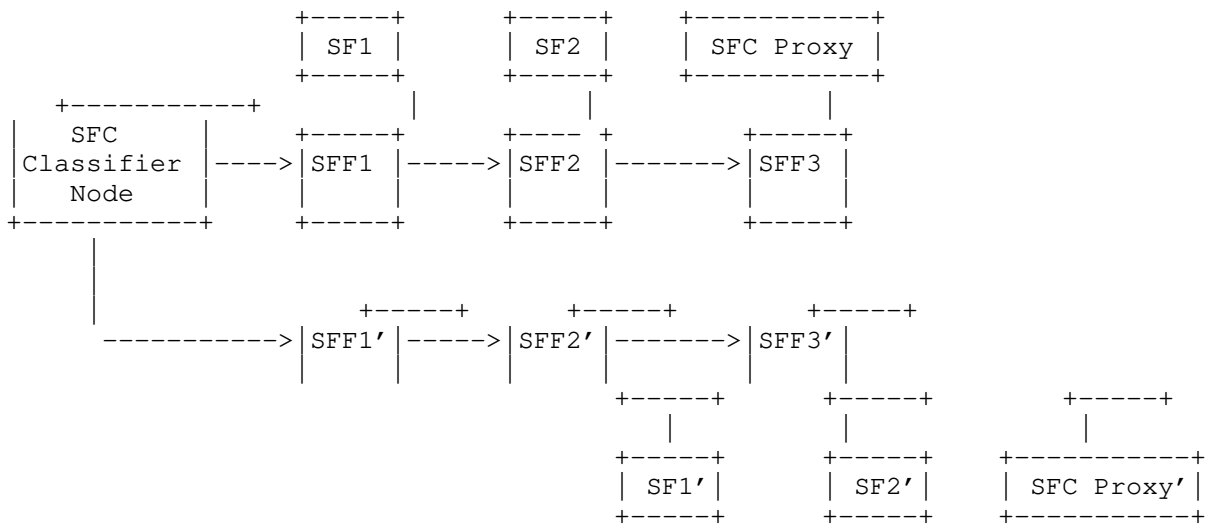


Figure 2

In this case, control plane need to notify the Classifier that the SFC is a group which contains two SFPs. The group can be used as protection or load balance. For the purpose of load balance, to ensure proper forwarding, the Flow Id field MUST be presented in the NSH as expression of entropy so that the forwarder in the classifier can select an SFP from the group according to the Flow Id. For the case of joint, the joint node also need to have capability to forward the traffic accroding to the Flow ID.

3.3. By-pass

This is an example of horizontal scale-in case. In this scenario some SFs are not removed from the SFC but just by-passed by the traffic so that the packets will not be processed by these SFs. Use cases for this scenario are described in [draft-ietf-sfc-long-lived-flow-use-cases] and [draft-ietf-sfc-offloads] . In these two drafts, the SF is offloaded because it is not necessary to steer the traffic to the SFs to improve the forwarding performance.

The corresponding solution is also provided in the above drafts.

3.4. Failure or Remove

This is a vertical SFC scale-in case. This happens only when the SFC is being protected or load balanced. When SF of one SFC has failed or needs to be removed because it is no longer needed to do the pretection, the ability of the SFC to scale-in is excercised.

In this case, the upstream SFF MUST be notified that its next hop has been changed to the next SF of the SF.

From the cases described we can conclude that no matter if is SFC scale-out case or scale-in cases, there are some requirements to SFC control protocol. And for some cases, there are requirements to data plane as well.

4. Data Plane Requirements

For the cases of load balancing or protection switchover of SFC scalability, it is highly beneficial to have an entropy field in the SFC header NSH. The entropy may be presented in the dedicated field named as Flow ID which be part of SFC encapsulation.

This means that SFF not only forwards the traffic based on different SFPID, but also MAY use Flow ID to select particular SF out of set of SFs of the same type.

According to the NSH draft in draft--ietf-sfc-nsh-27, we propose to extend NSH to include the entropy field. Two options can be considered. One is to use existing field, for example, some reserved bits. Suggested extended field in NSH Service Path Header is showed in Figure 3.

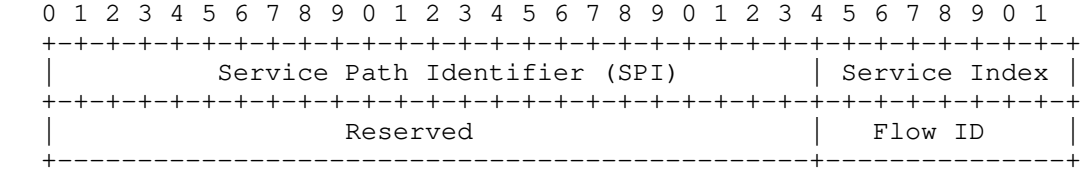


Figure 3

Another is to extend a new metadata to meet the requirement. Which has been described in the section 8 of the draft-quinn-sfc-nsh-tlv-04 .

5. Control Plane Requirements

5.1. Centralized CP

SFC Controller is required to:

a) Send a message to SFF that the joined SF connected to set the correct SFPID and its next hop.

b) Send register message to upstream SFF or classifier with some information. The information not only includes next hop locator, but

also includes an indicator if the next hop is a new joined SF or a group that a new SF that added into. If the indicator is a new joined SF, it means the new SF will join the SFC. If the indicator is a group, it means a new SF or a new SFP will be added into this group for load balance or protection.

c) Send de-register message to upstream SFF or classifier with some information. The information not only includes next hop locator, but also includes an indicator that if the next hop is by-passed, or the next hop is removed from a group. If the indicator is the by-passed SF, it means the current SF is by-passed or is leaving from the SFC. If the indicator is a group SF, it means the current SF or SFP will be removed from a protection group that is for load balance or protection.

5.2. Distributed CP

Distributed SFC CP can be used in Plug-and-Play scenario. Distributed SFC CP required:

a) The SF that needs to join into the SFC or be by-passed by the SFC should explicitly notify the SFF it is associated with.

b) Once get the connection notification from the SF, the associated SFF should send a register message to the upstream SFF with some information. Such information not only includes next hop locator, but also includes an indicator that if the next hop is a new joined SF or the next hop is a new SF that added into a group. If the indicator is a new joined SF, it means a new SF will join the SFC. If the indicator is a group, it means a new SF will be added into a group for load balance or protection.

c) The SFF send de-register message to upstream SFF with some information. Such information not only includes next hop locator, but also includes an indicator that if the next hop is the next SF because the current SF is by-passed, or the next hop is the SF that is removed from a group. If the indicator is the by-passed SF, it means the current SF is by-passed or is leaving from the SFC. If the indicator is group SF, it means the current SF will be removed into a protection group that is for load balance or protection.

6. Security Considerations

For the scalability of the SFC, security is very important to be considered. Before allow the SF to join to the SFC, it is required to make sure the SF’s security first.

7. IANA Considerations

TBD

8. Information References

[I-D.ietf-sfc-architecture]
Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-11 (work in progress), July 2015.

[I-D.ietf-sfc-long-lived-flow-use-cases]
Krishnan, R., Ghanwani, A., Halpern, J., Kini, S., and D. Lopez, "SFC Long-lived Flow Use Cases", draft-ietf-sfc-long-lived-flow-use-cases-03 (work in progress), February 2015.

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-27 (work in progress), October 2017.

[I-D.ietf-sfc-offloads]
Kumar, S., Guichard, J., Quinn, P., Halpern, J., and S. Majee, "Service Function Simple Offloads", draft-ietf-sfc-offloads-00 (work in progress), April 2017.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

Authors' Addresses

Ting Ao
ZTE Corporation
No.889, BiBo Road
Shanghai 201203
China

Phone: +86 21 68897642
Email: ao.ting@zte.com.cn

Greg Mirsky
ZTE Corp.
1900 McCarthy Blvd. #205
Milpitas, CA 95035
USA

Email: gregimirsky@gmail.com

SFC WG
Internet-Draft
Intended status: Informational
Expires: April 22, 2019

T. Ao
ZTE Corporation
G. Mirsky
ZTE Corp.

October 19, 2018

Analysis of the SFC scalability
draft-ao-sfc-scalability-analysis-04

Abstract

SFC is an ordered set of service function, should be scalable to meet broad range of requirements. The scalability of SFC can be interpreted as ability of the SFC to accommodate one or more SFs joining the SFC , or leaving the SFC without significant impact to SFC performance.

This document presents four aspects on SFC scalability, and provide analysis of the data plane and the control plane to implement the scalable SFC.

Table of Contents

1. Introduction

Service Function Chain (SFC) is the chain with a series of ordered Service Functions(SF). The SFC maybe changed because of load balance , failure, or other management requirement. We call it SFC scalability. The SFC being scalable means that the Service Functions can be added or removed from the path of this SFC without impact on other SFCs and minimal impact in the SFC being modified. With this capability, SFC is more flexible and elastic to adapt all kinds of requirements.

In this document, we will present four use cases on SFC scale-out and scale-in, and analysis some requirements to support SFC scalability.

2. Terminology

SFC(Service Function Chain): An ordered set of some abstract SFs.

SFC Scale-out: One or more SFs are added into the path of the SFC for the sake of load balance, protection or other new services requirement.

SFC Scale-in: One or more SFs are removed from the path of the SFC for the sake of the SFs are by-passed or the SFs are failed.

3. Four Use cases for scale-out/scale-in

Following describes four use cases to illustrate the scalability of the SFC.

3.1. Join

This is SFC horizontal scale-out use case. One or more new SFs must be added to a certain SFC for the traffic that has been classified to require application of new SF(s). This case is the reverse scenario to the by-pass. In this case one or more SFs that were by-passed need to be re-inserted into the SFC. And the SFC itself can be characterized as being scaled out.

There are two sub-cases of an SF joining the SFC. One when both the SF and corresponding SFF are new to the SFC. The second is when the SF attaches to an existing SFF. In the first scenario, control plane needs to notify the upstream SFF to modify its next hop to point to the new SFF and configure the new SFF's forwarding information. In the second scenario control plane needs to configure the existing SFF's forwarding information. In this scenario, SFF forwards the packets not only according to the SFPID but also according to the metadata in the SFC header.

3.2. Redundancy

3.2.1. SF Redundancy

This is an example of SFC vertical scale-out use case. One or more SFs are added into the SFC to meet the redundancy or load balance requirements for some certain SFs. This case is different from the Join case (section 3.1) in which the SF in this case is the same with one of the SF that is on the path of the SFC. The new SF have the same function with the existing SF, so that the new SF is added into the SFC to protect the existing corresponding SF and to load balance the existing corresponding SF. Figure 1 is the illustration about SF redundancy. In this figure, SF2' is the redundency of SF2, so that when SF2 is down, SF2' can keep working.

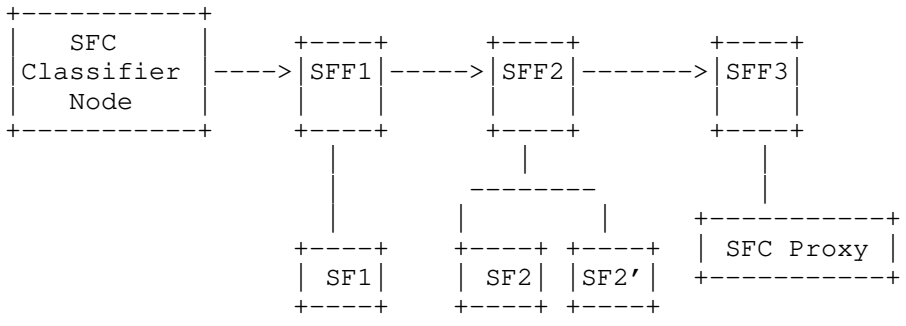


Figure 1

In this case, control plane need to notify the upstream SFF that a new SF joins the SFC as a redundancy SF for protection or load balance, and its next hop should be a protection group or ECMP group. For the purpose of load balance to ensure proper forwarding, the Flow Id field MUST be presented in the NSH as expression of entropy so that SFF can select an SF from the group according to the Flow Id. In the above figure, SFF2 knows that it is connecting a group of SFs and when it foward the packet, it would use Flow id in NSH.

3.2.2. SFC Redundancy

This is also an example of SFC vertical scal-out use case, namely Reduncancy. In this case, SFC is scaled out to two SFP paths. One SFP is redundant to another SFP, and the two SFPs are for protection or load balance. They belongs to a SFC, but have different SFP. The two SFPs are forming a group. Figure 2 is the illustration about the SFC redundancy. In this figure, we can see that SF1', SF2', SFC proxy' are the backup of the SF1, SF2, SFC Proxy seperately. The two SFPs are a group for the Classifier. All these nodes can be joint at some nodes and can be disjoint as well. In the figure 2, all the nodes are disjoint.

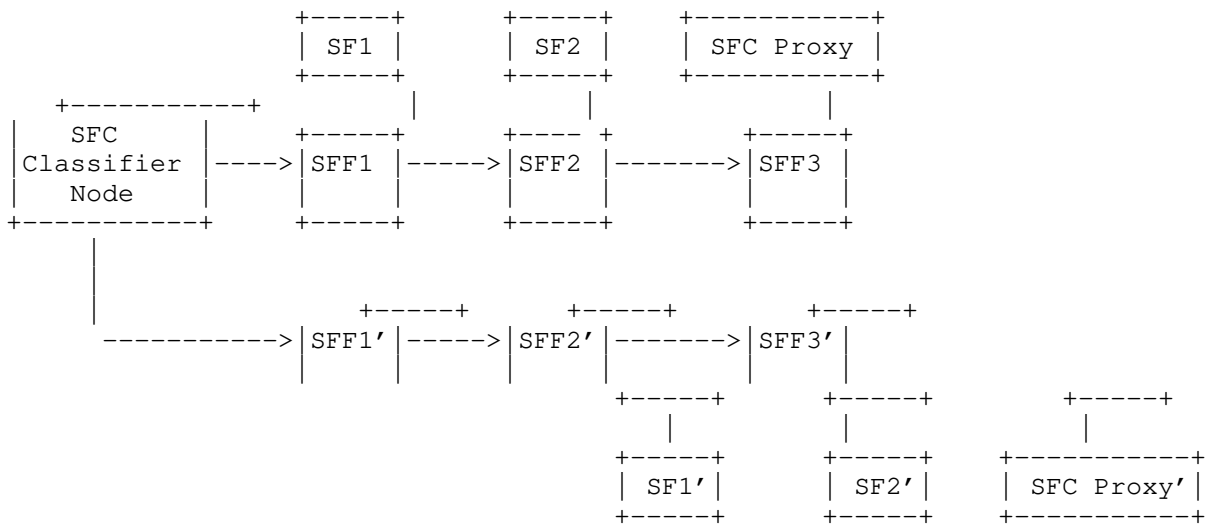


Figure 2

In this case, control plane need to notify the Classifier that the SFC is a group which contains two SFPs. The group can be used as protection or load balance. For the purpose of load balance, to ensure proper forwarding, the Flow Id field MUST be presented in the NSH as expression of entropy so that the forwarder in the classifier can select an SFP from the group according to the Flow Id. For the case of joint, the joint node also need to have capability to forward the traffic accroding to the Flow ID.

3.3. By-pass

This is an example of horizontal scale-in case. In this scenario some SFs are not removed from the SFC but just by-passed by the traffic so that the packets will not be processed by these SFs. Use cases for this scenario are described in [draft-ietf-sfc-long-lived-flow-use-cases] and [draft-ietf-sfc-offloads] . In these two drafts, the SF is offloaded because it is not necessary to steer the traffic to the SFs to improve the forwarding performance.

The corresponding solution is also provided in the above drafts.

3.4. Failure or Remove

This is a vertical SFC scale-in case. This happens only when the SFC is being protected or load balanced. When SF of one SFC has failed or needs to be removed because it is no longer needed to do the pretection, the ability of the SFC to scale-in is excercised.

In this case, the upstream SFF MUST be notified that its next hop has been changed to the next SF of the SF.

From the cases described we can conclude that no matter if is SFC scale-out case or scale-in cases, there are some requirements to SFC control protocol. And for some cases, there are requirements to data plane as well.

4. Data Plane Requirements

For the cases of load balancing or protection switchover of SFC scalability, it is highly beneficial to have an entropy field in the SFC header NSH. The entropy may be presented in the dedicated field named as Flow ID which be part of SFC encapsulation.

This means that SFF not only forwards the traffic based on different SFPID, but also MAY use Flow ID to select particular SF out of set of SFs of the same type.

According to the NSH draft in draft--ietf-sfc-nsh-27, we propose to extend NSH to include the entropy field. Two options can be considered. One is to use existing field, for example, some reserved bits. Suggested extended field in NSH Service Path Header is showed in Figure 3.

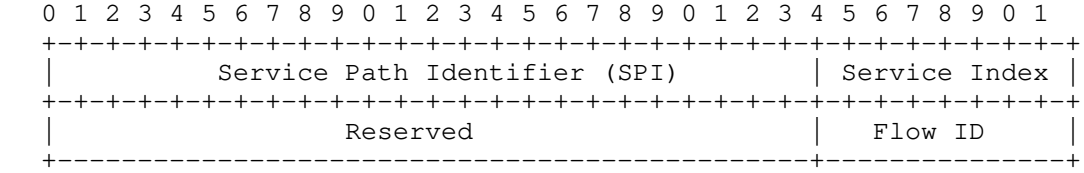


Figure 3

Another is to extend a new metadata to meet the requirement. Which has been described in the section 8 of the draft-quinn-sfc-nsh-tlv-04 .

5. Control Plane Requirements

5.1. Centralized CP

SFC Controller is required to:

a) Send a message to SFF that the joined SF connected to set the correct SFPID and its next hop.

b) Send register message to upstream SFF or classifier with some information. The information not only includes next hop locator, but

also includes an indicator if the next hop is a new joined SF or a group that a new SF that added into. If the indicator is a new joined SF, it means the new SF will join the SFC. If the indicator is a group, it means a new SF or a new SFP will be added into this group for load balance or protection.

c) Send de-register message to upstream SFF or classifier with some information. The information not only includes next hop locator, but also includes an indicator that if the next hop is by-passed, or the next hop is removed from a group. If the indicator is the by-passed SF, it means the current SF is by-passed or is leaving from the SFC. If the indicator is a group SF, it means the current SF or SFP will be removed from a protection group that is for load balance or protection.

5.2. Distributed CP

Distributed SFC CP can be used in Plug-and-Play scenario. Distributed SFC CP required:

a) The SF that needs to join into the SFC or be by-passed by the SFC should explicitly notify the SFF it is associated with.

b) Once get the connection notification from the SF, the associated SFF should send a register message to the upstream SFF with some information. Such information not only includes next hop locator, but also includes an indicator that if the next hop is a new joined SF or the next hop is a new SF that added into a group. If the indicator is a new joined SF, it means a new SF will join the SFC. If the indicator is a group, it means a new SF will be added into a group for load balance or protection.

c) The SFF send de-register message to upstream SFF with some information. Such information not only includes next hop locator, but also includes an indicator that if the next hop is the next SF because the current SF is by-passed, or the next hop is the SF that is removed from a group. If the indicator is the by-passed SF, it means the current SF is by-passed or is leaving from the SFC. If the indicator is group SF, it means the current SF will be removed into a protection group that is for load balance or protection.

6. Security Considerations

For the scalability of the SFC, security is very important to be considered. Before allow the SF to join to the SFC, it is required to make sure the SF’s security first.

7. IANA Considerations

TBD

8. Information References

[I-D.ietf-sfc-architecture]
Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-11 (work in progress), July 2015.

[I-D.ietf-sfc-long-lived-flow-use-cases]
Krishnan, R., Ghanwani, A., Halpern, J., Kini, S., and D. Lopez, "SFC Long-lived Flow Use Cases", draft-ietf-sfc-long-lived-flow-use-cases-03 (work in progress), February 2015.

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-28 (work in progress), November 2017.

[I-D.ietf-sfc-offloads]
Kumar, S., Guichard, J., Quinn, P., Halpern, J., and S. Majee, "Service Function Simple Offloads", draft-ietf-sfc-offloads-00 (work in progress), April 2017.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

Authors' Addresses

Ting Ao
ZTE Corporation
No.889, BiBo Road
Shanghai 201203
China

Phone: +86 21 68897642
Email: ao.ting@zte.com.cn

Greg Mirsky
ZTE Corp.
1900 McCarthy Blvd. #205
Milpitas, CA 95035
USA

Email: gregimirsky@gmail.com

sfc
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

F. Brockners
S. Bhandari
V. Govindan
C. Pignataro
Cisco
H. Gredler
RtBrick Inc.
J. Leddy
Comcast
S. Youell
JMPC
T. Mizrahi
Marvell
D. Mozes
Mellanox Technologies Ltd.
P. Lapukhov
Facebook
R. Chang
Barefoot Networks
October 30, 2017

NSH Encapsulation for In-situ OAM Data
draft-brockners-sfc-ioam-nsh-00

Abstract

In-situ Operations, Administration, and Maintenance (OAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document outlines how IOAM data fields are encapsulated in the Network Service Header (NSH).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2018.

Table of Contents

1. Introduction

In-situ OAM (IOAM) records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the OAM data is added to the data packets rather than is being sent within packets specifically dedicated to OAM. This document defines how IOAM data fields are transported as part of the Network Service Header (NSH) [I-D.ietf-sfc-nsh]) encapsulation. The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. An implementation of IOAM which leverages NSH to carry the IOAM data is available from the FD.io open source software project [FD.io].

2. Conventions

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

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

NSH: Network Service Header

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

SFC: Service Function Chain

TLV: Type, Length, Value

3. IOAM data fields encapsulation in NSH

IOAM data fields are carried within the NSH header following NSH MDx metadata TLVs.

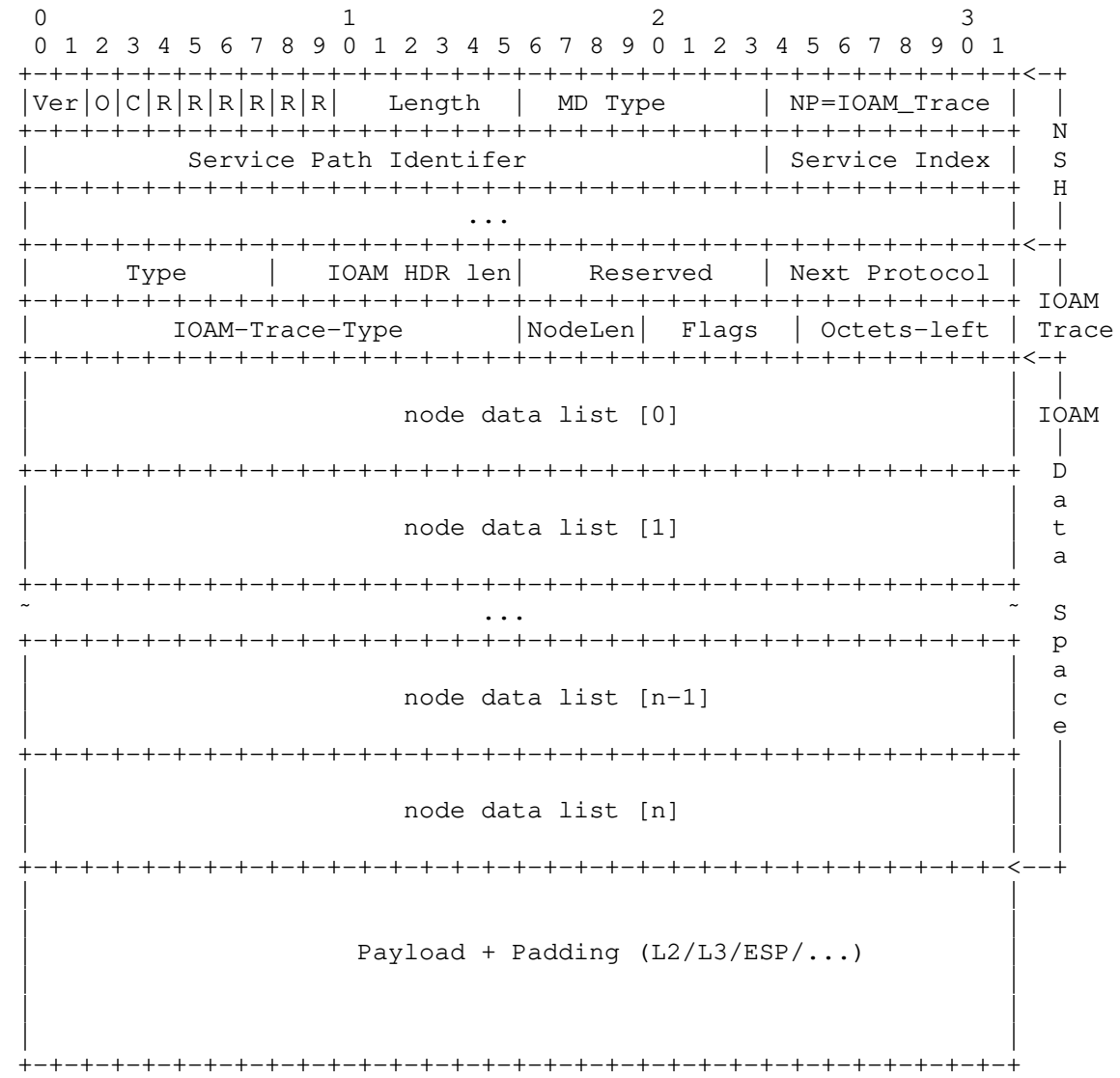
3.1. IOAM Trace Data in NSH

IOAM tracing data represents data that is inserted at nodes that a packet traverses. To allow for optimal implementations in both software as well as hardware forwarders, two different ways to encapsulate IOAM data are defined: "Pre-allocated" and "incremental". See [I-D.ietf-ippm-ioam-data] for details on IOAM tracing and the pre-allocated and incremental IOAM trace options.

The packet formats of the pre-allocated IOAM trace and incremental IOAM trace when transported in NSH are defined as below.

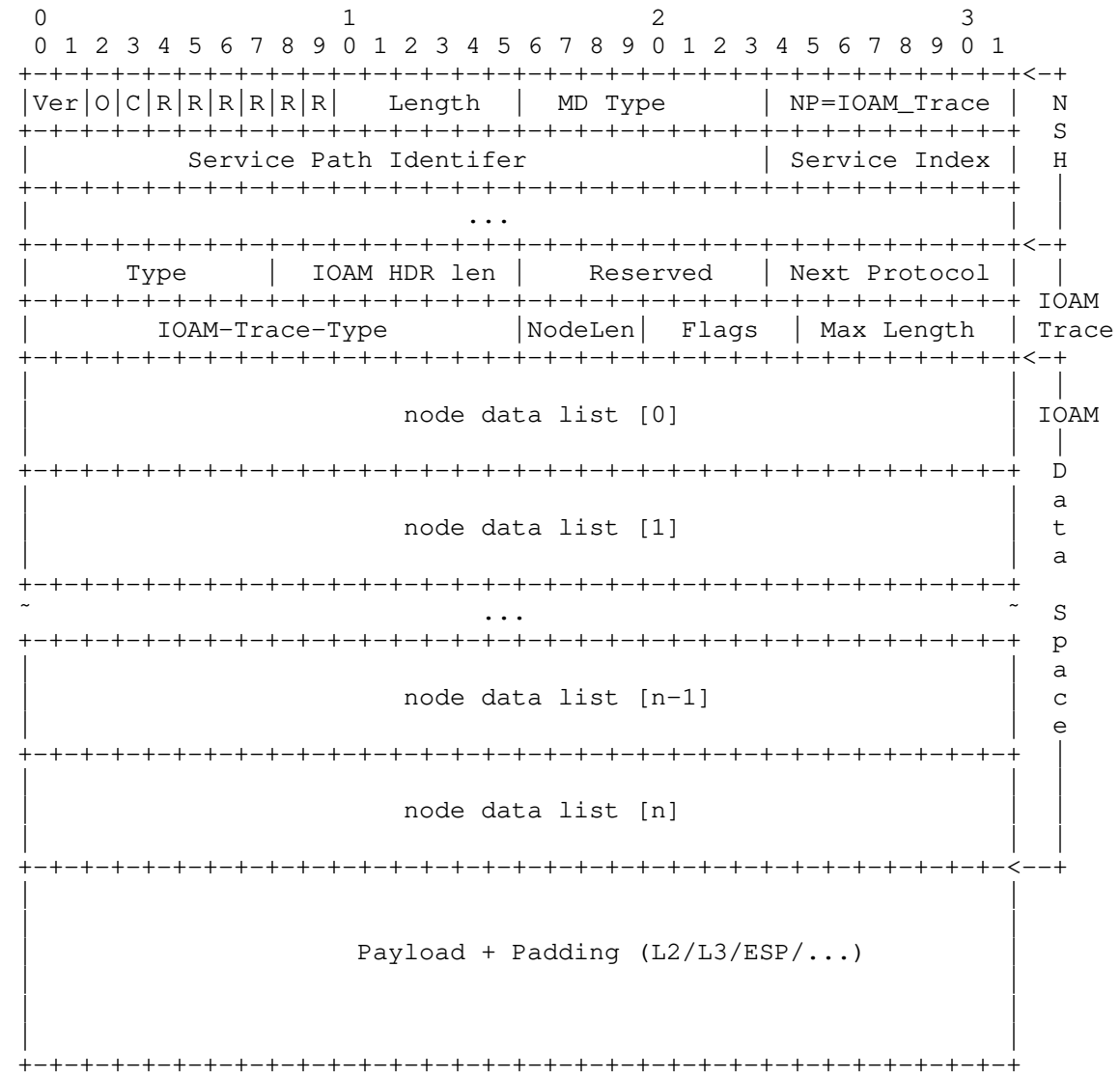
Note that in Service Function Chaining (SFC) [RFC7665], the Network Service Header (NSH) [I-D.ietf-sfc-nsh] already includes path tracing capabilities [I-D.penno-sfc-trace]. IOAM data fields for tracing complement the capabilities in NSH, in that IOAM data fields carry information complementary to information in NSH and benefit from the fact, that IOAM data fields use their own namespace. This allows intermediate nodes, which are not NSH hops to also process and update the IOAM data fields if configured to do so.

IOAM Trace header following NSH MDx header
(Pre-allocated IOAM trace):



IOAM Pre-allocated Trace Option Data MUST be 4-octet aligned:

IOAM Trace header following NSH MDx header
(Incremental IOAM trace):



IOAM Incremental Trace Option Data MUST be 4-octet aligned:

Next Protocol of NSH: TBD value for IOAM_Trace.

Type: 8-bit unsigned integer defining IOAM header type IOAM_TRACE_Preallocated or IOAM_Trace_Incremental are defined here.

IOAM HDR len: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

Reserved bits and R bits: Reserved bits are present for future use. The reserved bits MUST be set to 0x0.

Next Protocol: 8-bit unsigned integer that determines the type of header following IOAM protocol.

IOAM-Trace-Type: 16-bit identifier of IOAM Trace Type as defined in [I-D.ietf-ippm-ioam-data] IOAM-Trace-Types.

Node Data Length: 4-bit unsigned integer as defined in [I-D.ietf-ippm-ioam-data].

Flags: 5-bit field as defined in [I-D.ietf-ippm-ioam-data].

Octets-left: 7-bit unsigned integer as defined in [I-D.ietf-ippm-ioam-data].

Maximum-length: 7-bit unsigned integer as defined in [I-D.ietf-ippm-ioam-data].

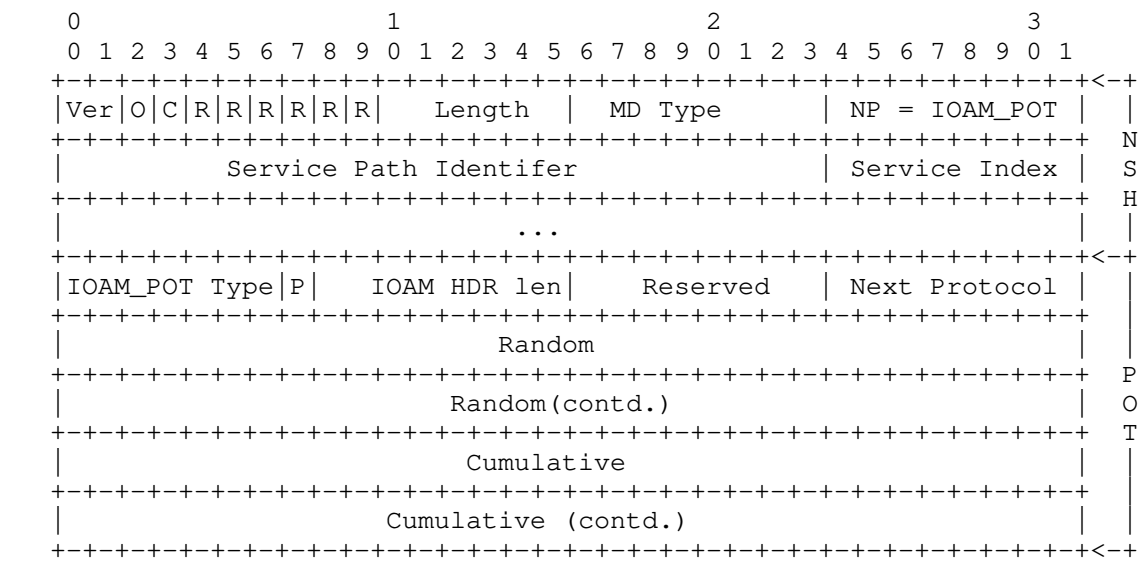
Node data List [n]: Variable-length field as defined in [I-D.ietf-ippm-ioam-data].

3.2. IOAM POT Data in NSH

IOAM proof of transit (POT, see [I-D.brockners-proof-of-transit]) offers a means to verify that a packet has traversed a defined set of nodes. In an administrative domain where IOAM is used, insertion of the IOAM data into the NSH header is enabled at the required nodes (i.e. at the IOAM encapsulating/decapsulating nodes) by means of configuration.

IOAM POT data fields are added as a TLV following NSH MDx metadata:

IOAM POT header following NSH MDx header:



Next Protocol of NSH: TBD value for IOAM_POT.

IOAM POT Type: 7-bit identifier of a particular POT variant that specifies the POT data that is to be included as defined in [I-D.ietf-ippm-ioam-data].

Profile to use (P): 1-bit as defined in [I-D.ietf-ippm-ioam-data] IOAM POT Option.

IOAM HDR len: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

Reserved bits and R bits: Reserved bits are present for future use. The reserved bits MUST be set to 0x0.

Next Protocol: 8-bit unsigned integer that determines the type of header following IOAM protocol.

Random: 64-bit Per-packet random number.

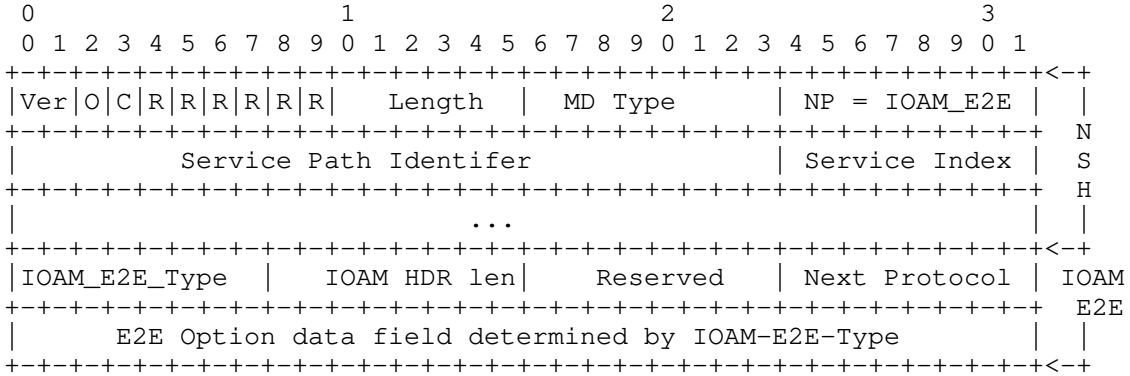
Cumulative: 64-bit Cumulative value that is updated by the Service Functions.

3.3. IOAM Edge-to-Edge Data in NSH

The IOAM edge-to-edge option is to carry data that is added by the IOAM encapsulating node and interpreted by the IOAM decapsulating node. The "Edge-to-Edge" capabilities (see [I-D.brockners-inband-oam-requirements]) of IOAM can be leveraged within NSH. In an administrative domain where IOAM is used, insertion of the IOAM data into the NSH header is enabled at the required nodes (i.e. at the IOAM encapsulating/decapsulating nodes) by means of configuration.

IOAM Edge-to-Edge data fields are added as a TLV following NSH MDx metadata:

IOAM E2E header following NSH MDx header:



Next Protocol of NSH: TBD value for IOAM_E2E.

IOAM E2E Type: 8-bit identifier of a particular E2E variant that specifies the IOAM E2E data that is to be included as defined in [I-D.ietf-ippm-ioam-data].

IOAM HDR len: 8-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

Reserved bits and R bits: Reserved bits are present for future use. The reserved bits MUST be set to 0x0.

Next Protocol: 8-bit unsigned integer that determines the type of header following IOAM protocol.

E2E Option data field: Variable length field as defined in [I-D.ietf-ippm-ioam-data] IOAM E2E Option.

4. Discussion of the encapsulation approach

This section is to support the working group discussion in selecting the most appropriate approach for encapsulating IOAM data fields in NSH.

An encapsulation of IOAM data fields in NSH should be friendly to an implementation in both hardware as well as software forwarders and support a wide range of deployment cases, including large networks that desire to leverage multiple IOAM data fields at the same time.

Hardware and software friendly implementation: Hardware forwarders benefit from an encapsulation that minimizes iterative look-ups of fields within the packet: Any operation which looks up the value of a field within the packet, based on which another lookup is performed, consumes additional gates and time in an implementation - both of which are desired to be kept to a minimum. This means that flat TLV structures are to be preferred over nested TLV structures. IOAM data fields are grouped into three option categories: Trace, proof-of-transit, and edge-to-edge. Each of these three options defines a TLV structure. A hardware-friendly encapsulation approach avoids grouping these three option categories into yet another TLV structure, but would rather carry the options as a serial sequence.

Total length of the IOAM data fields: The total length of IOAM data can grow quite large in case multiple different IOAM data fields are used and large path-lengths need to be considered. If for example an operator would consider using the IOAM trace option and capture node-id, app_data, egress/ingress interface-id, timestamp seconds, timestamps nanoseconds at every hop, then a total of 20 octets would be added to the packet at every hop. In case this particular deployment would have a maximum path length of 15 hops in the IOAM domain, then a maximum of 300 octets of IOAM data were to be encapsulated in the packet.

Two approaches for encapsulating IOAM data fields in NSH could be considered:

1. Encapsulation of IOAM data fields as "NSH MD Type 2" (see [I-D.ietf-sfc-nsh], section 2.5). Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) would be specified by a type, with the different IOAM data fields being TLVs within this the particular option type. NSH MD Type 2 offers support for variable length meta-data. The length field is 6-bits, resulting in a maximum of 256 (2^6 x 4) octets.

2. Encapsulation of IOAM data fields using the "Next Protocol" field. Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) would be specified by its own "next protocol".

The second option has been chosen here, because it avoids the additional layer of TLV nesting that the use of NSH MD Type 2 would result in. In addition, the second option does not constrain IOAM data to a maximum of 256 octets, thus allowing support for very large deployments.

5. IANA Considerations

IANA is requested to allocate protocol numbers for the following NSH "Next Protocols" related to IOAM:

Next Protocol	Description	Reference
x	IOAM_Trace	This document
y	IOAM_POT	This document
z	IOAM_E2E	This document

6. Security Considerations

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

7. Acknowledgements

The authors would like to thank Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, Stefano Previdi, Hemant Singh, Erik Nordmark, LJ Wobker, and Andrew Yourtchenko for the comments and advice.

8. References

8.1. Normative References

[ETYPES] "IANA Ethernet Numbers", <https://www.iana.org/assignments/ethernet-numbers/ethernet-numbers.xhtml>.

[I-D.brockners-inband-oam-requirements]
Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi, T., <>, P., and r. remy@barefootnetworks.com, "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements-03 (work in progress), March 2017.

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-00 (work in progress), September 2017.

[I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-04 (work in progress), April 2017.

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-27 (work in progress), October 2017.

[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>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <https://www.rfc-editor.org/info/rfc3232>.

8.2. Informative References

[FD.io] "Fast Data Project: FD.io", <https://fd.io/>.

[I-D.brockners-proof-of-transit]
Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Leddy, J., Youell, S., Mozes, D., and T. Mizrahi, "Proof of Transit", draft-brockners-proof-of-transit-03 (work in progress), March 2017.

[I-D.ietf-ippm-6man-pdm-option]
Elkins, N., Hamilton, R., and m. mackermann@bcbsm.com, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", draft-ietf-ippm-6man-pdm-option-13 (work in progress), June 2017.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-12 (work in progress), June 2017.

[I-D.kitamura-ipv6-record-route]
Kitamura, H., "Record Route for IPv6 (PR6) Hop-by-Hop Option Extension", draft-kitamura-ipv6-record-route-00 (work in progress), November 2000.

[I-D.penno-sfc-trace]
Penno, R., Quinn, P., Pignataro, C., and D. Zhou, "Services Function Chaining Traceroute", draft-penno-sfc-trace-03 (work in progress), September 2015.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Vengada Prasad Govindan
Cisco Systems, Inc.

Email: venggovi@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam 20692
Israel

Email: talmi@marvell.com

David Mozes
Mellanox Technologies Ltd.

Email: davidm@mellanox.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Remy Chang
Barefoot Networks
2185 Park Boulevard
Palo Alto, CA 94306
US

sfc
Internet-Draft
Intended status: Standards Track
Expires: September 4, 2018

F. Brockners
S. Bhandari
V. Govindan
C. Pignataro
Cisco
H. Gredler
RtBrick Inc.
J. Leddy
Comcast
S. Youell
JMPC
T. Mizrahi
Marvell
D. Mozes

P. Lapukhov
Facebook
R. Chang
Barefoot Networks
March 3, 2018

NSH Encapsulation for In-situ OAM Data
draft-brockners-sfc-ioam-nsh-01

Abstract

In-situ Operations, Administration, and Maintenance (OAM) records operational and telemetry information in the packet while the packet traverses a path between two points in the network. This document outlines how IOAM data fields are encapsulated in the Network Service Header (NSH).

This Internet-Draft will expire on September 4, 2018.

Table of Contents

1. Introduction

In-situ OAM (IOAM) records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the OAM data is added to the data packets rather than is being sent within packets specifically dedicated to OAM. This document defines how IOAM data fields are transported as part of the Network Service Header (NSH) [RFC8300] encapsulation. The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data]. An implementation of IOAM which leverages NSH to carry the IOAM data is available from the FD.io open source software project [FD.io].

2. Conventions

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

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

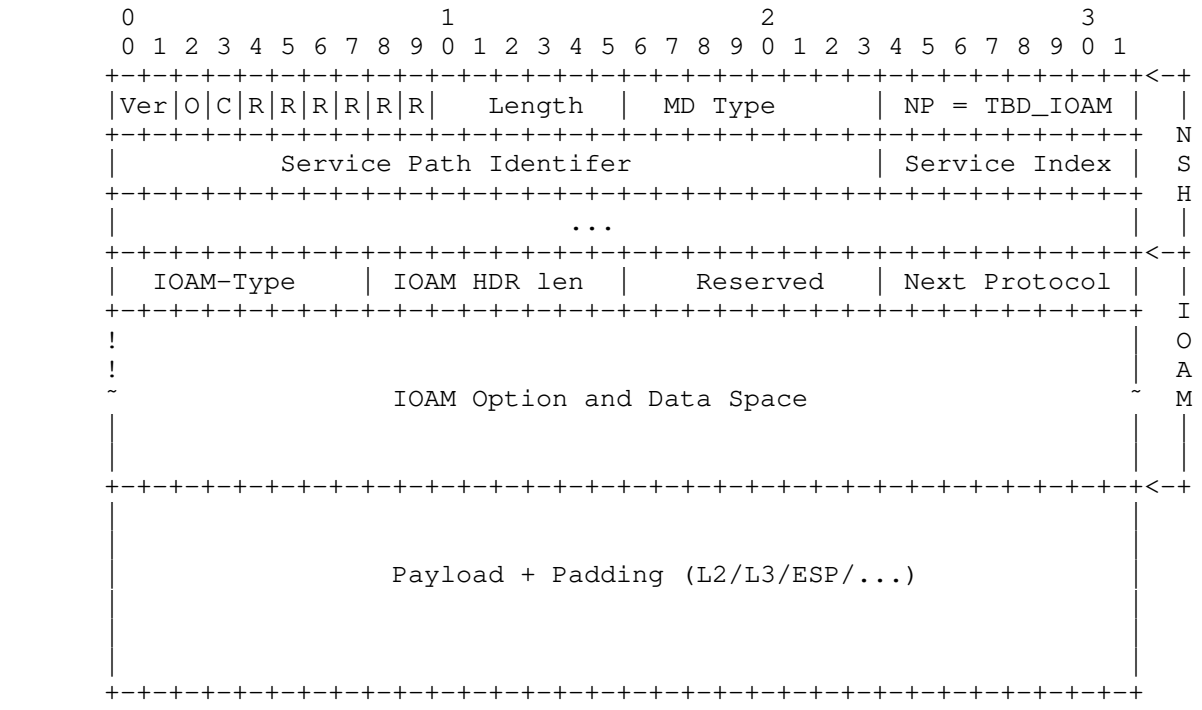
NSH: Network Service Header

OAM: Operations, Administration, and Maintenance

TLV: Type, Length, Value

3. IOAM data fields encapsulation in NSH

NSH is defined in [RFC8300]. IOAM data fields are carried in NSH using a next protocol header which follows the NSH MDx metadata TLVs. An IOAM header is added containing the different IOAM data fields defined in [I-D.ietf-ippm-ioam-data]. In an administrative domain where IOAM is used, insertion of the IOAM header in NSH is enabled at the NSH tunnel endpoints, which also serve as IOAM encapsulating/ decapsulating nodes by means of configuration.



The NSH header and fields are defined in [RFC8300]. The "NSH Next Protocol" value (referred to as "NP" in the diagram above) is TBD_IOAM.

The IOAM related fields in NSH are defined as follows:

IOAM-Type: 8-bit field defining the IOAM Option type, as defined in Section 7.2 of [I-D.ietf-ippm-ioam-data].

IOAM HDR Len: 8 bit Length field contains the length of the IOAM header in 4-octet units.

Reserved bits: Reserved bits are present for future use. The reserved bits MUST be set to 0x0 upon transmission and ignored upon receipt.

Next Protocol: 8-bit unsigned integer that determines the type of header following IOAM protocol.

IOAM Option and Data Space: IOAM option header and data is present as specified by the IOAM-Type field, and is defined in Section 4 of [I-D.ietf-ippm-ioam-data].

Multiple IOAM options MAY be included within the NSH encapsulation. For example, if a NSH encapsulation contains two IOAM options before a data payload, the Next Protocol field of the first IOAM option will contain the value of TBD_IOAM, while the Next Protocol field of the second IOAM option will contain the "NSH Next Protocol" number indicating the type of the data payload.

4. Considerations

This section summarizes a set of considerations on the overall approach taken for IOAM data encapsulation in NSH, as well as deployment considerations.

4.1. Discussion of the encapsulation approach

This section is to support the working group discussion in selecting the most appropriate approach for encapsulating IOAM data fields in NSH.

An encapsulation of IOAM data fields in NSH should be friendly to an implementation in both hardware as well as software forwarders and support a wide range of deployment cases, including large networks that desire to leverage multiple IOAM data fields at the same time.

Hardware and software friendly implementation: Hardware forwarders benefit from an encapsulation that minimizes iterative look-ups of fields within the packet: Any operation which looks up the value of a field within the packet, based on which another lookup is performed, consumes additional gates and time in an implementation - both of which are desired to be kept to a minimum. This means that flat TLV structures are to be preferred over nested TLV structures. IOAM data fields are grouped into three option categories: Trace, proof-of-transit, and edge-to-edge. Each of these three options defines a TLV structure. A hardware-friendly encapsulation approach avoids grouping these three option categories into yet another TLV structure, but would rather carry the options as a serial sequence.

Total length of the IOAM data fields: The total length of IOAM data can grow quite large in case multiple different IOAM data fields are used and large path-lengths need to be considered. If for example an operator would consider using the IOAM trace option and capture node-id, app_data, egress/ingress interface-id, timestamp seconds, timestamps nanoseconds at every hop, then a total of 20 octets would be added to the packet at every hop. In case this particular deployment would have a maximum path length of 15 hops in the IOAM domain, then a maximum of 300 octets of IOAM data were to be encapsulated in the packet.

Different approaches for encapsulating IOAM data fields in NSH could be considered:

1. Encapsulation of IOAM data fields as "NSH MD Type 2" (see [RFC8300], section 2.5). Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) would be specified by a type, with the different IOAM data fields being TLVs within this the particular option type. NSH MD Type 2 offers support for variable length meta-data. The length field is 6-bits, resulting in a maximum of 256 (2^6 x 4) octets.

2. Encapsulation of IOAM data fields using the "Next Protocol" field. Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) would be specified by its own "next protocol".

3. Encapsulation of IOAM data fields using the "Next Protocol" field. A single NSH protocol type code point would be allocated for IOAM. A "sub-type" field would then specify what IOAM options type (trace, proof-of-transit, edge-to-edge) is carried.

The third option has been chosen here. This option avoids the additional layer of TLV nesting that the use of NSH MD Type 2 would result in. In addition, this option does not constrain IOAM data to a maximum of 256 octets, thus allowing support for very large deployments.

4.2. IOAM and the use of the NSH O-bit

[RFC8300] defines an "O bit" for OAM packets. Per [RFC8300] the O bit must be set for OAM packets and must not be set for non-OAM packets. Packets with IOAM data included MUST follow this definition, i.e. the O bit MUST NOT be set for regular customer traffic which also carries IOAM data and the O bit MUST be set for OAM packets which carry only IOAM data without any regular data payload.

5. IANA Considerations

IANA is requested to allocate protocol numbers for the following "NSH Next Protocol" related to IOAM:

Next Protocol	Description	Reference
x	TBD_IOAM	This document

6. Security Considerations

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

7. Acknowledgements

The authors would like to thank Eric Vyncke, Nalini Elkins, Srihari Raghavan, Ranganathan T S, Karthik Babu Harichandra Babu, Akshaya Nadahalli, Stefano Previdi, Hemant Singh, Erik Nordmark, LJ Wobker, and Andrew Yourtchenko for the comments and advice.

8. References

8.1. Normative References

[I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-01 (work in progress), October 2017.

[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>.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <https://www.rfc-editor.org/info/rfc2784>.

[RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <https://www.rfc-editor.org/info/rfc3232>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

8.2. Informative References

[FD.io] "Fast Data Project: FD.io", <https://fd.io/>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Frank Brockners
Cisco Systems, Inc.
Hansaallee 249, 3rd Floor
DUESSELDORF, NORDRHEIN-WESTFALEN 40549
Germany

Email: fbrockne@cisco.com

Shwetha Bhandari
Cisco Systems, Inc.
Cessna Business Park, Sarjapura Marathalli Outer Ring Road
Bangalore, KARNATAKA 560 087
India

Email: shwethab@cisco.com

Vengada Prasad Govindan
Cisco Systems, Inc.

Email: venggovi@cisco.com

Carlos Pignataro
Cisco Systems, Inc.
7200-11 Kit Creek Road
Research Triangle Park, NC 27709
United States

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

Stephen Youell
JP Morgan Chase
25 Bank Street
London E14 5JP
United Kingdom

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam 20692
Israel

Email: talmi@marvell.com

David Mozes

Email: mosesster@gmail.com

Petr Lapukhov
Facebook
1 Hacker Way
Menlo Park, CA 94025
US

Email: petr@fb.com

Remy Chang
Barefoot Networks
2185 Park Boulevard
Palo Alto, CA 94306
US

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

A. Farrel
Juniper Networks
S. Bryant
Huawei
J. Drake
Juniper Networks

October 30, 2017

An MPLS-Based Forwarding Plane for Service Function Chaining
draft-farrel-mpls-sfc-02

Abstract

Service Function Chaining (SFC) is the process of directing packets through a network so that they can be acted on by an ordered set of abstract service functions before being delivered to the intended destination. An architecture for SFC is defined in RFC7665.

The Network Service Header (NSH) can be inserted into packets to steer them along a specific path to realize a Service Function Chain.

Multiprotocol Label Switching (MPLS) is a widely deployed forwarding technology that uses labels to identify the forwarding actions to be taken at each hop through a network. Segment Routing is a mechanism that provides a source routing paradigm for steering packets in an MPLS network.

This document describes how Service Function Chaining can be achieved in an MPLS network by means of a logical representation of the NSH in an MPLS label stack.

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 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

Service Function Chaining (SFC) is the process of directing packets through a network so that they can be acted on by an ordered set of abstract service functions before being delivered to the intended destination. An architecture for SFC is defined in [RFC7665].

When applying a particular Service Function Chain to the traffic selected by a service classifier, the traffic needs to be steered through an ordered set of Service Functions (SFs) in the network. This ordered set of SFs is termed a Service Function Path (SFP), and the traffic is passed between Service Function Forwarders (SFFs) that are responsible for delivering the packets to the SFs and for forwarding them onward to the next SFF.

In order to steer the selected traffic between SFFs and to the correct SFs the service classifier needs to attach information to each packet. This information indicates the SFP on which the packet is being forwarded and hence the SFs to which it must be delivered. The information also indicates the progress the packet has already made along the SFP.

The Network Service Header (NSH) [I-D.ietf-sfc-nsh] has been defined to carry the necessary information for Service Function Chaining in packets. The NSH can be inserted into packets and contains various information including a Service Path Indicator (SPI), a Service Index (SI), and a Time To Live (TTL) counter.

Multiprotocol Label Switching (MPLS) [RFC3031] is a widely deployed forwarding technology that uses labels to identify the forwarding actions to be taken at each hop through a network. In many cases, MPLS will be used as a tunneling technology to carry packets through networks between SFFs.

Segment Routing [RFC7855] introduces a source routing paradigm into packet switched networks. The application of Segment Routing in MPLS networks is described in [I-D.ietf-spring-segment-routing-mpls] and is known as MPLS-SR.

This document describes how Service Function Chaining can be achieved in an MPLS network by means of a logical representation of the NSH in an MPLS label stack. This approach is applicable to both classical MPLS forwarding (where labels are looked up at each hop, and swapped for the next hop [RFC3031]) and MPLS Segment Routing (where labels are looked up at each hop, and popped to reveal the next label to action [I-D.ietf-spring-segment-routing-mpls]). The mechanisms described in this document are a compromise between the full function

that can be achieved using the NSH, and the benefits of reusing the existing MPLS forwarding paradigms.

It is assumed that the reader is fully familiar with the terms and concepts introduced in [RFC7665] and [I-D.ietf-sfc-nsh].

2. Choice of Data Plane SPI/SI Representation

While [I-D.ietf-sfc-nsh] defines the NSH that can be used in a number of environments, this document provides a mechanism to handle situations in which the NSH is not ubiquitously deployed. In this case it is possible to use an alternative data plane representation of the SPI/SI by carrying the identical semantics in MPLS labels.

In order to correctly select the mechanism by which SFC information is encoded and carried between SFFs, it may be necessary to configure the capabilities and choices either within the whole Service Function Overlay Network, or on a hop by hop basis. It is a requirement that both ends of a tunnel over the underlay network know that the tunnel is used for SFC and know what form of NSH representation is used. A control plane signalling approach to achieve these objectives is provided using BGP in [I-D.ietf-bess-nsh-bgp-control-plane].

Note that the encoding of the SFC information is independent of the choice of tunneling technology used between SFFs. Thus, an MPLS representation of the logical NSH (as defined in this document) may be used even if the tunnel between a pair of SFFs is not an MPLS tunnel. Conversely, MPLS tunnels may be used to carry other encodings of the logical NSH (specifically, the NSH itself).

3. Basic Unit of Representation

When an MPLS label stack is used to carry a logical NSH, a basic unit of representation is used. This unit comprises two MPLS labels as shown below. The unit may be present one or more times in the label stack as explained in subsequent sections.

In order to convey the same information as is present in the NSH, two MPLS label stack entries are used. One carries a label to provide context within the SFC scope (the SFC Context Label), and the other carries a label to show which service function is to be actioned (the SF Label). This two-label unit is shown in Figure 1.

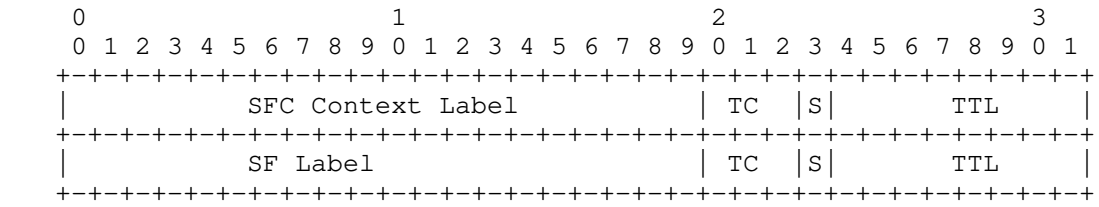


Figure 1: The Basic Unit of MPLS Label Stack for SFC

The fields of these two label stack entries are encoded as follows:

Label: The Label fields contain the values of the SFC Context Label and the SF Label encoded as 20 bit integers. The precise semantics of these label fields are dependent on whether the label stack entries are used for MPLS swapping (see Section 4) or MPLS-SR (see Section 5).

TC: The TC bits have no meaning. They SHOULD be set to zero in both label stack entries and MUST be ignored.

S: The bottom of stack flag has its usual meaning in MPLS. It MUST be clear in the SFC Context label stack entry and MAY be set in the SF label stack entry depending on whether the label is the bottom of stack.

TTL: The TTL field in the SFC Context label stack entry SHOULD be set to 1. The TTL in SF label stack entry (called the SF TTL) is set according to its use for MPLS swapping (see Section 4) or MPLS-SR (see Section 5 and is used to mitigate packet loops.

The sections that follow show how this basic unit of MPLS label stack may be used for SFC in the MPLS label swapping case and in the MPLS-SR case. For simplicity, these sections do not describe the use of metadata: that is covered separately in Section 9.

4. MPLS Label Swapping

This section describes how the basic unit of MPLS label stack for SFC introduced in Section 3 is used when MPLS label swapping is in use. As can be seen from Figure 2, the top of the label stack comprises the labels necessary to deliver the packet over the MPLS tunnel between SFFs. Any MPLS encapsulation may be used (i.e., MPLS, MPLS in UDP, MPLS in GRE, and MPLS in VXLAN or GPE), thus the tunnel technology does not need to be MPLS, but that is shown here for simplicity.

An entropy label ([RFC6790]) may also be present as described in Section 8

Under these labels (or other encapsulation) comes a single instance of the basic unit of MPLS label stack for SFC. In addition to the interpretation of the fields of these label stack entries provided in Section 3 the following meanings are applied:

SPI Label: The Label field of the SFC Context label stack entry contains the value of the SPI encoded as a 20 bit integer. The semantics of the SPI is exactly as defined in [I-D.ietf-sfc-nsh]. Note that an SPI as defined by [I-D.ietf-sfc-nsh] can be encoded in 3 octets (i.e., 24 bits), but that the Label field allows for only 20 bits and reserves the values 0 though 15 as 'special purpose' labels [RFC7274]. Thus, a system using MPLS representation of the logical NSH MUST NOT assign SPI values greater than 2^20 - 1 or less than 16.

SI Label: The Label field of the SF label stack entry contains the value of the SI exactly as defined in [I-D.ietf-sfc-nsh]. Since the SI requires only 8 bits, and to avoid overlap with the 'special purpose' label range of 0 through 15 [RFC7274], the SI is carried in the top (most significant) 8 bits of the Label field with the low order 12 bits set to zero.

TC: The TC fields are as described in Section 3.

S: The S fields are as described in Section 3.

TTL: The TTL field in the SPI label stack entry SHOULD be set to 1 as stated in Section 3. The TTL in SF label stack entry is decremented once for each forwarding hop in the SFP, i.e., for each SFF transited, and so mirrors the TTL field in the NSH.

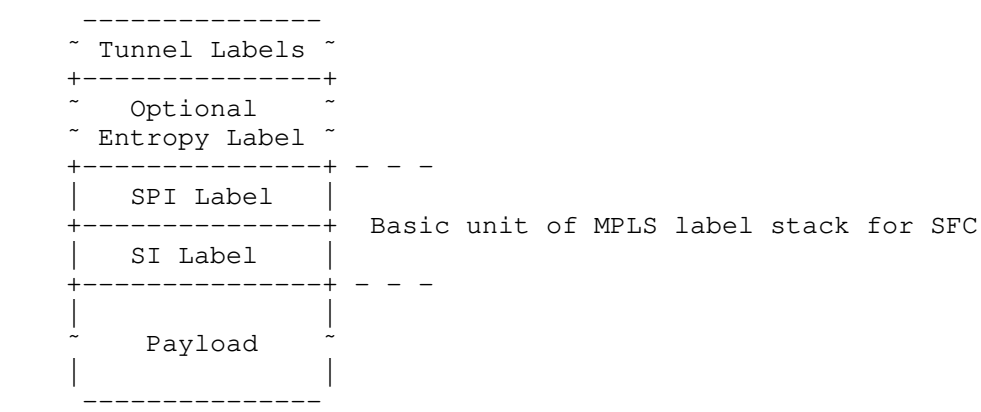


Figure 2: The MPLS SFC Label Stack

The following processing rules apply to the Label fields:

- When a Classifier inserts a packet onto an SFP it sets the SPI Label to indicate the identity of the SFP, and sets the SI Label to indicate the first SF in the path.

- When a component of the SFC system processes a packet it uses the SPI Label to identify the SFP and the SI Label to determine to which SFF or SFI to deliver the packet. Under normal circumstances (with the exception of branching and reclassification - see [I-D.ietf-bess-nsh-bgp-control-plane]) the SPI Label value is preserved on all packets. The SI Label value is modified by SFFs and through reclassification to indicate the next hop along the SFP.

The following processing rules apply to the TTL field of the SF label stack entry, and are derived from section 2.2 of [I-D.ietf-sfc-nsh]:

- When a Classifier places a packet onto an SFP it MUST set the TTL to a value between 1 and 255. It SHOULD set this according to the expected length of the SFP (i.e., the number of SFs on the SFP), but it MAY set it to a larger value according to local configuration. The maximum TTL value supported in an NSH is 63, and so the practical limit here may also be 63.

- When an SFF receives a packet from any component of the SFC system (Classifier, SFI, or another SFF) it MUST discard any packets with TTL set to zero. It SHOULD log such occurrences, but MUST apply rate limiting to any such logs.

- An SFF MUST decrement the TTL by one each time it performs a forwarding lookup.

- If an SFF decrements the TTL to zero it MUST NOT send the packet, and MUST discard the packet. It SHOULD log such occurrences, but MUST apply rate limiting to any such logs.

- SFIs MUST ignore the TTL, but MUST mirror it back to the SFF unmodified along with the SI (which may have been changed by local reclassification).

- If a Classifier along the SFP makes any change to the intended path of the packet including for looping, jumping, or branching (see [I-D.ietf-bess-nsh-bgp-control-plane] it MUST NOT change the SI TTL of the packet. In particular, each component of the SFC system MUST NOT increase the SI TTL value otherwise loops may go undetected.

5. MPLS Segment Routing

This section describes how the basic unit of MPLS label stack for SFC introduced in Section 3 is used when in an MPLS-SR network. As can be seen Figure 3, the top of the label stack comprises the labels necessary to deliver the packet over the MPLS tunnel between SFFs. Any MPLS encapsulation may be used and the tunnel technology does not need to be MPLS or MPLS-SR, but MPLS-SR is shown here for simplicity.

An entropy label ([RFC6790]) may also be present as described in Section 8

Under these labels (or other encapsulation) comes one of more instances of the basic unit of MPLS label stack for SFC. In addition to the interpretation of the fields of these label stack entries provided in Section 3 the following meanings are applied:

SFC Context Label: The Label field of the SFC Context label stack entry contains a label that delivers SFC context. This label may be used to indicate the SPI encoded as a 20 bit integer using the semantics of the SPI is exactly as defined in [I-D.ietf-sfc-nsh] and noting that in this case a system using MPLS representation of the logical NSH MUST NOT assign SPI values greater than 2^20 - 1 or less than 16. This label may also be used to convey other SFC context-speific semantics such as indicating, perhaps with a node SID (see [I-D.ietf-spring-segment-routing]), how to interpret the SF Label.

SF Label: The Label field of the SF label stack entry contains a value that identifies the next SFI to be actioned for the packet.

This label may be scoped globally or within the context of the preceding SFC Context Label and comes from the range 16 ... 2^20 - 1.

TC: The TC fields are as described in Section 3.

S: The S fields are as described in Section 3.

TTL: The TTL field in the SFC Context label stack entry SHOULD be set to 1 as stated in Section 3. The TTL in SF label stack entry is set according to the norms for MPLS-SR.

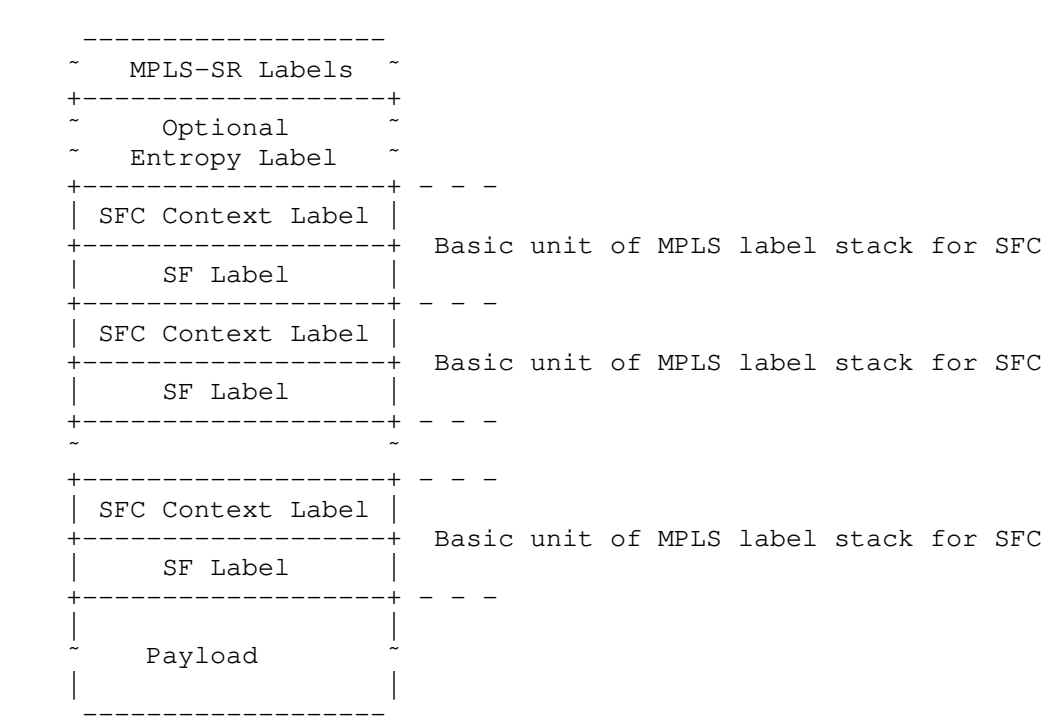


Figure 3: The MPLS SFC Label Stack for Segment Routing

The following processing rules apply to the Label fields:

- When a Classifier inserts a packet onto an SFP it adds a stack comprising one or more instances of the basic unit of MPLS label stack for SFC. Taken together, this stack defines the SFs to be actioned and so defines the SFP that the packet will traverse.

- When a component of the SFC system processes a packet it uses the top basic unit of label stack for SFC to determine to which SFI to next deliver the packet. When an SFF receives a packet it examines the top basic unit of MPLS label stack for SFC to determine where to send the packet next. If the next recipient is a local SFI, the SFC strips the basic unit of MPLS label stack for SFC before forwarding the packet.

6. Mixed Mode Forwarding

The previous sections describe homogeneous networks where SFC forwarding is either all label swapping or all label popping. But it is also possible that different parts of the network utilize swapping or popping for different purposes.

When an SFF receives a packet containing an MPLS label stack, it checks whether it is processing an {SFP, SI} label pair for label swapping or a {context label, SFI index} label pair for MPLS-SR. It then selects the appropriate SFI to which to send the packet. When it receives the packet back from the SFI, it has four cases to consider.

- If the current hop requires an {SFP, SI} and the next hop requires an {SFP, SI}, it sets the SI label to the SI value of the current hop, selects an instance of the SF to be executed at the next hop, and tunnels the packet to the SFF for that SFI.

- If the current hop requires an {SFP, SI} and the next hop requires a {context label, SFI label}, it pops the {SFP, SI} from the top of the MPLS label stack and tunnels the packet to the SFF indicated by the context label.

- If the current hop requires a {context label, SFI label}, it pops the {context label, SFI label} from the top of the MPLS label stack.

  * If the new top of the MPLS label stack contains an {SFP, SI} label pair, it selects an SFI to use at the next hop, and tunnels the packet to SFF for that SFI.

  * If the top of the MPLS label stack contains a {context label, SFI label}, it tunnels the packet to the SFF indicated by the context label.

7. Control Plane Considerations

In order that a packet may be forwarded along an SFP several functional elements must be executed.

- Discovery/advertisement of SFIs.
- Computation of SFP.
- Programming of Classifiers.
- Advertisement of forwarding instructions.

Various approaches may be taken. These include a fully centralized model where SFFs report to a central controller the SFIs that they support, the central controller computes the SFP and programs the Classifiers, and (if the label swapping approach is taken) the central controller installs forwarding state in the SFFs that lie on the SFP.

Alternatively, a dynamic control plane may be used such as that described in [I-D.ietf-bess-nsh-bgp-control-plane]. In this case the SFFs use the control plane to advertise the SFIs that they support, a central controller computes the SFP and programs the Classifiers, and (if the label swapping approach is taken) the central controller uses the control plane to advertise the SFPs so that SFFs that lie on the SFP can install the necessary forwarding state.

8. Use of the Entropy Label

Entropy is used in ECMP situations to ensure that packets from the same flow travel down the same path, thus avoiding jitter or re-ordering issues within a flow.

Entropy is often determined by hashing on specific fields in a packet header such as the "five-tuple" in the IP and transport headers. However, when an MPLS label stack is present, the depth of the stack could be too large for some processors to correctly determine the entropy hash. This problem is addressed by the inclusion of an Entropy Label as described in [RFC6790].

When entropy is desired for packets as they are carried in MPLS tunnels over the underlay network, it is RECOMMENDED that an Entropy Label is included in the label stack immediately after the tunnel labels and before the SFC labels as shown in Figure 2 and Figure 3.

If an Entropy Label is present in a packet received by an SR-capabale node (at the end of a tunnel across the underlay network), it is

RECOMMENDED that the value of that label is preserved and used in an Entropy Label inserted in the label stack when the packet is forwarded (on the next tunnel) to the next SFF.

If an Entropy Label is present in an MPLS payload, it is RECOMMENDED that the initial Classifier use that value in an Entropy Label inserted in the label stack when the packet is forwarded (on the first tunnel) to the first SFF. In this case it is not necessary to remove the Entropy Label from the payload.

9. Metadata

Metadata is defined in [RFC7665] as providing "the ability to exchange context information between classifiers and SFs, and among SFs." [I-D.ietf-sfc-nsh] defines how this context information can be directly encoded in fields that form part of the NSH encapsulation.

The next two sections describe how metadata is associated with user data packets, and how metadata may is exchanged between SFC nodes in the network, when using an MPLS encoding of the logical representation of the NSH.

9.1. Indicating Metadata in User Data Packets

Metadata is achieved in the MPLS realization of the logical NSH by the use of an SFC Metadata Label which uses the Extended Special Purpose Label construct [RFC7274]. Thus, three label stack entries are present as shown in Figure 4:

- The Extension Label (value 15)
- An extended special purpose label called the Metadata Label Indicator (MLI) (value TBD1 by IANA)
- The Metadata Label (ML).

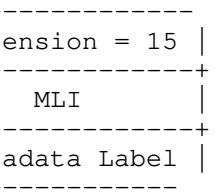


Figure 4: The MPLS SFC Metadata Label

The Metadata Label value is an index into a table of metadata that is programmed into the network using in-band or out-of-band mechanisms. Out-of-band mechanisms potentially include management plane and control plane solutions (such as [I-D.ietf-bess-nsh-bgp-control-plane]), but are out of scope for this document. The in-band mechanism is described in Section 9.2

The SFC Metadata Label (as a set of three labels as indicated in Figure 4) may be present zero, one, or more times in an MPLS SFC packet. For MPLS label swapping, the SFC Metadata Labels are placed immediately after the basic unit of MPLS label stack for SFC as shown in Figure 5. For MPLS-SR, the SFC Metadata Labels can be present zero, one, or more times and are placed at the bottom of the label stack as shown in Figure 6.

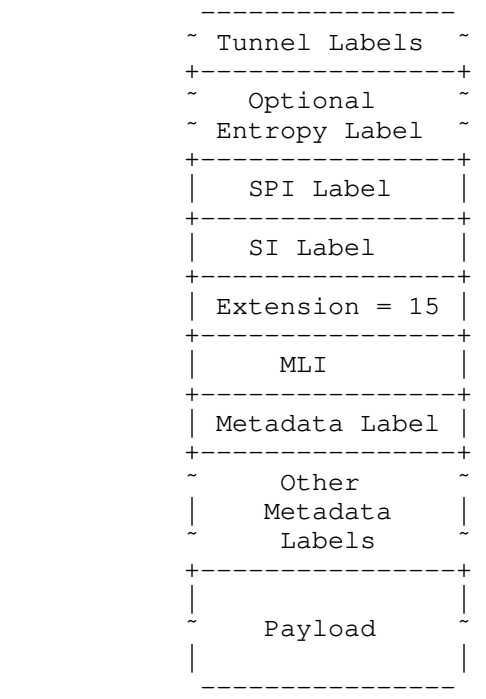


Figure 5: The MPLS SFC Label Stack for Label Swapping with Metadata Label

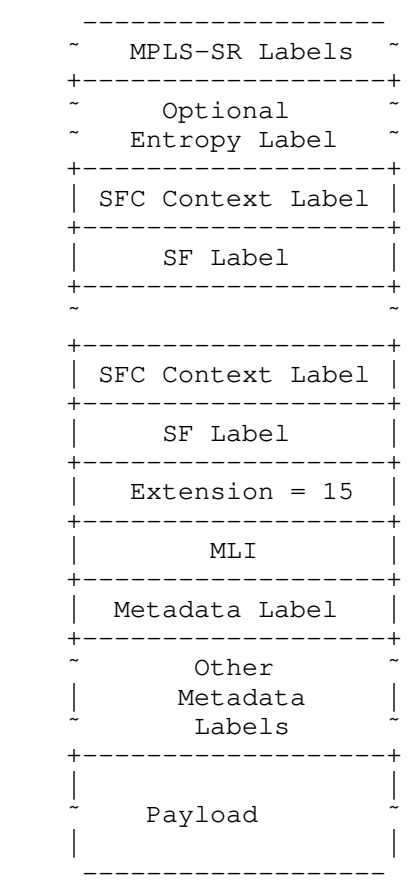


Figure 6: The MPLS SFC Label Stack for MPLS-SR with Metadata Label

9.2. Inband Programming of Metadata

A mechanism for sending metadata associated with an SFP without a payload packet is described in [I-D.farrel-sfc-convent]. The same approach can be used in an MPLS network where the NSH is logically represented by an MPLS label stack.

The packet header is formed exactly as previously described in this document so that the packet will follow the SFP through the SFC network. However, instead of payload data, metadata is included after the bottom of the MPLS label stack. An Extended Special Purpose Label is used to indicate that the metadata is present. Thus, three label stack entries are present:

- The Extension Label (value 15)

- An extended special purpose label called the Metadata Present Indicator (MPI) (value TBD2 by IANA)

- The Metadata Label (ML) that is associated with this metadata on this SFP and can be used to indicate the use of the metadata as described in Section 9.

The SFC Metadata Present Label, if present, is placed immediately after the last basic unit of MPLS label stack for SFC. The resultant label stacks are shown in Figure 7 for the MPLS label swapping case and Figure 8 for the MPLS-SR case.

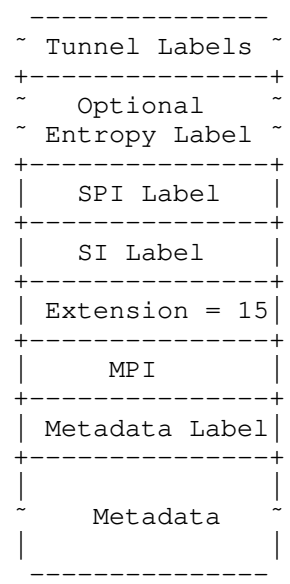


Figure 7: The MPLS SFC Label Stack Carrying Metadata

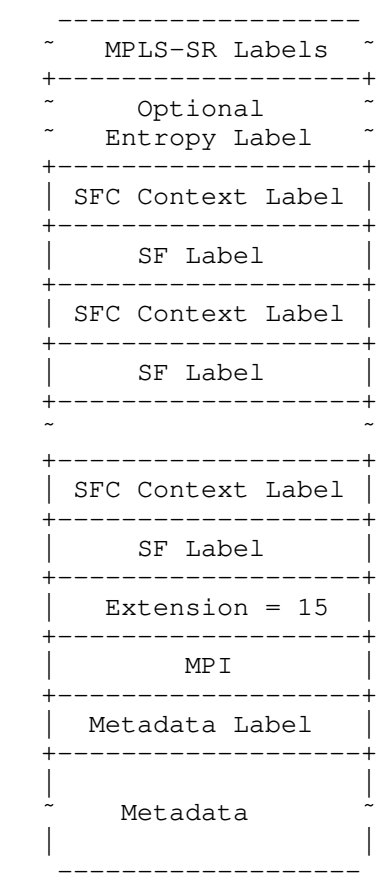


Figure 8: The MPLS SFC Label Stack for MPLS-SR Carrying Metadata

In both cases the metadata is formatted as a TLV as shown in Figure 9.

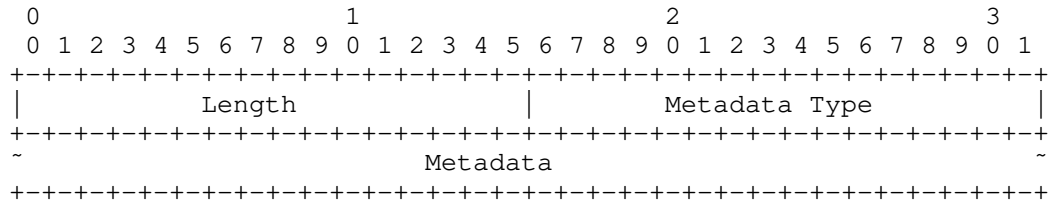


Figure 9: The Metadata TLV

The fields of this TLV are interpreted as follows:

Length: The length of the metadata carried in the Metadata field in octets not including any padding.

Metadata Type: The type of the metadata present. Values for this field are taken from the "MD Types" registry maintained by IANA and defined in [I-D.ietf-sfc-nsh].

Metadata: The actual metadata formatted as described in whatever document defines the metadata. This field is end-padded with zero to three octets of zeroes to take it up to a four octet boundary.

10. Worked Examples

Consider the simplistic MPLS SFC overlay network shown in Figure 10. A packet is classified for an SFP that will see it pass through two Service Functions, SFa and SFb, that are accessed through Service Function Forwarders SFFa and SFFb respectively. The packet is ultimately delivered to destination, D.

Let us assume that the SFP is computed and assigned the SPI of 239. The forwarding details of the SFP are distributed (perhaps using the mechanisms of [I-D.ietf-bess-nsh-bgp-control-plane]) so that the SFFs are programmed with the necessary forwarding instructions.

The packet progresses as follows:

a. The Classifier assigns the packet to the SFP and imposes two label stack entries comprising a single basic unit of MPLS SFC representation:

   * The higher label stack entry contains a label carrying the SPI value of 239.

   * The lower label stack entry contains a label carrying the SI value of 255.

   Further labels may be imposed to tunnel the packet from the Classifier to SFFa.

b. When the packet arrives at SFFa it strips any labels associated with the tunnel that runs from the Classifier to SFFa. SFFa examines the top labels and matches the SPI/SI to identify that the packet should be forwarded to SFa. The packet is forwarded to SFa unmodified.

c. SFa performs its designated function and returns the packet to SFFa.

d. SFFa modifies the SI in the lower label stack entry (to 254) and uses the SPI/SI to look up the forwarding instructions. It sends the packet with two label stack entries:

   * The higher label stack entry contains a label carrying the SPI value of 239.

   * The lower label stack entry contains a label carrying the SI value of 254.

   Further labels may be imposed to tunnel the packet from the SFFa to SFFb.

e. When the packet arrives at SFFb it strips any labels associated with the tunnel from SFFa. SFFb examines the top labels and matches the SPI/SI to identify that the packet should be forwarded to SFb. The packet is forwarded to SFb unmodified.

f. SFb performs its designated function and returns the packet to SFFb.

g. SFFb modifies the SI in the lower label stack entry (to 253) and uses the SPI/SI to lookup up the forwarding instructions. It determines that it is the last SFF in the SFP so it strips the two SFC label stack entries and forwards the payload toward D using the payload protocol.

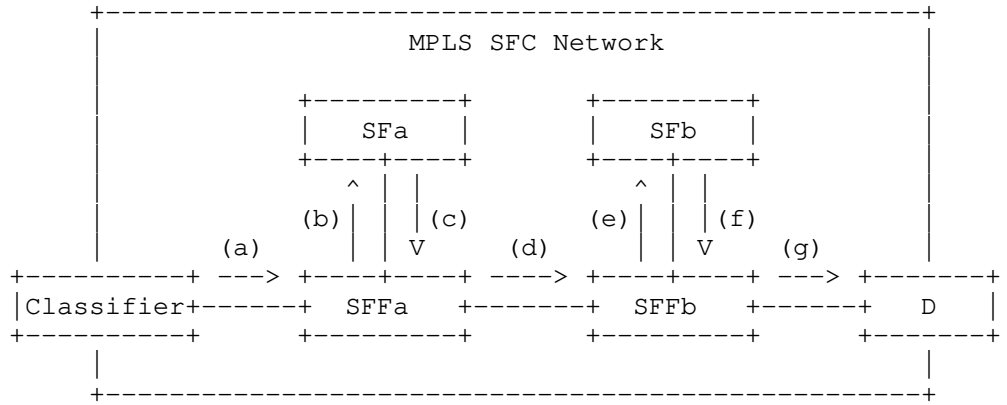


Figure 10: Service Function Chaining in an MPLS Network

Alternatively, consider the MPLS SFC overlay network shown in Figure 11. A packet is classified for an SFP that will see it pass through two Service Functions, SF1 and SF2, that are accessed through Service Function Forwarders SFF1 and SFF2 respectively. The packet is ultimately delivered to destination, D.

Let us assume that the SFP is computed and assigned the SPI of 239. However, the forwarding state for the SFP is not distributed and installed in the network. Instead it will be attached to the individual packets using MPLS-SR.

The packet progresses as follows:

1. The Classifier assigns the packet to the SFP and imposes two basic units of MPLS SFC representation to describe the full SFP:

   * The top basic unit comprises two label stack entries as follows:

     + The higher label stack entry contains a label carrying the SFC context.

     + The lower label stack entry contains a label carrying the SF indicator for SF1.

   * The lower basic unit comprises two label stack entries as follows:

     + The higher label stack entry contains a label carrying the SFC context.

     + The lower label stack entry contains a label carrying the SF indicator for SF2.

   Further labels may be imposed to tunnel the packet from the Classifier to SFF1.

2. When the packet arrives at SFF1 it strips any labels associated with the tunnel from the Classifier. SFF1 examines the top labels and matches the context/SF values to identify that the packet should be forwarded to SF1. The packet is forwarded to SF1 unmodified.

3. SF1 performs its designated function and returns the packet to SFF1.

4. SFF1 strips the top basic unit of MPLS SFC representation revealing the next basic unit. It then uses the revealed

context/SF values to determine how to route the packet to the next SFF, SFF2. It sends the packet with just one basic unit of MPLS SFC representation comprising two label stack entries:

* The higher label stack entry contains a label carrying the SFC context.

* The lower label stack entry contains a label carrying the SF indicator for SF2.

Further labels may be imposed to tunnel the packet from the SFF1 to SFF2.

5. When the packet arrives at SFF2 it strips any labels associated with the tunnel from SFF1. SFF2 examines the top labels and matches the context/SF values to identify that the packet should be forwarded to SF2. The packet is forwarded to SF2 unmodified.

6. SF2 performs its designated function and returns the packet to SFF2.

7. SFF2 strips the top basic unit of MPLS SFC representation revealing the payload packet. It forwards the payload toward D using the payload protocol.

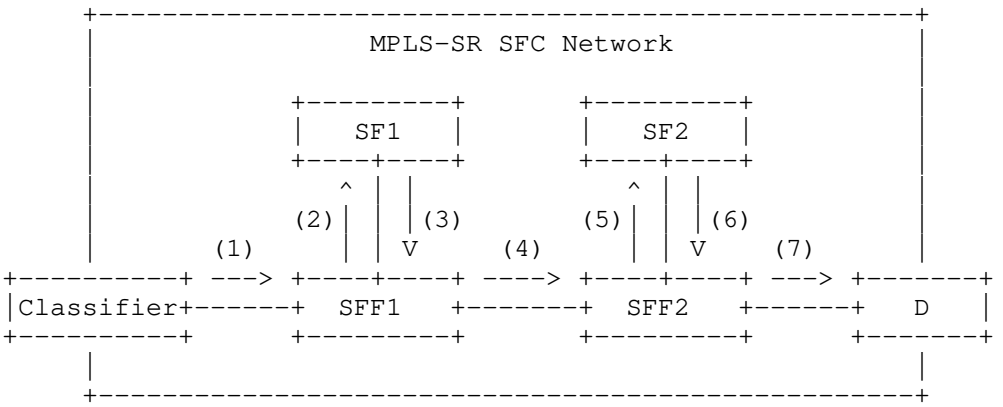


Figure 11: Service Function Chaining in an MPLS-SR Network

11. Security Considerations

Discussion of the security properties of SFC networks can be found in [RFC7665]. Further security discussion for the NSH and its use is present in [I-D.ietf-sfc-nsh].

It is fundamental to the SFC design that the classifier is a trusted resource which determines the processing that the packet will be subject to, including for example the firewall. It is also fundamental to the Segment Routing design that packets are routed through the network using the path specified by the node imposing the SIDs. Where an SF is not encapsulation aware the packet may exist as an IP packet, however this is an intrinsic part of the SFC design which needs to define how a packet is protected in that environment. Where a tunnel is used to link two non-MPLS domains, the tunnel design needs to specify how it is secured. Thus the security vulnerabilities are addressed in the underlying technologies used by this design, which itself does not introduce any new security vulnerabilities.

12. IANA Considerations

This document requests IANA to make allocations from the "Extended Special-Purpose MPLS Label Values" subregistry of the "Special-Purpose Multiprotocol Label Switching (MPLS) Label Values" registry as follows:

Value	Description	
TBD1	Metadata Label Indicator (MLI)	[This.I-D]
TBD2	Metadata Present Indicator (MPI)	[This.I-D]

13. Acknowledgements

This document derives ideas and text from [I-D.ietf-bess-nsh-bgp-control-plane].

The authors are grateful to all those who contributed to the discussions that led to this work: Loa Andersson, Andrew G. Malis, Alexander Vainshtein, Joel M. Halpern, Tony Przygienda, Stuart Mackie, Keyur Patel, and Jim Guichard.

14. References

14.1. Normative References

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-27 (work in progress), October 2017.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-10 (work in progress), June 2017.

[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>.

[RFC7274] Kompella, K., Andersson, L., and A. Farrel, "Allocating and Retiring Special-Purpose MPLS Labels", RFC 7274, DOI 10.17487/RFC7274, June 2014, <https://www.rfc-editor.org/info/rfc7274>.

14.2. Informative References

[I-D.farrel-sfc-convent]
Farrel, A. and J. Drake, "Operating the Network Service Header (NSH) with Next Protocol "None"", draft-farrel-sfc-convent-03 (work in progress), October 2017.

[I-D.ietf-bess-nsh-bgp-control-plane]
Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for NSH SFC", draft-ietf-bess-nsh-bgp-control-plane-01 (work in progress), September 2017.

[I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-13 (work in progress), October 2017.

[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>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7855] Previdi, S., Ed., Filsfils, C., Ed., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "Source Packet Routing in Networking (SPRING) Problem Statement and Requirements", RFC 7855, DOI 10.17487/RFC7855, May 2016, <https://www.rfc-editor.org/info/rfc7855>.

Authors' Addresses

Adrian Farrel
Juniper Networks

Email: afarrel@juniper.net

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

John Drake
Juniper Networks

Email: jdrake@juniper.net

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 23, 2018

A. Farrel
Juniper Networks
S. Bryant
Huawei
J. Drake
Juniper Networks

March 22, 2018

An MPLS-Based Forwarding Plane for Service Function Chaining
draft-farrel-mpls-sfc-05

Abstract

Service Function Chaining (SFC) is the process of directing packets through a network so that they can be acted on by an ordered set of abstract service functions before being delivered to the intended destination. An architecture for SFC is defined in RFC7665.

The Network Service Header (NSH) can be inserted into packets to steer them along a specific path to realize a Service Function Chain.

Multiprotocol Label Switching (MPLS) is a widely deployed forwarding technology that uses labels placed in a packet in a label stack to identify the forwarding actions to be taken at each hop through a network. Actions may include swapping or popping the labels as well, as using the labels to determine the next hop for forwarding the packet. Labels may also be used to establish the context under which the packet is forwarded.

This document describes how Service Function Chaining can be achieved in an MPLS network by means of a logical representation of the NSH in an MPLS label stack. It does not deprecate or replace the NSH, but acknowledges that there may be a need for an interim deployment of SFC functionality in brownfield networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 23, 2018.

Table of Contents

1. Introduction

Service Function Chaining (SFC) is the process of directing packets through a network so that they can be acted on by an ordered set of abstract service functions before being delivered to the intended destination. An architecture for SFC is defined in [RFC7665].

When applying a particular Service Function Chain to the traffic selected by a service classifier, the traffic needs to be steered through an ordered set of Service Functions (SFs) in the network. This ordered set of SFs is termed a Service Function Path (SFP), and the traffic is passed between Service Function Forwarders (SFFs) that are responsible for delivering the packets to the SFs and for forwarding them onward to the next SFF.

In order to steer the selected traffic between SFFs and to the correct SFs the service classifier needs to attach information to each packet. This information indicates the SFP on which the packet is being forwarded and hence the SFs to which it must be delivered. The information also indicates the progress the packet has already made along the SFP.

The Network Service Header (NSH) [RFC8300] has been defined to carry the necessary information for Service Function Chaining in packets. The NSH can be inserted into packets and contains various information including a Service Path Indicator (SPI), a Service Index (SI), and a Time To Live (TTL) counter.

Multiprotocol Label Switching (MPLS) [RFC3031] is a widely deployed forwarding technology that uses labels placed in a packet in a label stack to identify the forwarding actions to be taken at each hop through a network. Actions may include swapping or popping the labels as well, as using the labels to determine the next hop for forwarding the packet. Labels may also be used to establish the context under which the packet is forwarded. In many cases, MPLS will be used as a tunneling technology to carry packets through networks between SFFs.

This document describes how Service Function Chaining can be achieved in an MPLS network by means of a logical representation of the NSH in an MPLS label stack. This approach is applicable to all forms of MPLS forwarding (where labels are looked up at each hop, and swapped or popped [RFC3031]). It does not deprecate or replace the NSH, but acknowledges that there may be a need for an interim deployment of SFC functionality in brownfield networks. The mechanisms described in this document are a compromise between the full function that can be achieved using the NSH, and the benefits of reusing the existing MPLS forwarding paradigms.

It is assumed that the reader is fully familiar with the terms and concepts introduced in [RFC7665] and [RFC8300].

Note that one of the features of the SFC architecture described in [RFC7665] is the "SFC proxy" that exists to include legacy SFs that are not able to process NSH-encapsulated packets. This issue is equally applicable to the use of MPLS-encapsulated packets that encode a logical representation of an NSH. It is discussed further in Section 8.

2. Requirements Language

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

3. Choice of Data Plane SPI/SI Representation

While [RFC8300] defines the NSH that can be used in a number of environments, this document provides a mechanism to handle situations in which the NSH is not ubiquitously deployed. In this case it is possible to use an alternative data plane representation of the SPI/SI by carrying the identical semantics in MPLS labels.

In order to correctly select the mechanism by which SFC information is encoded and carried between SFFs, it may be necessary to configure the capabilities and choices either within the whole Service Function Overlay Network, or on a hop by hop basis. It is a requirement that both ends of a tunnel over the underlay network (i.e., a pair of SFFs adjacent in the SFC) know that the tunnel is used for SFC and know what form of NSH representation is used. A control plane signalling approach to achieve these objectives is provided using BGP in [I-D.ietf-bess-nsh-bgp-control-plane].

Note that the encoding of the SFC information is independent of the choice of tunneling technology used between SFFs. Thus, an MPLS representation of the logical NSH (as defined in this document) may be used even if the tunnel between a pair of SFFs is not an MPLS tunnel. Conversely, MPLS tunnels may be used to carry other encodings of the logical NSH (specifically, the NSH itself).

4. Basic Unit of Representation

When an MPLS label stack is used to carry a logical NSH, a basic unit of representation is used. This unit comprises two MPLS labels as

shown below. The unit may be present one or more times in the label stack as explained in subsequent sections.

In order to convey the same information as is present in the NSH, two MPLS label stack entries are used. One carries a label to provide context within the SFC scope (the SFC Context Label), and the other carries a label to show which service function is to be actioned (the SF Label). This two-label unit is shown in Figure 1.

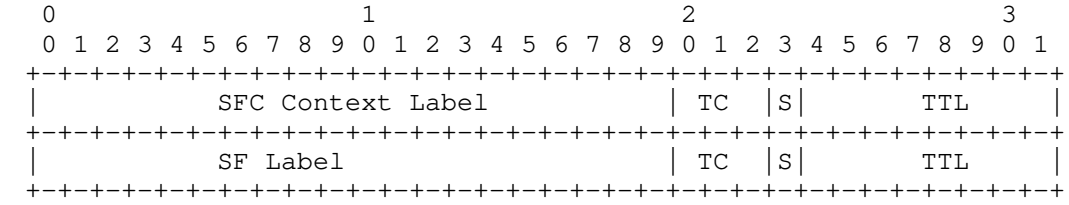


Figure 1: The Basic Unit of MPLS Label Stack for SFC

The fields of these two label stack entries are encoded as follows:

Label: The Label fields contain the values of the SFC Context Label and the SF Label encoded as 20 bit integers. The precise semantics of these label fields are dependent on whether the label stack entries are used for MPLS label swapping (see Section 5) or MPLS label stacking (see Section 6).

TC: The TC bits have no meaning. They SHOULD be set to zero in both label stack entries when a packet is sent and MUST be ignored on receipt.

S: The bottom of stack bit has its usual meaning in MPLS. It MUST be clear in the SFC Context label stack entry and MAY be set in the SF label stack entry depending on whether the label is the bottom of stack.

TTL: The TTL field in the SFC Context label stack entry SHOULD be set to 1. The TTL in SF label stack entry (called the SF TTL) is set according to its use for MPLS label swapping (see Section 5) or MPLS label stacking (see Section 6 and is used to mitigate packet loops.

The sections that follow show how this basic unit of MPLS label stack may be used for SFC in the MPLS label swapping case and in the MPLS label stacking. For simplicity, these sections do not describe the use of metadata: that is covered separately in Section 11.

5. MPLS Label Swapping

This section describes how the basic unit of MPLS label stack for SFC introduced in Section 4 is used when MPLS label swapping is in use. As can be seen from Figure 2, the top of the label stack comprises the labels necessary to deliver the packet over the MPLS tunnel between SFFs. Any MPLS encapsulation may be used (i.e., MPLS, MPLS in UDP, MPLS in GRE, and MPLS in VXLAN or GPE), thus the tunnel technology does not need to be MPLS, but that is shown here for simplicity.

An entropy label ([RFC6790]) may also be present as described in Section 10

Under these labels (or other encapsulation) comes a single instance of the basic unit of MPLS label stack for SFC. In addition to the interpretation of the fields of these label stack entries provided in Section 4 the following meanings are applied:

SPI Label: The Label field of the SFC Context label stack entry contains the value of the SPI encoded as a 20 bit integer. The semantics of the SPI is exactly as defined in [RFC8300]. Note that an SPI as defined by [RFC8300] can be encoded in 3 octets (i.e., 24 bits), but that the Label field allows for only 20 bits and reserves the values 0 though 15 as 'special purpose' labels [RFC7274]. Thus, a system using MPLS representation of the logical NSH MUST NOT assign SPI values greater than 2^20 - 1 or less than 16.

SI Label: The Label field of the SF label stack entry contains the value of the SI exactly as defined in [RFC8300]. Since the SI requires only 8 bits, and to avoid overlap with the 'special purpose' label range of 0 through 15 [RFC7274], the SI is carried in the top (most significant) 8 bits of the Label field with the low order 12 bits set to zero.

TC: The TC fields are as described in Section 4.

S: The S bits are as described in Section 4.

TTL: The TTL field in the SPI label stack entry SHOULD be set to 1 as stated in Section 4. The TTL in SF label stack entry is decremented once for each forwarding hop in the SFP, i.e., for each SFF transited, and so mirrors the TTL field in the NSH.

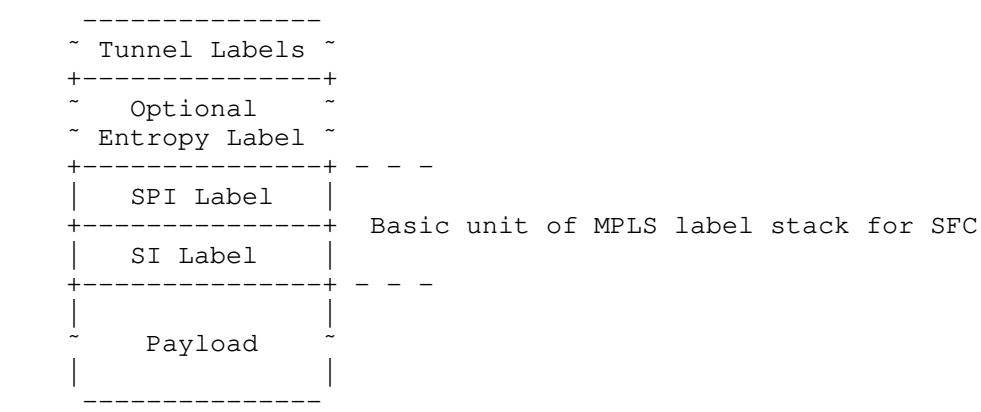


Figure 2: The MPLS SFC Label Stack

The following processing rules apply to the Label fields:

- When a Classifier inserts a packet onto an SFP it sets the SPI Label to indicate the identity of the SFP, and sets the SI Label to indicate the first SF in the path.

- When a component of the SFC system processes a packet it uses the SPI Label to identify the SFP and the SI Label to determine to which SFF or instance of an SF (an SFI) to deliver the packet. Under normal circumstances (with the exception of branching and reclassification - see [I-D.ietf-bess-nsh-bgp-control-plane]) the SPI Label value is preserved on all packets. The SI Label value is modified by SFFs and through reclassification to indicate the next hop along the SFP.

The following processing rules apply to the TTL field of the SF label stack entry, and are derived from section 2.2 of [RFC8300]:

- When a Classifier places a packet onto an SFP it MUST set the TTL to a value between 1 and 255. It SHOULD set this according to the expected length of the SFP (i.e., the number of SFs on the SFP), but it MAY set it to a larger value according to local configuration. The maximum TTL value supported in an NSH is 63, and so the practical limit here may also be 63.

- When an SFF receives a packet from any component of the SFC system (Classifier, SFI, or another SFF) it MUST discard any packets with TTL set to zero. It SHOULD log such occurrences, but MUST apply rate limiting to any such logs.

- An SFF MUST decrement the TTL by one each time it performs a forwarding lookup.

- If an SFF decrements the TTL to zero it MUST NOT send the packet, and MUST discard the packet. It SHOULD log such occurrences, but MUST apply rate limiting to any such logs.

- SFIs MUST ignore the TTL, but MUST mirror it back to the SFF unmodified along with the SI (which may have been changed by local reclassification).

- If a Classifier along the SFP makes any change to the intended path of the packet including for looping, jumping, or branching (see [I-D.ietf-bess-nsh-bgp-control-plane] it MUST NOT change the SI TTL of the packet. In particular, each component of the SFC system MUST NOT increase the SI TTL value otherwise loops may go undetected.

6. MPLS Label Stacking

This section describes how the basic unit of MPLS label stack for SFC introduced in Section 4 is used when MPLS label stacking is used to carry information about the SFP and SFs to be executed. As can be seen in Figure 3, the top of the label stack comprises the labels necessary to deliver the packet over the MPLS tunnel between SFFs. Any MPLS encapsulation may be used.

An entropy label ([RFC6790]) may also be present as described in Section 10

Under these labels comes one of more instances of the basic unit of MPLS label stack for SFC. In addition to the interpretation of the fields of these label stack entries provided in Section 4 the following meanings are applied:

SFC Context Label: The Label field of the SFC Context label stack entry contains a label that delivers SFC context. This label may be used to indicate the SPI encoded as a 20 bit integer using the semantics of the SPI is exactly as defined in [RFC8300] and noting that in this case a system using MPLS representation of the logical NSH MUST NOT assign SPI values greater than 2^20 - 1 or less than 16. This label may also be used to convey other SFC context-speific semantics such as indicating how to interpret the SF Label or how to forward the packet to the node that offers the SF.

SF Label: The Label field of the SF label stack entry contains a value that identifies the next SFI to be actioned for the packet.

This label may be scoped globally or within the context of the preceding SFC Context Label and comes from the range 16 ... 2^20 - 1.

TC: The TC fields are as described in Section 4.

S: The S bits are as described in Section 4.

TTL: The TTL fields in the SFC Context label stack entry SF label stack entry SHOULD be set to 1 as stated in Section 4, but MAY be set to larger values if the label indicated a forwarding operation towards the node that hosts the SF.

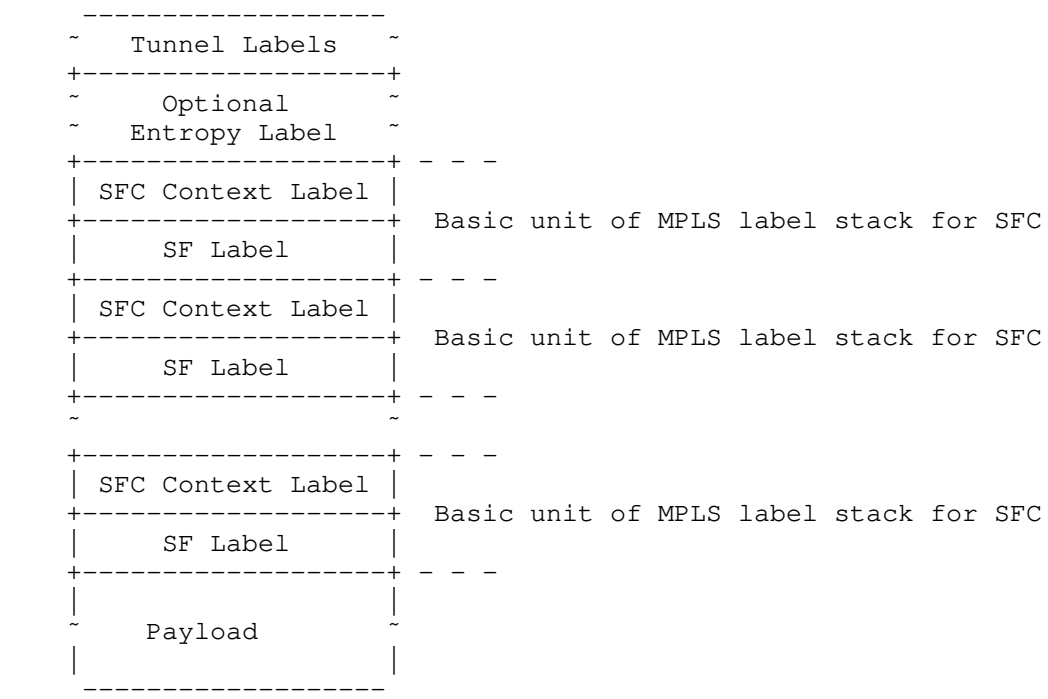


Figure 3: The MPLS SFC Label Stack for Label Stacking

The following processing rules apply to the Label fields:

- When a Classifier inserts a packet onto an SFP it adds a stack comprising one or more instances of the basic unit of MPLS label stack for SFC. Taken together, this stack defines the SFs to be actioned and so defines the SFP that the packet will traverse.

- When a component of the SFC system processes a packet it uses the top basic unit of label stack for SFC to determine to which SFI to next deliver the packet. When an SFF receives a packet it examines the top basic unit of MPLS label stack for SFC to determine where to send the packet next. If the next recipient is a local SFI, the SFC strips the basic unit of MPLS label stack for SFC before forwarding the packet.

7. Mixed Mode Forwarding

The previous sections describe homogeneous networks where SFC forwarding is either all label swapping or all label popping (stacking). But it is also possible that different parts of the network utilize swapping or popping. It is also worth noting that a Classifier may be content to use an SFP as installed in the network by a control plane or management plane and so would use label swapping, but that there may be a point in the SFP where a choice of SFIs can be made (perhaps for load balancing) and where, in this instance, the Classifier wishes to exert control over that choice by use of a specific entry on the label stack.

When an SFF receives a packet containing an MPLS label stack, it checks whether it is processing an {SFP, SI} label pair for label swapping or a {context label, SFI index} label pair for label stacking. It then selects the appropriate SFI to which to send the packet. When it receives the packet back from the SFI, it has four cases to consider.

- If the current hop requires an {SFP, SI} and the next hop requires an {SFP, SI}, it sets the SI label to the SI value of the current hop, selects an instance of the SF to be executed at the next hop, and tunnels the packet to the SFF for that SFI.

- If the current hop requires an {SFP, SI} and the next hop requires a {context label, SFI label}, it pops the {SFP, SI} from the top of the MPLS label stack and tunnels the packet to the SFF indicated by the context label.

- If the current hop requires a {context label, SFI label}, it pops the {context label, SFI label} from the top of the MPLS label stack.

  * If the new top of the MPLS label stack contains an {SFP, SI} label pair, it selects an SFI to use at the next hop, and tunnels the packet to SFF for that SFI.

* If the top of the MPLS label stack contains a {context label, SFI label}, it tunnels the packet to the SFF indicated by the context label.

8. A Note on Service Function Capabilities and SFC Proxies

The concept of an "SFC Proxy" is introduced in [RFC7665]. An SFC Proxy is logically located between an SFF and an SFI that is not "SFC-aware". Such SFIs are not capable of handling the SFC encapsulation (whether that be NSH or MPLS) and need the encapsulation stripped from the packets they are to process. In many cases, legacy SFIs that were once deployed as "bumps in the wire" fit into this category until they have been upgraded to be SFC-aware.

The job of an SFC Proxy is to remove and then reimpose SFC encapsulation so that the SFF is able to process as though it was communication with an SFC-aware SFI, and so that the SFI is unaware of the SFC encapsulation. In this regard, the job of an SFC Proxy is no different when NSH encapsulation is used and when MPLS encapsulation is used as described in this document, although (of course) it is different encapsulation bytes that must be removed and reimposed.

It should be noted that the SFC Proxy is a logical function. It could be implemented as a separate physical component on the path from the SFF to SFI, but it could be coresident with the SFF or it could be a component of the SFI. This is purely an implementation choice.

Note also that the delivery of metadata (see Section 11) requires specific processing if an SFC Proxy is in use. This is also no different when NSH or the MPLS encoding defined in this document is in use, and how it is handled will depend on how (or if) each non-SFC-aware SFI can receive metadata.

9. Control Plane Considerations

In order that a packet may be forwarded along an SFP several functional elements must be executed.

- Discovery/advertisement of SFIs.
- Computation of SFP.
- Programming of Classifiers.
- Advertisement of forwarding instructions.

Various approaches may be taken. These include a fully centralized model where SFFs report to a central controller the SFIs that they support, the central controller computes the SFP and programs the Classifiers, and (if the label swapping approach is taken) the central controller installs forwarding state in the SFFs that lie on the SFP.

Alternatively, a dynamic control plane may be used such as that described in [I-D.ietf-bess-nsh-bgp-control-plane]. In this case the SFFs use the control plane to advertise the SFIs that they support, a central controller computes the SFP and programs the Classifiers, and (if the label swapping approach is taken) the central controller uses the control plane to advertise the SFPs so that SFFs that lie on the SFP can install the necessary forwarding state.

10. Use of the Entropy Label

Entropy is used in ECMP situations to ensure that packets from the same flow travel down the same path, thus avoiding jitter or re-ordering issues within a flow.

Entropy is often determined by hashing on specific fields in a packet header such as the "five-tuple" in the IP and transport headers. However, when an MPLS label stack is present, the depth of the stack could be too large for some processors to correctly determine the entropy hash. This problem is addressed by the inclusion of an Entropy Label as described in [RFC6790].

When entropy is desired for packets as they are carried in MPLS tunnels over the underlay network, it is RECOMMENDED that an Entropy Label is included in the label stack immediately after the tunnel labels and before the SFC labels as shown in Figure 2 and Figure 3.

If an Entropy Label is present in an MPLS payload, it is RECOMMENDED that the initial Classifier use that value in an Entropy Label inserted in the label stack when the packet is forwarded (on the first tunnel) to the first SFF. In this case it is not necessary to remove the Entropy Label from the payload.

11. Metadata

Metadata is defined in [RFC7665] as providing "the ability to exchange context information between classifiers and SFs, and among SFs." [RFC8300] defines how this context information can be directly encoded in fields that form part of the NSH encapsulation.

The next two sections describe how metadata is associated with user data packets, and how metadata may be exchanged between SFC nodes in

the network, when using an MPLS encoding of the logical representation of the NSH.

It should be noted that the MPLS encoding is slightly less functional than the direct use of the NSH. Both methods support metadata that is "per-SFP" or "per-packet-flow" (see [I-D.farrel-sfc-convent] for definitions of these terms), but "per-packet" metadata (where the metadata must be carried on each packet because it differs from one packet to the next even on the same flow or SFP) is only supported using the NSH and not using the mechanisms defined in this document.

11.1. Indicating Metadata in User Data Packets

Metadata is achieved in the MPLS realization of the logical NSH by the use of an SFC Metadata Label which uses the Extended Special Purpose Label construct [RFC7274]. Thus, three label stack entries are present as shown in Figure 4:

- The Extension Label (value 15)
- An extended special purpose label called the Metadata Label Indicator (MLI) (value TBD1 by IANA)
- The Metadata Label (ML).

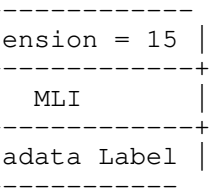


Figure 4: The MPLS SFC Metadata Label

The Metadata Label value is an index into a table of metadata that is programmed into the network using in-band or out-of-band mechanisms. Out-of-band mechanisms potentially include management plane and control plane solutions (such as [I-D.ietf-bess-nsh-bgp-control-plane]), but are out of scope for this document. The in-band mechanism is described in Section 11.2

The SFC Metadata Label (as a set of three labels as indicated in Figure 4) may be present zero, one, or more times in an MPLS SFC packet. For MPLS label swapping, the SFC Metadata Labels are placed immediately after the basic unit of MPLS label stack for SFC as shown

in Figure 5. For MPLS label stacking, the SFC Metadata Labels can be present zero, one, or more times and are placed at the bottom of the label stack as shown in Figure 6.

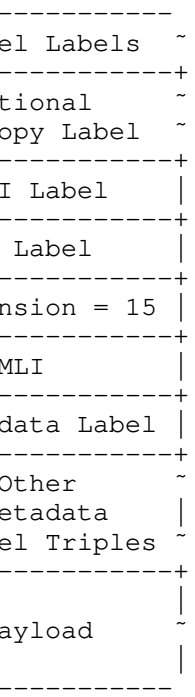


Figure 5: The MPLS SFC Label Stack for Label Swapping with Metadata Label

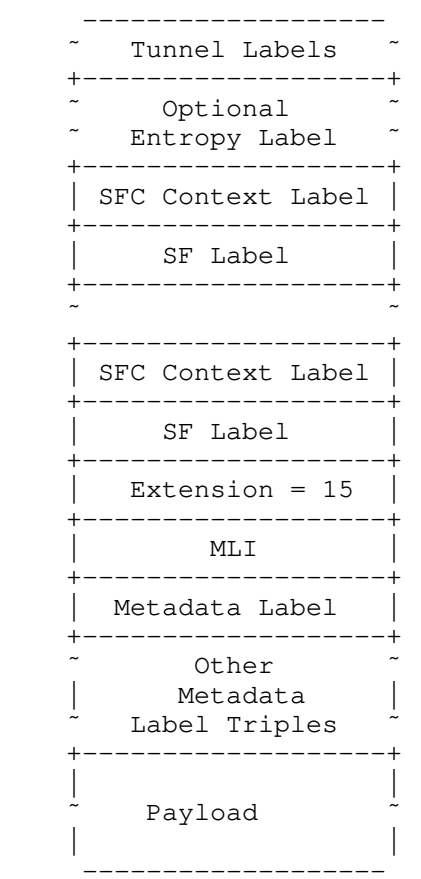


Figure 6: The MPLS SFC Label Stack for Label Stacking with Metadata Label

11.2. Inband Programming of Metadata

A mechanism for sending metadata associated with an SFP without a payload packet is described in [I-D.farrel-sfc-convent]. The same approach can be used in an MPLS network where the NSH is logically represented by an MPLS label stack.

The packet header is formed exactly as previously described in this document so that the packet will follow the SFP through the SFC network. However, instead of payload data, metadata is included after the bottom of the MPLS label stack. An Extended Special Purpose Label is used to indicate that the metadata is present. Thus, three label stack entries are present:

- The Extension Label (value 15)
- An extended special purpose label called the Metadata Present Indicator (MPI) (value TBD2 by IANA)
- The Metadata Label (ML) that is associated with this metadata on this SFP and can be used to indicate the use of the metadata as described in Section 11.

The SFC Metadata Present Label, if present, is placed immediately after the last basic unit of MPLS label stack for SFC. The resultant label stacks are shown in Figure 7 for the MPLS label swapping case and Figure 8 for the MPLS label stacking case.

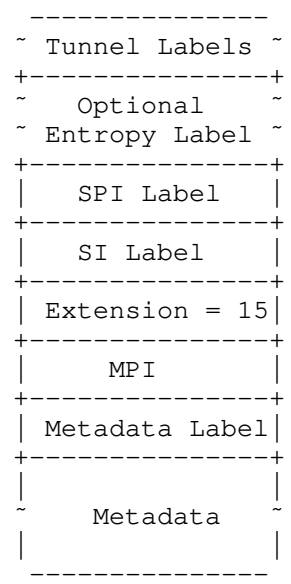


Figure 7: The MPLS SFC Label Stack for Label Swapping Carrying Metadata

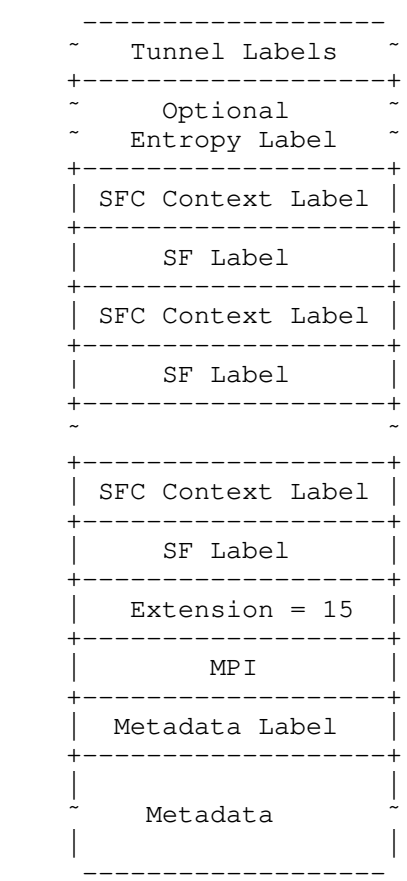


Figure 8: The MPLS SFC Label Stack for Label Stacking Carrying Metadata

In both cases the metadata is formatted as a TLV as shown in Figure 9.

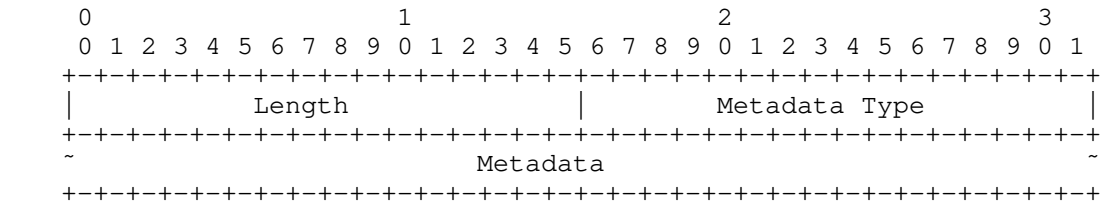


Figure 9: The Metadata TLV

The fields of this TLV are interpreted as follows:

Length: The length of the metadata carried in the Metadata field in octets not including any padding.

Metadata Type: The type of the metadata present. Values for this field are taken from the "MD Types" registry maintained by IANA and defined in [RFC8300].

Metadata: The actual metadata formatted as described in whatever document defines the metadata. This field is end-padded with zero to three octets of zeroes to take it up to a four octet boundary.

12. Worked Examples

Consider the simplistic MPLS SFC overlay network shown in Figure 10. A packet is classified for an SFP that will see it pass through two Service Functions, SFa and SFb, that are accessed through Service Function Forwarders SFFa and SFFb respectively. The packet is ultimately delivered to destination, D.

Let us assume that the SFP is computed and assigned the SPI of 239. The forwarding details of the SFP are distributed (perhaps using the mechanisms of [I-D.ietf-bess-nsh-bgp-control-plane]) so that the SFFs are programmed with the necessary forwarding instructions.

The packet progresses as follows:

a. The Classifier assigns the packet to the SFP and imposes two label stack entries comprising a single basic unit of MPLS SFC representation:

   * The higher label stack entry contains a label carrying the SPI value of 239.

   * The lower label stack entry contains a label carrying the SI value of 255.

Further labels may be imposed to tunnel the packet from the Classifier to SFFa.

b. When the packet arrives at SFFa it strips any labels associated with the tunnel that runs from the Classifier to SFFa. SFFa examines the top labels and matches the SPI/SI to identify that the packet should be forwarded to SFa. The packet is forwarded to SFa unmodified.

c. SFa performs its designated function and returns the packet to SFFa.

d. SFFa modifies the SI in the lower label stack entry (to 254) and uses the SPI/SI to look up the forwarding instructions. It sends the packet with two label stack entries:

   * The higher label stack entry contains a label carrying the SPI value of 239.

   * The lower label stack entry contains a label carrying the SI value of 254.

   Further labels may be imposed to tunnel the packet from the SFFa to SFFb.

e. When the packet arrives at SFFb it strips any labels associated with the tunnel from SFFa. SFFb examines the top labels and matches the SPI/SI to identify that the packet should be forwarded to SFb. The packet is forwarded to SFb unmodified.

f. SFb performs its designated function and returns the packet to SFFb.

g. SFFb modifies the SI in the lower label stack entry (to 253) and uses the SPI/SI to lookup up the forwarding instructions. It determines that it is the last SFF in the SFP so it strips the two SFC label stack entries and forwards the payload toward D using the payload protocol.

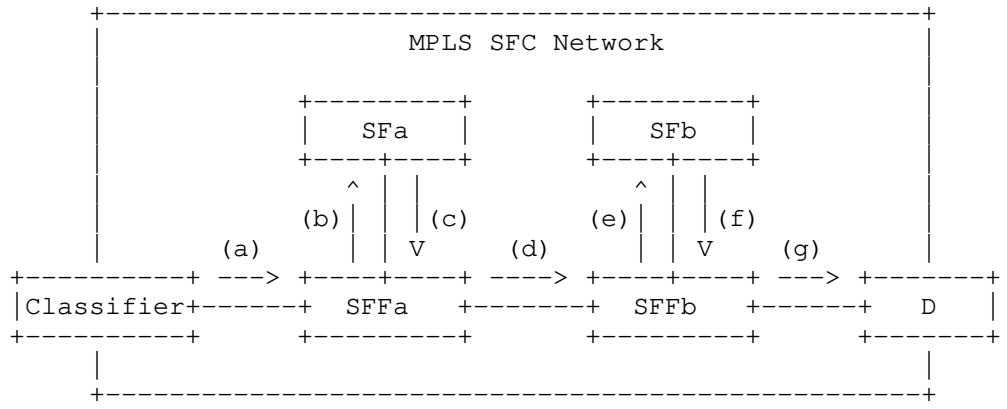


Figure 10: Service Function Chaining in an MPLS Network

Alternatively, consider the MPLS SFC overlay network shown in Figure 11. A packet is classified for an SFP that will see it pass through two Service Functions, SFx and SFy, that are accessed through Service Function Forwarders SFFx and SFFy respectively. The packet is ultimately delivered to destination, D.

Let us assume that the SFP is computed and assigned the SPI of 239. However, the forwarding state for the SFP is not distributed and installed in the network. Instead it will be attached to the individual packets using the MPLS label stack.

The packet progresses as follows:

1. The Classifier assigns the packet to the SFP and imposes two basic units of MPLS SFC representation to describe the full SFP:

   * The top basic unit comprises two label stack entries as follows:

     + The higher label stack entry contains a label carrying the SFC context.

     + The lower label stack entry contains a label carrying the SF indicator for SFx.

   * The lower basic unit comprises two label stack entries as follows:

     + The higher label stack entry contains a label carrying the SFC context.

+ The lower label stack entry contains a label carrying the SF indicator for SFy.

Further labels may be imposed to tunnel the packet from the Classifier to SFFx.

2. When the packet arrives at SFFx it strips any labels associated with the tunnel from the Classifier. SFFx examines the top labels and matches the context/SF values to identify that the packet should be forwarded to SFx. The packet is forwarded to SFx unmodified.

3. SFx performs its designated function and returns the packet to SFFx.

4. SFFx strips the top basic unit of MPLS SFC representation revealing the next basic unit. It then uses the revealed context/SF values to determine how to route the packet to the next SFF, SFFy. It sends the packet with just one basic unit of MPLS SFC representation comprising two label stack entries:

   * The higher label stack entry contains a label carrying the SFC context.

   * The lower label stack entry contains a label carrying the SF indicator for SFy.

   Further labels may be imposed to tunnel the packet from the SFFx to SFFy.

5. When the packet arrives at SFFy it strips any labels associated with the tunnel from SFFx. SFFy examines the top labels and matches the context/SF values to identify that the packet should be forwarded to SFy. The packet is forwarded to SFy unmodified.

6. SFy performs its designated function and returns the packet to SFFy.

7. SFFy strips the top basic unit of MPLS SFC representation revealing the payload packet. It forwards the payload toward D using the payload protocol.

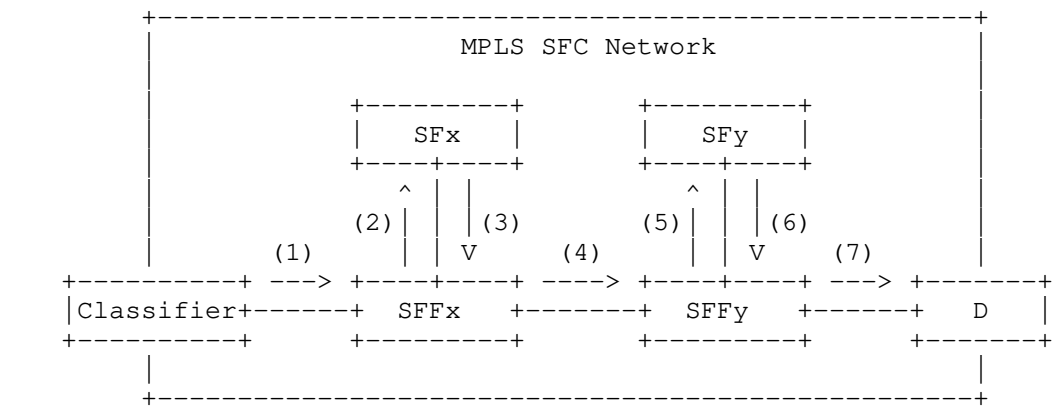


Figure 11: Service Function Chaining Using MPLS Label Stacking

13. Security Considerations

Discussion of the security properties of SFC networks can be found in [RFC7665]. Further security discussion for the NSH and its use is present in [RFC8300].

It is fundamental to the SFC design that the classifier is a trusted resource which determines the processing that the packet will be subject to, including for example the firewall. It is also fundamental to the MPLS design that packets are routed through the network using the path specified by the node imposing the labels, and that labels are swapped or popped correctly. Where an SF is not encapsulation aware the encapsulation may be stripped by an SFC proxy such that packet may exist as a native packet (perhaps IP) on the path between SFC proxy and SF, however this is an intrinsic part of the SFC design which needs to define how a packet is protected in that environment.

Additionally, where a tunnel is used to link two non-MPLS domains, the tunnel design needs to specify how the tunnel is secured.

Thus the security vulnerabilities are addressed (or should be addressed) in all the underlying technologies used by this design, which itself does not introduce any new security vulnerabilities.

14. IANA Considerations

This document requests IANA to make allocations from the "Extended Special-Purpose MPLS Label Values" subregistry of the "Special-

Purpose Multiprotocol Label Switching (MPLS) Label Values" registry as follows:

Value	Description	
TBD1	Metadata Label Indicator (MLI)	[This.I-D]
TBD2	Metadata Present Indicator (MPI)	[This.I-D]

15. Acknowledgements

This document derives ideas and text from [I-D.ietf-bess-nsh-bgp-control-plane].

The authors are grateful to all those who contributed to the discussions that led to this work: Loa Andersson, Andrew G. Malis, Alexander Vainshtein, Joel M. Halpern, Tony Przygienda, Stuart Mackie, Keyur Patel, and Jim Guichard. Loa Andersson provided helpful review comments.

Thanks to Loa Andersson, Lizhong Jin, Matthew Bocci, and Mach Chen for reviews of this text.

16. References

16.1. Normative References

[I-D.farrel-sfc-convent] Farrel, A. and J. Drake, "Operating the Network Service Header (NSH) with Next Protocol "None"", draft-farrel-sfc-convent-06 (work in progress), February 2018.

[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>.

[RFC7274] Kompella, K., Andersson, L., and A. Farrel, "Allocating and Retiring Special-Purpose MPLS Labels", RFC 7274, DOI 10.17487/RFC7274, June 2014, <https://www.rfc-editor.org/info/rfc7274>.

[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>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

16.2. Informative References

[I-D.ietf-bess-nsh-bgp-control-plane] Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for NSH SFC", draft-ietf-bess-nsh-bgp-control-plane-03 (work in progress), March 2018.

[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>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <https://www.rfc-editor.org/info/rfc6790>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Adrian Farrel
Juniper Networks

Email: afarrel@juniper.net

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

John Drake
Juniper Networks

Email: jdrake@juniper.net

SFC WG
Internet-Draft
Intended status: Informational
Expires: May 1, 2018

R. Khalili
Z. Despotovic
A. Hecker
Huawei ERC, Munich, Germany
D. Purkayastha
A. Rahman
D. Trossen
InterDigital Communications, LLC

October 28, 2017

Optimized Service Function Chaining
draft-khalili-optimized-service-function-chaining-00

Abstract

This draft investigates possibilities to use so-called 'transport-derived service function forwarders' (tSFFs) that ignore the SFC encapsulation, using instead existing transport information for explicit service path information. The draft discusses two such possibilities. In the first one, the transport network is SDN-based. The second one introduces and explains a specific service request routing (SRR) function to support URL-level routing of service requests.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Table of Contents

1 Introduction

The delivery of end-to-end network services often requires steering traffic through a sequence of individual service functions. Deployment of these functions and particularly creation of a composite service from them, i.e. steering traffic through them, had traditionally been coupled to the underlying network topology and as such awkward to configure and operate [RFC7498].

To remedy the problems identified by [RFC7498], [RFC7665] defines architecture for service function chaining that is topology independent. The architecture is predicated on a service indirection layer composed of architectural elements such as service functions (SF), service function forwarders (SFF), classifiers, etc. SFFs are the key architectural element as they connect the attached SFs and thus create a service plane.

[RFC7665] proposes SFC encapsulation as a means for service plane elements to communicate. The SFC encapsulation serves essentially two purposes. It provides path identification in the service plane (which is the primary and mandatory usage of the encapsulation) and serves as a placeholder for metadata transferred among SFs. [Quinn2017] defines NSH as a particular realization of the SFC encapsulation.

Standalone SFC encapsulation such as NSH is the mainstream SFC forwarding method with the intention to work over multiple (possibly inter-domain) transport networks. However, SFC has been identified as a suitable methodology to chain services even within single transport networks or, as outlined in [Kumar2017], even in data centers. In such cases, [RFC7665] points at the possibility of utilizing so-called 'transport-derived service function forwarders' (tSFFs) that ignore the SFC encapsulation, using existing transport information for explicit service path information.

In this document, we expand on this possibility by focusing on the realization of efficient chaining over a single transport network. In our first solution, said transport network is an SDN-based one where we represent the SFP (service function path) through a vector of aggregated flow identifiers. This solution is positioned as a tSFF between two or more SFs with no need for this solution to be SFC encapsulation aware. In our second solution, we refer to [Purka2017] which uses a specific service request routing (SRR) function to support URL-level routing of service requests. Chaining more than one SRR-connected SFs can be optimized for reducing the initial request latency, while supporting at least three different tSFFs, including the flow aggregation one presented as the first solution.

1.1 Terminology

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].

2 SFC Forwarding Solutions

2.1 Edge classification and network forwarding aggregation

Assume we are free to choose network locators (routable addresses in the considered network) for edge nodes in a network. Besides, assume that routers (switches in SDN terminology) in that network can forward packets based on wildcard matching on bit-fields in the destination address. For example, a switch somewhere in the network can forward a packet by following this logic: "the packet should be sent out the port k, because the bits 15, 16 and 17 of the destination address are 1, 0 and 1, respectively." This is possible with SDN deployments compliant e.g. with OpenFlow v1.3 and higher.

One can then come up with a multi-level classification of edge nodes, which leads to an assignment of locators to the edge nodes such that for every switch of the network the following holds:

> The switch has as many forwarding rules as it has ports
>
> For switch port k, the rule takes the form: when the destination address of the incoming packet contains a bit-field of a specific form, forward the packet to port k . For example, if the packet has 1 in the bit p of the destination address, forward to port 4.

When this is done, the network essentially becomes a fabric that delivers a packet arriving at one of its inports to the appropriate outport. It does that while maintaining the minimum internal state. [Khalili2017] explains details of the approach. In particular, it shows that large networks and networks with particular topologies require a large ID space. With that in mind, [Khalili2017] proposes an approximate method that trades node state for ID (address) space and shows that a small increase of the node state brings a large reduction or the address space (additional forwarding rules that don’t follow the above form). It is this approximate method that we refer to in the rest of this section.

2.1.1 Example

Consider a simple network with an ingress (classifier) and an egress node, two transport switches/routers C1 and C2, and two service function forwarders, SFF1 and SFF2 (as depicted in Figure 1). Service functions SF1 and SF2 are attached to SFF1 and SF3 and SF4 are attached to SFF2. In this example, we assume that edge nodes are SFF1, SFF2, and the egress node.

The ASC algorithm proposed in [Khalili2017] assigns to an edge node in the network an ID of the form (v(1), v(2), ..., v(K)), where v(j), j\in[1, K], being 1 if there is a path crossing link j that ends in the corresponding edge node, and 0 otherwise. K is the size of IDs assigned to edge nodes and is an output of the algorithm.

Applying ASC algorithm to our example, we have IDSFF1 = (0, 1, 1, 0, 0), IDSFF2 = (1, 0, 0, 1, 0), and IDEgress = (1, 0, 0, 0, 1). Assuming that the destination-edge IDs are embedded in the header of the packets, e.g. via encapsulation, the forwarding rules at C1 and C2 can be aggregated by matching on bits of these IDs:

> At C1: if 1st bit is 1, forward over port 3; if 2nd bit is 1, forward over port 2.
>
> At C2: if 3rd bit is 1, forward over port 1, if 4th bit is 1, forward over port 2, if 5th bit is 1, forward over port 3.

Note from this example that each edge node has a unique ID and that we put no limitation on how SFCs are defined.

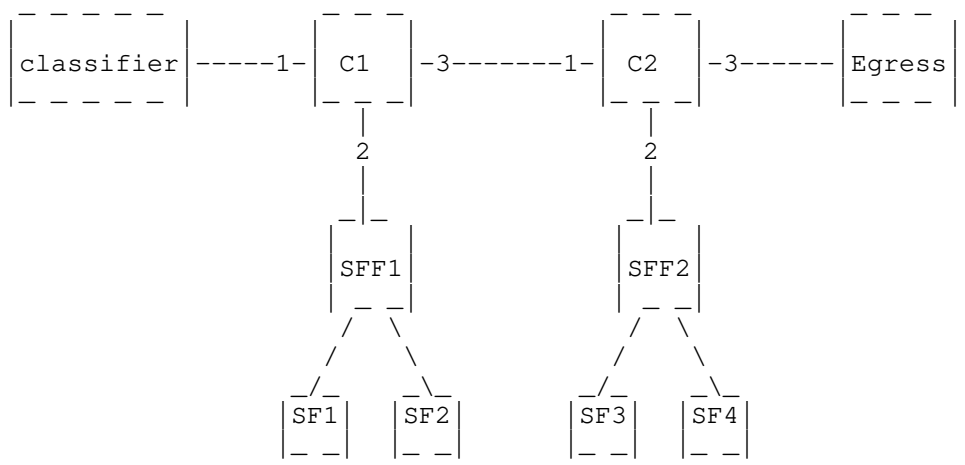


Figure 1: A simple topology with two SFFs and two transport switches/routers.

2.2 SRR

In [Purka2017], an extension to the Service Function Chaining (SFC)

concept is being proposed for a flexible chaining of service functions in an SFC environment, where a number of virtual instances for a single service function might exist. Hence, instead of explicitly (re-)chaining a given SFC in order to utilize a new virtual instance for an existing SF, a special service function called SRR (service request routing) is utilized to direct the requests via a URL-based abstraction (here, www.foo.com) for the SF address. As a first step, the work in [Purka2017] proposes to extend the notion of the service function path (SFP) to include such URLs in addition to already defined Ethernet or IP addresses. This is shown in Figure 1. Here the SFP includes the URLs of the service functions 1 to N (i.e., www.foo.com to www.fooN.com) as well as link-local IP addresses being used for forwarding at the local access (here shown as simple 192.168.x.x IP addresses). The creation of a suitable SFP is assumed to be part of an orchestration process, which is not within the scope of the SFC framework per se.

The SRR service function in Figure 2 can be further divided into sub-functions for realizing the dynamic chaining capabilities, as shown in [Purka2017]. Here, the service functions (such as clients and SF1 in Figure 2) communicate with local NAPs (network attachment points), while the latter communicate with the PCE (path computation element) to realize the IP and HTTP-level communication. In this case, the incoming NAP is denoted as the client NAP (cNAP) and the outgoing NAP as server NAP (sNAP). The Layer 2 transport is realized via the tSFF1 function (transport-derived service function forwarder). Here we assume that each service function is connected to an own NAP (via link-local IP communication) although one or more service functions could also reside at a single NAP.

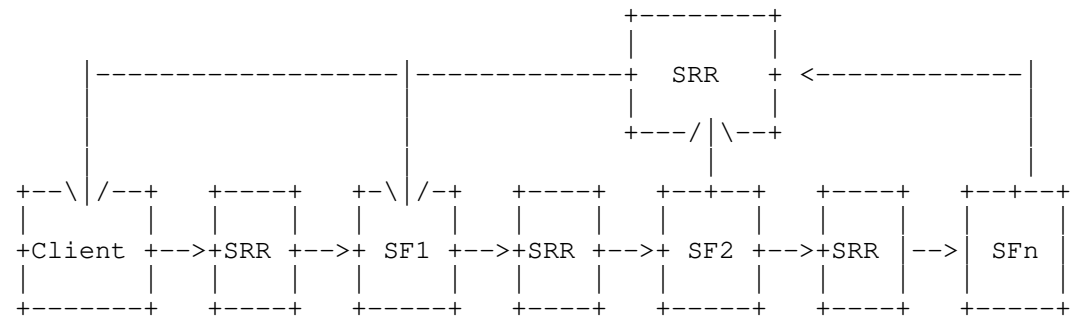


Figure 2: Dynamic Chaining SFC, as proposed in [Purka2017]. SFP: 192.168.x.x -> www.foo.com -> 192.168.x.x -> www.foo2.com -> ... -> www.fooN.com

As presented in [Purka2017], the hierarchical addressing presented in

Section 3.1.1 can be utilized for the realization of said tSFF1, while other realizations could utilize SDN-based transport networks or a BIER routing layer [Wijnands2017]. With this, the SRR service function is placed in-between specific tSFFs (the three aforementioned ones) and general service functions to be chained.

3 Optimized SFC Chaining

3.1 Utilizing Transport-derived SFFs

Our model retains the architectural behavior of the SFC architecture of [RFC7665]. Yet, the SFC and the transport encapsulation are merged into the transport header. Thus everything, both transport and service plane forwarding, is happening based on transport encapsulation bits. The model builds on the edge node classification presented in Section 2.1 and comes in two flavors. The first one (Section 3.1.1) treats SFFs as edge nodes. The second one (Section 3.1.2) assigns fictitious edge nodes to entire service chains. In both cases, the key points are how we identify the service chains, and related to that, how we embed these identifiers into the available address space.

3.1.1 Hierarchical addressing for service chaining

This approach treats SFFs as edge nodes. The set of SFFs, as points of attachment of SFs, is normally static, known in advance in a network. In that sense, SFFs do not impose any stronger requirements than edge nodes, so the approach presented next looks viable.

The hierarchical service chain addressing works with the address structure (IDSFF.IDSF.IDChain), in which IDSFF identifies an SFF in the network, IDSF identifies an SF attached to that SFF, while IDChain refers to a specific chain handled by that SF. (Note that SFs can be attached to multiple SFFs, i.e. the approach is not limiting in this sense. It is rather obvious that multiple SFs can be attached to a single SFF.)

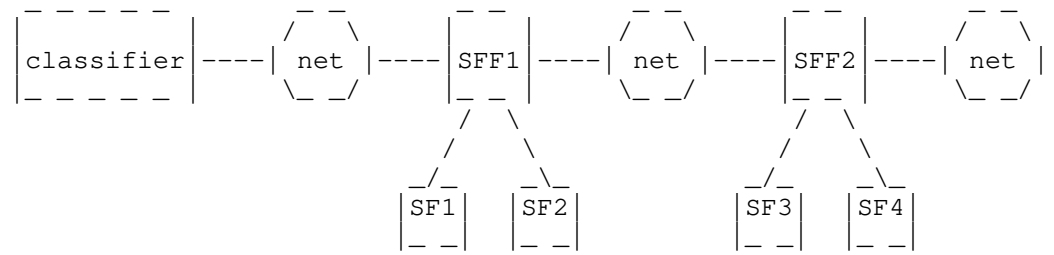


Figure 3: a service chain of SF1-SF2-SF3 is considered in this example.

See Figure 2 for an example. Assume that the edge nodes in the shown

network are SFF1 and SFF2 (with possibly many other nodes which are not shown), while the service functions SF1, ..., SF4 are considered end nodes, i.e. they are not edge nodes as such do not underlie classification. (Note that this will be changed in Section 3.1.2.). Assume that the classification yields the locators (IDs) IDSFF1 and IDSFF2 for SFF1, respectively SFF2, and that a service chain SF1-SF2-SF3 has the same identifier IDChain1 at each SF (SF1, SF2, SF3). This is just for simplicity, these IDs can be different at different SFs. The service chain SF1-SF2-SF3 can operate as follows.

The classifier first adds the outer transport header with the destination address (IDSFF1.IDSF1.IDChain1). The network uses the IDSFF1 bitfield to route the packet to SFF1. SFF1 uses the middle part of the address, IDSF1, to deliver the packet to SF1. SF1, being SFC-aware, strips off the transport header and saves it, then processes the packet and, after restoring the saved transport header, sends it back to SFF1. SFF1 changes the transport header destination address to (IDSFF1.IDSF2.IDchain1) and forwards the packet to SF2. SF2 performs similar steps as SF1 and returns the packet to SFF1. SFF1 changes the transport header to (IDSFF2.IDSF3.IDChain1) and sends the packet towards SFF2. (In an SDN network, switches can manipulate with the headers by means of suitable flow rules, which should match on the (IDSF.IDChain) fraction of the destination address. A second pass through the SDN processing stack will select the appropriate port to send the packet towards SFF2.) SFF2 performs the very same sequence of steps to deliver the packet to the correct SF and then further to the network.

Note the role of the (IDChain) part of the address. That tag serves to differentiate between different service chains that pass a single SF. For example, if in addition to SF1-SF3 there is a service chain SF1-SF5, where SF5 is attached to SFF3, SFF1 will use the (IDChain) to forward packets coming from SF1.

3.1.2 Edge classification and service chains

Continuing with the classification discussion from Section 3.1.1, let us assign a fictitious edge node to a service chain under consideration. More precisely, let us assign one such node to every subsequence of the chain that starts at each possible position in the chain and goes until its end. For example, for a chain SF1-SF2-SF3, define three such nodes for sub-chains SF1-SF2-SF3, SF2-SF3 and SF3. Let locators of these fictitious edge nodes be the SFs that start the corresponding sub-chains. So, in the example, the locators are SF1, SF2 and SF3. If we had another chain that goes over SF1, then we would simply add another node, say SF1', and attach it to SFF1, next to SF1. This is to indicate that we need a distinct locator for each chain that goes over SF1. So we now have the starting network and

additional imaginary edge nodes which topologically coincide with existing service functions but require additional, separate classification vectors.

Assume that, after the classification as described in Section 3.1.1, we generate locators (classification vectors) IDSF1, IDSF2 and IDSF3 for the chain SF1-SF2-SF3 and setup rules (e.g. OpenFlow compliant) that:

At SFF1:

Forward to SF1 packets with destination IDSF1, that come from the network.

Replace IDSF1 with IDSF2 and then forward to SF2 packet that arrive from SF1.

Replace IDSF2 with IDSF3 and then forward to SFF2.

At SFF2:

Forward to SF3 packets with destination IDSF3 that come from the network.

We can distinguish between the packets that are received from the network and those received from SF1 by using the inport information.

3.1.3 Dynamic addition of service chains

The method just described assumes that all service chains are known in advance, before the classification. That assumption is not realistic, i.e. presents a strong, undesired constraint. This section will in a future version discuss how we relax that assumption, how we handle dynamic additions of service chains, etc.

3.2 Pre-Warming SFP Information for SRR-based Chaining

One issue when chaining service functions utilizing the SRR function is the initial delay incurred through the necessary path computation for a new service segment along the overall service function path. For instance, when the service function 'client' residing at the first SRR in Figure 1 issues a request to foo.com, i.e., the URL for the second service function, the NAP sub-function will trigger a PCE request for path resolution within the Layer 2 transport network. Such PCE request incurs said delay for the initial request while all subsequent requests along the same path are likely going to use locally cached information at the SRR function (we here assume but do

not detail suitable path information update procedures being implemented by the SRR sub-functions in case of path changes to another service function).

It is reasonable to assume that SFPs can be established across the realm of more than one PCE, e.g., each administering one administrative domain. However, in the case of a single PCE across a number of SRR functions, Figure 2 can be redrawn as follows.

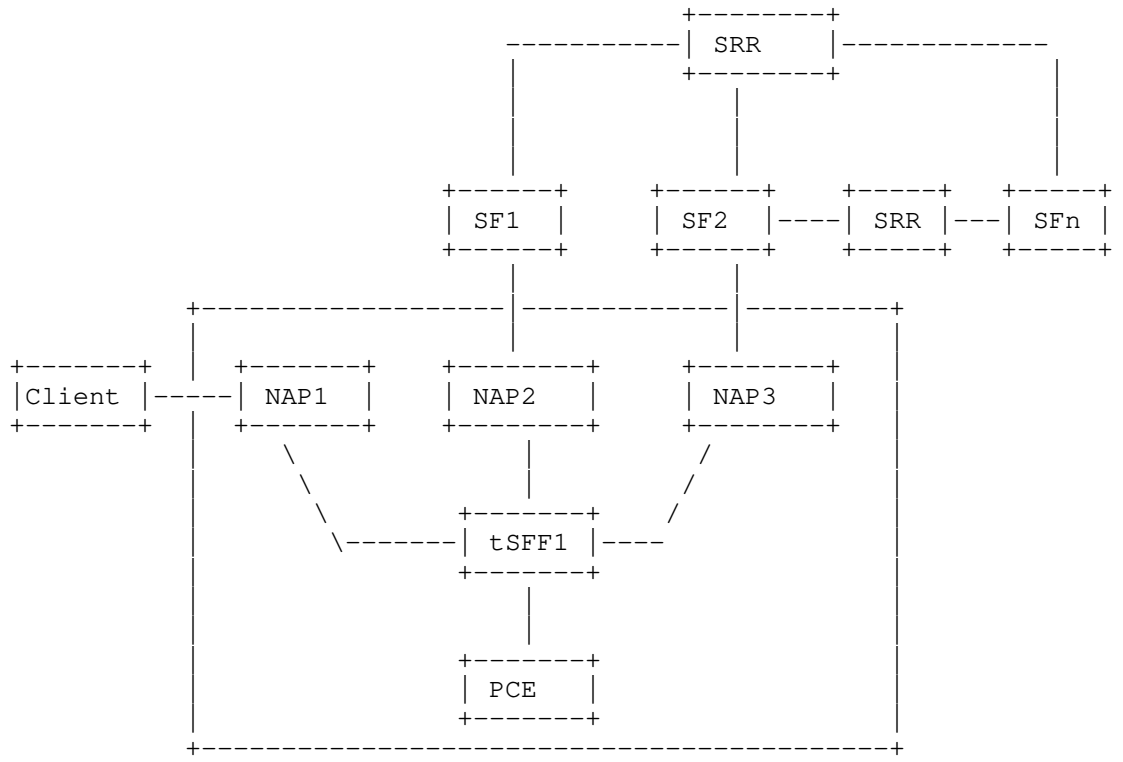


Figure 4. Decomposed Dynamic Chaining SFC across two or more SFCs.

Here, two SRR functions utilize the same PCE, e.g., within a single transport network. In this case, we propose to reduce such initial

chaining delay by virtue of a 'pre-warming' of the SRR sub-functions, specifically the incoming NAP at the suitable SRR along the SFP. For this, we require a communication of the NSH and therefore the SFP information to the PCE - such communication is subject to a standardized protocol based on a trigger that led to the formation of said SFP, as shown in Figure 4. Once such SFP information has been received by the PCE, it then executes the following procedure.

FOR ALL SF requests routed via an SRR served by the PCE:

1. Determine the incoming NAP of the first SF request, e.g., 192.168.x.x in Figure 2.

2. Determine the outgoing NAP of the service endpoint address at the outgoing SF, e.g., www.foo.com in Figure 2.

3. Compute path between incoming NAP and outgoing NAP - path computation might include a policy constraint, such as shortest path or shortest delay.

4. Deliver path information to incoming NAP.

END FOR

Figure 5 outlines the messages being exchanged between the joint PCE and the various NAPs of the SRR function. The exact nature of the messages is subject to standardization and not shown at this stage of the draft.

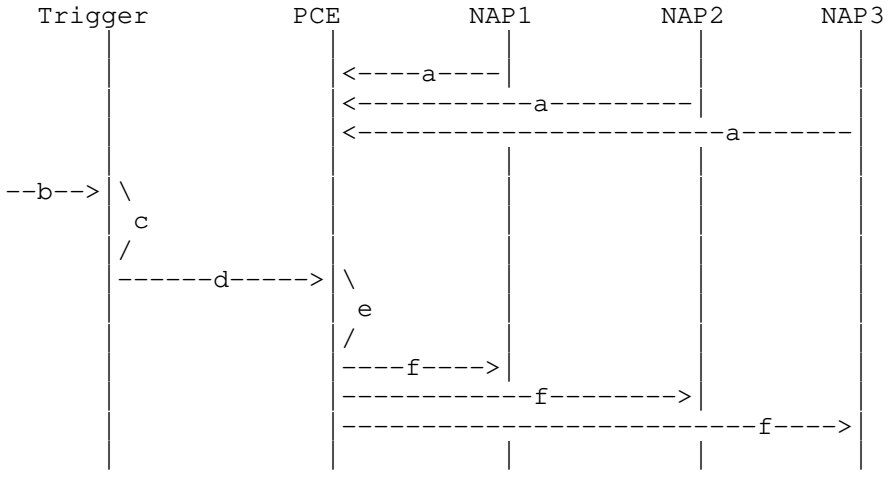


Figure 5. Message Sequence Chart Resulting in Pre-Warming of Routing Entries. a) subscribe to pre-warming information, b)initiate service chaining based on external mgmt. trigger, c) compute SFP, d) send SFP, e) map SFP information onto paths from incoming to ongoing NAPs, f) push path information with forwarding/path identifier and URL.

4 Applicability

This draft investigates whether transport encapsulation can be used for service function chaining. The main message it delivers is that this seems possible. This was demonstrated on an example of underlying SDN network.

However, we are not normative here with respect to what transport encapsulation and which bits thereof are used for service function chaining, i.e. which existing transport encapsulations give us the needed features (e.g. said assignment of transport identifiers and their handling at transport nodes) to successfully incorporate service chaining. This will be a subject of future investigations.

5 Discussion

Transport-derived SFC forwarding is related to a number of advantages. In particular, easier deployment of service chaining, as SFs and SFFs in a transport-derived chaining do not have to be SFC encapsulation aware. Further discussion will be included in future version on specific advantages and downsides of individual investigated methods, from Section 2 and Section 3.

6 Informative References

[RFC7498] P. Quinn, et al., "Problem Statement for Service Function Chaining", RFC 7498 (INFORMATIONAL), April 2015.

[RFC7665] Joel Halpern, et al., "Service Function Chaining (SFC) Architecture", RFC 7665 (INFORMATIONAL), October 2015.

[Quinn2017] P. Quinn, et al., "Network Service Header", IETF draft, draft-ietf-sfc-nsh-27 (work in progress), October 2017.

[Kumar2017] S. Kumar, et al., "Service Function Chaining Use Cases In Data Centers", IETF draft, draft-ietf-sfc-dc-use-cases-06 (work in progress), February 2017.

[Khalili2016] R. Khalili, et al., "Reducing State of OpenFlow Switches in Mobile Core Networks by Flow Rule Aggregation" IEEE ICCCN 2016.

[Purka2017] D. Purkayastha, et al., "Use Case for Handling of Dynamic Chaining and Service Indirection", IETF draft, draft-purkayastha-sfc-service-indirection-00 (work in progress), July 2017

[Wijnands2017] IJ. Wijnands, et al., "Multicast using Bit Index Explicit Replication", IETF draft, draft-ietf-bier-architecture-08 (work in progress), September 2017

Authors' Addresses

Ramin Khalili
Huawei ERC, Munich, Germany
Email: Ramin.khalili@huawei.com

Zoran Despotovic
Huawei ERC, Munich, Germany
Email: Zoran.Despotovic@huawei.com

Artur Hecker
Huawei ERC, Munich, Germany
Email: Artur.Hecker@huawei.com

Debashish Purkayastha
InterDigital Communications, LLC

Conchoken, USA
Email: Debashish.Purkayastha@InterDigital.com

Akbar Rahman
InterDigital Communications, LLC
Montreal, Canada
Email: Akbar.Rahman@InterDigital.com

Dirk Trossen
InterDigital Communications, LLC
London, United Kingdom
Email: Dirk.Trossen@InterDigital.com

SFC Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 16, 2018

G. Mirsky
ZTE Corp.
G. Fioccola
Telecom Italia
T. Mizrahi
Marvell

September 12, 2017

Performance Measurement (PM) with Alternate Marking Method in Service Function Chaining (SFC) Domain
draft-mirsky-sfc-pmamm-02

Abstract

This document describes how the alternate marking method be used as the passive performance measurement method in a Service Function Chaining (SFC) domain.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 16, 2018.

Copyright Notice

Table of Contents

1. Introduction

[RFC7665] introduced architecture of a Service Function Chain (SFC) in the network and defined its components as classifier, Service Function Forwarder (SFF), and Service Function (SF). [I-D.ietf-ippm-alt-mark] describes passive performance measurement method, which can be used to measure packet loss, latency and jitter on live traffic. Because this method is based on marking consecutive batches of packets the method often referred as Alternate Marking Method (AMM).

This document defines how the alternate marking method can be used to measure packet loss and delay metrics of a service flow over e2e or any segment of the SFC.

2. Conventions used in this document

2.1. Terminology

MM: Marking Method

OAM: Operations, Administration and Maintenance

SFC: Service Function Chain

SF: Service Function

SFF: Service Function Forwarder

SFP: Service Function Path

NSH: Network Service Header

2.2. Requirements Language

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

3. Mark Field in NSH Base Header

[I-D.ietf-sfc-nsh] defines format of the Network Service Header (NSH).

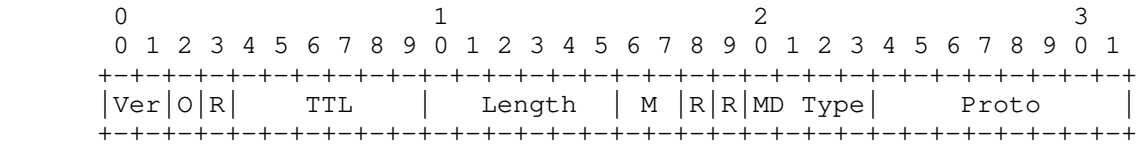


Figure 1: NSH Base format

This document defines two bit long field, referred as Mark field (M in Figure 1, as part of NSH Base and designated for the alternate marking performance measurement method [I-D.ietf-ippm-alt-mark]. The Mark field MUST NOT be used in defining forwarding and/or quality of service treatment of a SFC packet. The Mark field MUST be used only for the performance measurement of data traffic in SFC layer. Because setting of the field to any value does not affect forwarding and/or quality of service treatment of a packet, the alternate marking method in SFC layer can be viewed as true example of passive performance measurement method.

The Figure 2 displays format of the Mark field.

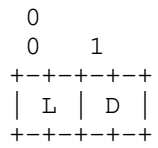


Figure 2: Mark field format

where:

- L- Loss flag;
- D - Delay flag.

4. Theory of Operation

The marking method can be successfully used in the SFC. Without limiting any generality consider SFC presented in Figure 3. Any combination of markings, Loss and/or Delay, can be applied to a service flow by any component of the SFC at either ingress or egress point to perform node, link, segment or end-to-end measurement to detect performance degradation defect and localize it efficiently.

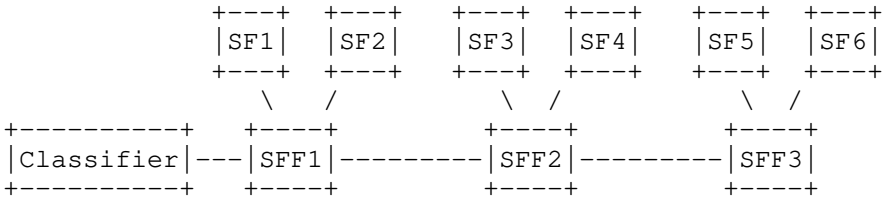


Figure 3: SFC network

Using the marking method a component of the SFC creates distinct sub-flows in the particular service traffic over SFC. Each sub-flow consists of consecutive blocks that are unambiguously recognizable by a monitoring point at any component of the SFC and can be measured to calculate packet loss and/or packet delay metrics.

4.1. Single Mark Enabled Measurement

As explained in the [I-D.ietf-ippm-alt-mark], marking can be applied to delineate blocks of packets based either on equal number of packets in a block or based on equal time interval. The latter method offers better control as it allows better account for capabilities of downstream nodes to report statistics related to

batches of packets and, at the same time, time resolution that affects defect detection interval.

If the Single Mark measurement used, then the Delay flag Figure 2 MUST be set to zero on transmit and ignored on reception by monitoring point.

The Loss flag is used to create alternate flows to measure the packet loss by switching value of the Loss flag every N-th packet or at certain time intervals. Delay metrics MAY be calculated with the alternate flow using any of the following methods:

- First/Last Packet Delay calculation: whenever the marking, i.e. value of Loss flag, changes a component of the SFC can store the timestamp of the first/last packet of the block. The timestamp can be compared with the timestamp of the packet that arrived in the same order through a monitoring point at downstream component of the SFC to compute packet delay. Because timestamps collected based on order of arrival this method is sensitive to packet loss and re-ordering of packets

- Average Packet Delay calculation: an average delay is calculated by considering the average arrival time of the packets within a single block. A component of the SFC may collect timestamps for each packet received within a single block. Average of the timestamp is the sum of all the timestamps divided by the total number of packets received. Then difference between averages calculated at two monitoring points is the average packet delay on that segment. This method is robust to out of order packets and also to packet loss (only a small error is introduced). This method only provides single metric for the duration of the block and it doesn't give the minimum and maximum delay values. This limitation could be overcome by reducing the duration of the block by means of an highly optimized implementation of the method.

4.2. Double Mark Enabled Measurement

Double Mark method allows measurement of minimum and maximum delays for the monitored flow but it requires more nodal and network resources. If the Double Mark method used, then the Loss flag MUST be used to create the alternate flow, i.e. mark larger batches of packets. The Delay flag MUST be used to mark single packets to measure delay jitter.

The first marking (Loss flag alternation) is needed for packet loss and also for average delay measurement. The second marking (Delay flag is put to one) creates a new set of marked packets that are fully identified over the SFC, so that a component can store the

timestamps of these packets; these timestamps can be compared with the timestamps of the same packets on another component of the SFC to compute packet delay values for each packet. The number of measurements can be easily increased by changing the frequency of the second marking. But the frequency of the second marking must be not too high in order to avoid out of order issues. This method is useful to have not only the average delay but also the minimum and maximum delay values and, in wider terms, to know more about the statistic distribution of delay values.

4.3. Residence Time Measurement with the Alternate Marking Method

Residence time is the variable part of the propagation delay that a packet experiences traversing a network, e.g. SFC. Residence Time over an SFC is the sum of the nodal residence times, i.e. periods that the packet spent in each of SFFs that compose the SFC. The nodal residence time in SFC itself is the sum of sub-nodal residence times that the packet spent in each of SFs that are part of the given SFC and are mapped to the SFF. The residence time and deviation of the residence time metrics may include any combination of minimum, maximum, values over measurement period, as well as mean, median, percentile. These metrics may be used to evaluate performance of the SFC and its elements before and during its operation.

Use of the specially marked packets simplifies residence time measurement and correlation of the measured metrics over the SFC end-to-end. For example, the alternate marking method may be used as described in Section 4.2 to identify packets in the data flow to be used to measure the residence time. The nodal and sub-nodal residence time metrics can be locally calculated and then collected using either in-band or out-band OAM mechanisms.

5. IANA Considerations

5.1. Mark Field in NSH Base Header

This document requests IANA to allocate Mark field as two bits-long field from NSH Base Header Reserved Bits [I-D.ietf-sfc-nsh].

This document requests IANA to register values of the Mark field of NSH as the following:

Bit Position	Marking	Description	Reference
0	S	Single Mark Measurement	This document
1	D	Double Mark Measurement	This document

Table 1: Mark field of SFC NSH

6. Security Considerations

This document lists the OAM requirement for SFC domain and does not raise any security concerns or issues in addition to ones common to networking and SFC.

7. Acknowledgement

TBD

8. References

8.1. Normative References

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-20 (work in progress), September 2017.

[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>.

[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>.

8.2. Informative References

[I-D.ietf-ippm-alt-mark]
Fioccola, G., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate Marking method for passive and hybrid performance monitoring", draft-ietf-ippm-alt-mark-10 (work in progress), September 2017.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Giuseppe Fioccola
Telecom Italia

Email: giuseppe.fioccola@telecomitalia.it

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam
Israel

Email: talmi@marvell.com

SFC Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 13, 2019

G. Mirsky
ZTE Corp.
G. Fioccola
Huawei Technologies
T. Mizrahi
Huawei Network.IO Innovation Lab

October 10, 2018

Performance Measurement (PM) with Alternate Marking Method in Service Function Chaining (SFC) Domain
draft-mirsky-sfc-pmamm-06

Abstract

This document describes how the alternate marking method be used as the passive performance measurement method in a Service Function Chaining (SFC) domain.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 13, 2019.

Copyright Notice

Table of Contents

1. Introduction

[RFC7665] introduced architecture of a Service Function Chain (SFC) in the network and defined its components as classifier, Service Function Forwarder (SFF), and Service Function (SF). [RFC8321] describes the passive performance measurement method, which can be used to measure packet loss, latency, and jitter on live traffic. Because this method is based on marking consecutive batches of packets the method often referred to as Alternate Marking Method (AMM).

This document defines how the alternate marking method can be used to measure packet loss and delay metrics of a service flow over e2e or any segment of the SFC.

2. Conventions used in this document

2.1. Terminology

MM: Marking Method

OAM: Operations, Administration and Maintenance

SFC: Service Function Chain

SF: Service Function

SFF: Service Function Forwarder

SFP: Service Function Path

NSH: Network Service Header

2.2. Requirements Language

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

3. Mark Field in NSH Base Header

[RFC8300] defines the format of the Network Service Header (NSH).

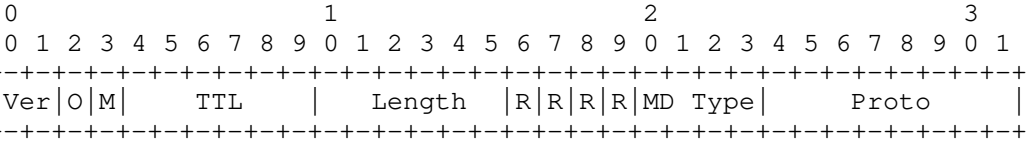


Figure 1: NSH Base format

This document defines the one-bit long field, referred to as Mark field (M in Figure 1, as part of NSH Base and designated for the alternate marking performance measurement method [RFC8321]. The Mark field MUST NOT be used in defining forwarding and/or quality of service treatment of an SFC packet. The Mark field MUST be used only for the performance measurement of data traffic in the SFC layer. Because the setting of the field to any value does not affect forwarding and/or quality of service treatment of a packet, the alternate marking method in SFC layer can be viewed as a real example of passive performance measurement method.

4. Theory of Operation

The marking method can be successfully used in the SFC. Without limiting any generality consider SFC presented in Figure 2. Any combination of markings, Loss and/or Delay, can be applied to a service flow by any component of the SFC at either ingress or egress

point to perform node, link, segment or end-to-end measurement to detect performance degradation defect and localize it efficiently.

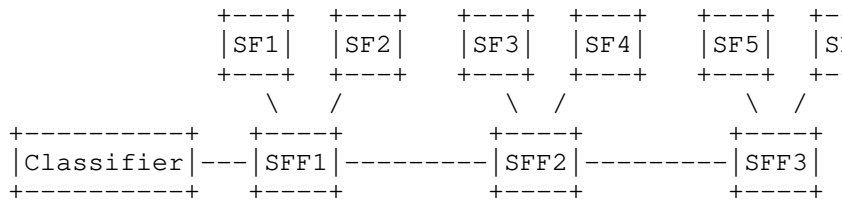


Figure 2: SFC network

Using the marking method a component of the SFC creates distinct sub-flows in the particular service traffic over SFC. Each sub-flow consists of consecutive blocks that are unambiguously recognizable by a monitoring point at any component of the SFC and can be measured to calculate packet loss and/or packet delay metrics.

4.1. Single Mark Enabled Measurement

As explained in the [RFC8321], marking can be applied to delineate blocks of packets based either on the equal number of packets in a block or based on the same time interval. The latter method offers better control as it allows better account for capabilities of downstream nodes to report statistics related to batches of packets and, at the same time, time resolution that affects defect detection interval.

The Loss flag is used to create alternate flows to measure the packet loss by switching the value of the Loss flag every N-th packet or at specified time intervals. Delay metrics MAY be calculated with the alternate flow using any of the following methods:

- First/Last Packet Delay calculation: whenever the marking, i.e., the value of Loss flag, changes a component of the SFC can store the timestamp of the first/last packet of the block. The timestamp can be compared with the timestamp of the packet that arrived in the same order through a monitoring point at a downstream component of the SFC to compute packet delay. Because timestamps collected based on order of arrival, this method is sensitive to packet loss and re-ordering of packets

- Average Packet Delay calculation: an average delay is calculated by considering the average arrival time of the packets within a single block. A component of the SFC may collect timestamps for

each packet received within a single block. Average of the timestamp is the sum of all the timestamps divided by the total number of packets received. Then the difference between averages calculated at two monitoring points is the average packet delay on that segment. This method is robust to out of order packets and also to packet loss (only a small error is introduced). This method only provides a single metric for the duration of the block, and it doesn't give the minimum and maximum delay values. Highly optimized implementation of the method can reduce the duration of the block and thus overcome the limitation.

4.2. Double Mark Enabled Measurement

Double Mark method allows measurement of minimum and maximum delays for the monitored flow, but it requires more nodal and network resources. If the Double Mark method used, then the Loss flag MUST be used to create the alternate flow, i.e., mark larger batches of packets. The Delay flag MUST be used to denote single packets to measure delay jitter.

The first marking (Loss flag alternation) is needed for packet loss and also for average delay measurement. The second marking (Delay flag is put to one) creates a new set of marked packets that are fully identified over the SFC, so that a component can store the timestamps of these packets; these timestamps can be compared with the timestamps of the same packets on another element of the SFC to compute packet delay values for each packet. The number of measurements can be easily increased by changing the frequency of the second marking. But the rate of the second marking must be not too high to avoid out of order issues. This method supports the calculation of not only the average delay but also the minimum and maximum delay values and, in broader terms, to know more about the statistic distribution of delay values.

4.3. Multiplexed Mark Enabled Measurement

There is also a scheme that provides the benefits of Double Mark method, but uses only one bit like Single Mark. This methodology is described in [I-D.mizrahi-ippm-compact-alternate-marking]. The concept is that in the middle of each block of packets with a certain value of the L flag, a single packet has the L flag inverted. So, by examining the stream, the packets with the inverted bit can be easily identified and employed for delay measurement. This Alternate Marking variation is advantageous because it requires only one bit from each packet, and such bits are always in short supply.

4.4. Residence Time Measurement with the Alternate Marking Method

Residence time is the variable part of the propagation delay that a packet experiences while traversing a network, e.g., SFC. Residence Time over an SFC is the sum of the nodal residence times, i.e., periods that the packet spent in each of SFFs that compose the SFC. The nodal residence time in SFC itself is the sum of sub-nodal residence times that the packet spent in each of SFs that are part of the given SFC and are mapped to the SFF. The residence time and deviation of the residence time metrics may include any combination of minimum, maximum, values over measurement period, as well as mean, median, percentile. These metrics may be used to evaluate the performance of the SFC and its elements before and during its operation.

Use of the specially marked packets simplifies residence time measurement and correlation of the measured metrics over the SFC end-to-end. For example, the alternate marking method may be used as described in Section 4.2 to identify packets in the data flow to be used to measure the residence time. The nodal and sub-nodal residence time metrics can be locally calculated and then collected using either in-band or out-band OAM mechanisms.

5. IANA Considerations

5.1. Mark Field in NSH Base Header

This document requests IANA to allocate the one-bit field from NSH Base Header Bits [RFC8300] as the Mark field of NSH as the following:

Bit Position	Description	Reference
TBA	Mark field	This document

Table 1: Mark field of SFC NSH

6. Security Considerations

This document lists the OAM requirement for SFC domain and does not raise any security concerns or issues in addition to ones common to networking and SFC.

7. Acknowledgment

TBD

8. References

8.1. Normative References

[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>.

[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>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

8.2. Informative References

[I-D.mizrahi-ippm-compact-alternate-marking] Mizrahi, T., Arad, C., Fioccola, G., Cociglio, M., Chen, M., Zheng, L., and G. Mirsky, "Compact Alternate Marking Methods for Passive and Hybrid Performance Monitoring", draft-mizrahi-ippm-compact-alternate-marking-03 (work in progress), October 2018.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Giuseppe Fioccola
Huawei Technologies

Email: giuseppe.fioccola@huawei.com

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: February 21, 2018

T. Mizrahi
I. Yerushalmi
D. Melman
Marvell
R. Browne
Intel
August 20, 2017

Network Service Header (NSH) Context Header Allocation: Timestamp
draft-mymb-sfc-nsh-allocation-timestamp-02

Abstract

This memo defines an allocation for the Context Headers of the Network Service Header (NSH), which incorporates the packet's timestamp, a sequence number, and a source interface identifier.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 21, 2018.

Table of Contents

1. Introduction

The Network Service Header (NSH), defined in [I-D.ietf-sfc-nsh], is an encapsulation header that is used in Service Function Chains (SFC).

The NSH specification [I-D.ietf-sfc-nsh] supports two possible methods of including metadata in the NSH; MD Type 0x1 and MD Type 0x2. When using MD Type 0x1 the NSH includes 16 octets of Context Header fields. The current memo proposes an allocation for the MD Type 0x1 Context Headers, which incorporates the timestamp of the packet, a sequence number, and a source interface identifier.

In a nutshell, packets that enter the SFC-Enabled Domain are timestamped. The timestamp is measured by the Classifier [RFC7665], and incorporated in the NSH. The timestamp may be used for various different purposes, including delay measurement, packet marking for passive performance monitoring, and timestamp-based policies. Notably, the timestamp does not increase the packet length, since it is incorporated in the MD Type 0x1 Mandatory Context Headers.

The source interface identifier indicates the interface through which the packet was received at the classifier. This identifer may specify a physical or a virtual interface. The sequence numbers can be used by Service Functions (SFs) to detect out-of-order delivery or

duplicate transmissions. The sequence number is maintained on a per-source-interface basis.

KPI-stamping [I-D.browne-sfc-nsh-kpi-stamp] defines an NSH timestamping mechanism that uses the MD Type 0x2 format. The current memo defines a compact MD Type 0x1 Context Header that does not require the packet to be extended beyond the NSH header. Furthermore, the two timestamping mechanisms can be used in concert, as further discussed below.

2. Terminology

2.1. 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].

2.2. Abbreviations

The following abbreviations are used in this document:

KPI	Key Performance Indicators [I-D.browne-sfc-nsh-kpi-stamp]
NSH	Network Service Header [I-D.ietf-sfc-nsh]
MD	Metadata [I-D.ietf-sfc-nsh]
SF	Service Function [RFC7665]
SFC	Service Function Chaining [RFC7665]

3. NSH Context Header Allocation Allocation

This memo defines the following Context Header allocation, as presented in Figure 1.

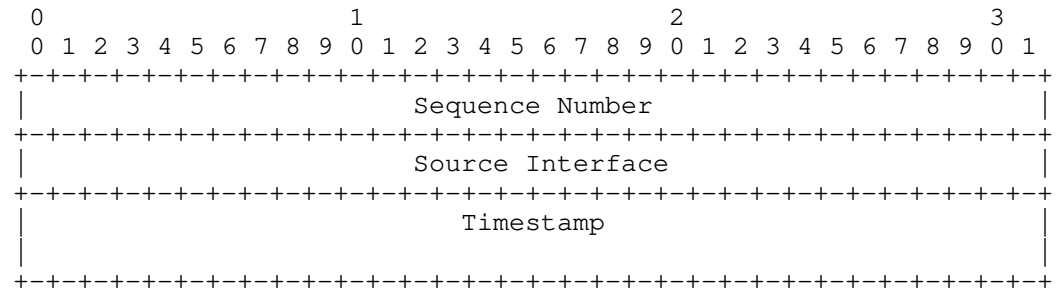


Figure 1: NSH Timestamp Allocation.

The NSH Timestamp Allocation includes the following fields:

- Sequence Number - a 32-bit sequence number. The sequence number is maintained on a per-source-interface basis. The sequence numbers can be used by SFs to detect out-of-order delivery, or duplicate transmissions.

- Source Interface - a 32-bit source interface identifier that is assigned by the Classifier.

- Timestamp - this field is 8 octets long, and specifies the time at which the packet was received by the Classifier. Two possible timestamp formats can be used for this field: the two 64-bit recommended formats specified in [I-D.mizrahi-intarea-packet-timestamps]. One of the formats is based on the [IEEE1588] timestamp format, and the other is based on the [RFC5905] format. It is assumed that in a given administrative domain only one of the formats will be used, and that the control plane determines which timestamp format is used.

The two timestamp formats that can be used in the timestamp field are:

- IEEE 1588 Truncated Timestamp Format: the format of this field uses the 64 least significant bits of the IEEE 1588-2008 Precision Time Protocol format [IEEE1588]. This truncated format consists of a 32-bit seconds field followed by a 32-bit nanoseconds field. As defined in [IEEE1588], the timestamp specifies the number of seconds elapsed since 1 January 1970 00:00:00 according to the International Atomic Time (TAI).

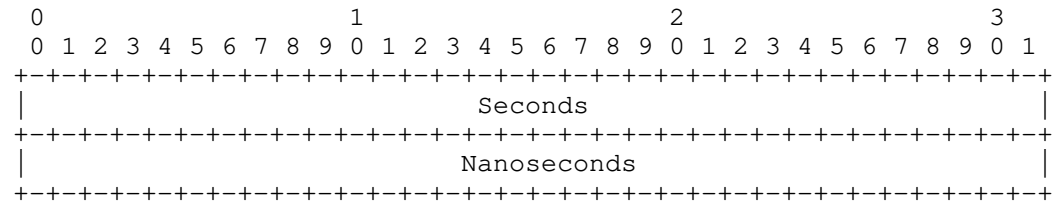


Figure 2: IEEE 1588 Truncated Timestamp Format [IEEE1588].

- NTP 64-bit Timestamp Format: this format consists of a 32-bit seconds field followed by a 32-bit fractional second field. As defined in [RFC5905], the timestamp specifies the number of seconds elapsed since 1 January 1900 00:00:00 according to the Coordinated Universal Time (UTC).

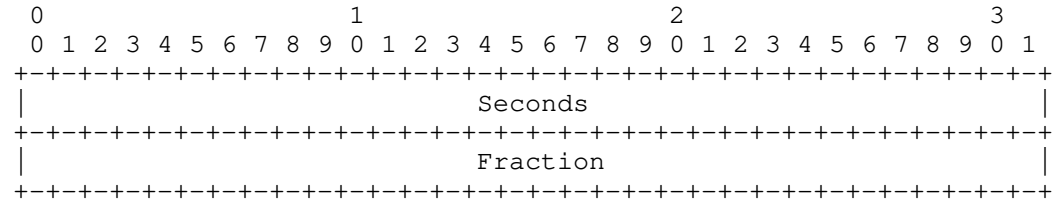


Figure 3: NTP [RFC5905] 64-bit Timestamp Format

4. Timestamping Use Cases

4.1. Network Analytics

Per-packet timestamping enables coarse-grained monitoring of the network delay along the Service Function Chain. Once a potential problem or bottleneck is detected, for example when the delay exceeds a certain policy, a highly-granular hop-by-hop monitoring mechanism, such as [I-D.browne-sfc-nsh-kpi-stamp] or [I-D.brockners-inband-oam-data], can be triggered, allowing to analyze and localize the problem.

Timestamping is also useful for logging and for flow analytics. It is often useful to maintain the timestamp of the first and last packet of the flow. Furthermore, traffic mirroring and sampling often requires a timestamp to be attached to analyzed packets. Attaching the timestamp to the NSH Context Header provides an in-band common time reference that can be used for various network analytics applications.

4.2. Alternate Marking

A possible approach for passive performance monitoring is to use an alternate marking method [I-D.ietf-ippm-alt-mark]. This method requires data packets to carry a field that marks (colors) the traffic, and enables passive measurement of packet loss, delay, and delay variation. The value of this marking field is periodically toggled between two values.

When the timestamp is incorporated in the NSH Context Header, it can natively be used for alternate marking. For example, the least significant bit of the timestamp Seconds field can be used for this purpose, since the value of this bit is inherently toggled every second.

4.3. Consistent Updates

The timestamp can be used for taking policy decisions such as 'Perform action A if timestamp>=T_0'. This can be used for enforcing time-of-day policies or periodic policies in service functions. Furthermore, timestamp-based policies can be used for enforcing consistent network updates, as discussed in [DPT].

5. Synchronization Considerations

Some of the applications that make use of the timestamp require the Classifer and SFs to be synchronized to a common time reference, for example using the Network Time Protocol [RFC5905], or the Precision Time Protocol [IEEE1588].

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

The security considerations of NSH in general are discussed in [I-D.ietf-sfc-nsh]. The security considerations of in-band timestamping in the context of NSH is discussed in [I-D.browne-sfc-nsh-kpi-stamp], and the current section is based on that discussion.

The use of in-band timestamping, as defined in this document, can be used as a means for network reconnaissance. By passively eavesdropping to timestamped traffic, an attacker can gather information about network delays and performance bottlenecks. A man-in-the-middle attacker can maliciously modify timestamps in order to

attack applications that use the timestamp values, such as performance monitoring applications.

Since the timestamping mechanism relies on an underlying time synchronization protocol, by attacking the time protocol an attack can potentially compromise the integrity of the NSH timestamp. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [RFC7384].

8. References

8.1. Normative References

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-19 (work in progress), August 2017.

[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>.

8.2. Informative References

[DPT] Mizrahi, T., Moses, Y., "The Case for Data Plane Timestamping in SDN", IEEE INFOCOM Workshop on Software-Driven Flexible and Agile Networking (SWFAN), 2016.

[I-D.brockners-inband-oam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", draft-brockners-inband-oam-data-07 (work in progress), July 2017.

[I-D.browne-sfc-nsh-kpi-stamp]
Browne, R., Chilikin, A., and T. Mizrahi, "Network Service Header KPI Stamping", draft-browne-sfc-nsh-kpi-stamp-01 (work in progress), April 2017.

[I-D.ietf-ippm-alt-mark]
Fioccola, G., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate Marking method for passive and hybrid performance monitoring", draft-ietf-ippm-alt-mark-06 (work in progress), July 2017.

[I-D.mizrahi-intarea-packet-timestamps]
Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", draft-mizrahi-intarea-packet-timestamps-00 (work in progress), June 2017.

[IEEE1588]
IEEE, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", 2008.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <https://www.rfc-editor.org/info/rfc7384>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Tal Mizrahi
Marvell
6 Hamada
Yokneam 2066721
Israel

Email: talmi@marvell.com

Ilan Yerushalmi
Marvell
6 Hamada
Yokneam 2066721
Israel

Email: yilan@marvell.com

David Melman
Marvell
6 Hamada
Yokneam 2066721
Israel

Email: davidme@marvell.com

Rory Browne
Intel
Dromore House
Shannon, Co.Clare
Ireland

Email: rory.browne@intel.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 13, 2019

T. Mizrahi
Huawei Network.IO Innovation Lab
I. Yerushalmi
D. Melman
Marvell
R. Browne
Intel

October 10, 2018

Network Service Header (NSH) Context Header Allocation: Timestamp
draft-mymb-sfc-nsh-allocation-timestamp-05

Abstract

This memo defines an allocation for the Context Headers of the Network Service Header (NSH), which incorporates the packet's timestamp, a sequence number, and a source interface identifier.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 13, 2019.

Copyright Notice

Table of Contents

1. Introduction

The Network Service Header (NSH), defined in [I-D.ietf-sfc-nsh], is an encapsulation header that is used in Service Function Chains (SFC).

The NSH specification [I-D.ietf-sfc-nsh] supports two possible methods of including metadata in the NSH; MD Type 0x1 and MD Type 0x2. When using MD Type 0x1 the NSH includes 16 octets of Context Header fields. The current memo proposes an allocation for the MD Type 0x1 Context Headers, which incorporates the timestamp of the packet, a sequence number, and a source interface identifier.

In a nutshell, packets that enter the SFC-Enabled Domain are timestamped. The timestamp is measured by the Classifier [RFC7665], and incorporated in the NSH. The timestamp may be used for various different purposes, including delay measurement, packet marking for passive performance monitoring, and timestamp-based policies. Notably, the timestamp does not increase the packet length, since it is incorporated in the MD Type 0x1 Mandatory Context Headers.

The source interface identifier indicates the interface through which the packet was received at the classifier. This identifer may specify a physical or a virtual interface. The sequence numbers can be used by Service Functions (SFs) to detect out-of-order delivery or

duplicate transmissions. The sequence number is maintained on a per-source-interface basis.

KPI-stamping [I-D.browne-sfc-nsh-kpi-stamp] defines an NSH timestamping mechanism that uses the MD Type 0x2 format. The current memo defines a compact MD Type 0x1 Context Header that does not require the packet to be extended beyond the NSH header. Furthermore, the two timestamping mechanisms can be used in concert, as further discussed below.

2. Terminology

2.1. 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].

2.2. Abbreviations

The following abbreviations are used in this document:

KPI	Key Performance Indicators [I-D.browne-sfc-nsh-kpi-stamp]
NSH	Network Service Header [I-D.ietf-sfc-nsh]
MD	Metadata [I-D.ietf-sfc-nsh]
SF	Service Function [RFC7665]
SFC	Service Function Chaining [RFC7665]

3. NSH Context Header Allocation

This memo defines the following Context Header allocation, as presented in Figure 1.

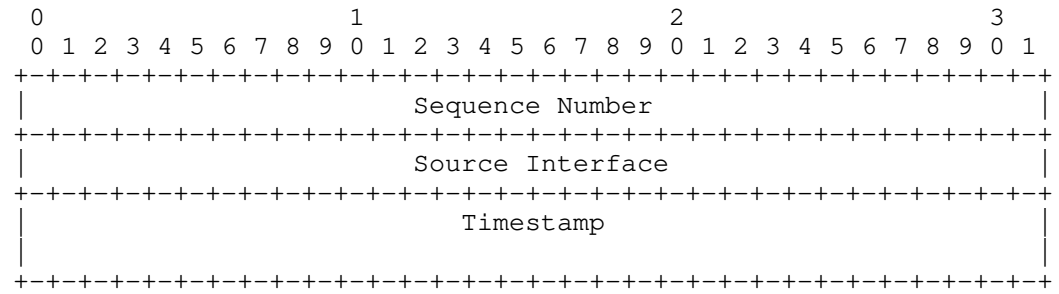


Figure 1: NSH Timestamp Allocation.

The NSH Timestamp Allocation includes the following fields:

- Sequence Number - a 32-bit sequence number. The sequence number is maintained on a per-source-interface basis. The sequence numbers can be used by SFs to detect out-of-order delivery, or duplicate transmissions.

- Source Interface - a 32-bit source interface identifier that is assigned by the Classifier.

- Timestamp - this field is 8 octets long, and specifies the time at which the packet was received by the Classifier. Two possible timestamp formats can be used for this field: the two 64-bit recommended formats specified in [I-D.ietf-ntp-packet-timestamps]. One of the formats is based on the [IEEE1588] timestamp format, and the other is based on the [RFC5905] format. It is assumed that in a given administrative domain only one of the formats will be used, and that the control plane determines which timestamp format is used.

The two timestamp formats that can be used in the timestamp field are:

- IEEE 1588 Truncated Timestamp Format: as specified in Section 4.3 of [I-D.ietf-ntp-packet-timestamps]. This timestamp format uses the 64 least significant bits of the IEEE 1588-2008 Precision Time Protocol format [IEEE1588], and consists of a 32-bit seconds field followed by a 32-bit nanoseconds field.

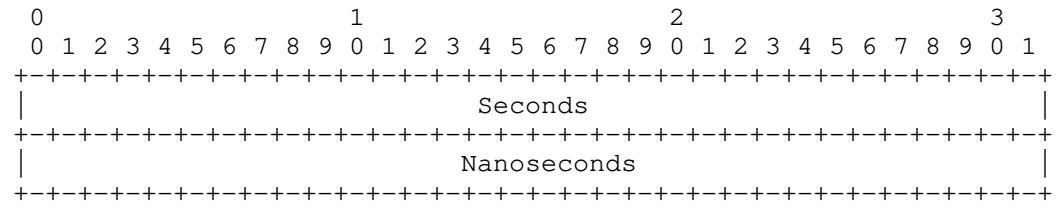


Figure 2: IEEE 1588 Truncated Timestamp Format [IEEE1588].

- NTP [RFC5905] 64-bit Timestamp Format: as specified in Section 4.2.1 of [I-D.ietf-ntp-packet-timestamps]. This format consists of a 32-bit seconds field followed by a 32-bit fractional second field.

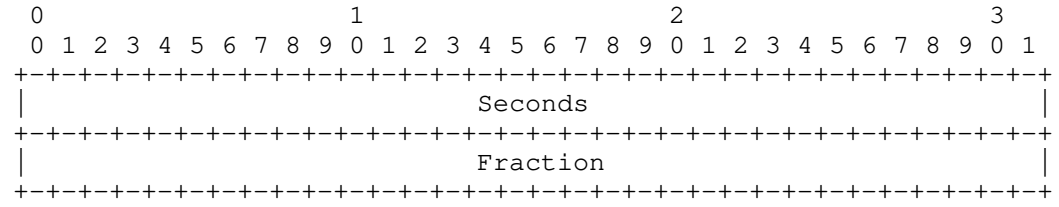


Figure 3: NTP [RFC5905] 64-bit Timestamp Format

Synchronization aspects of the timestamp format are discussed in Section 5.

4. Timestamping Use Cases

4.1. Network Analytics

Per-packet timestamping enables coarse-grained monitoring of the network delay along the Service Function Chain. Once a potential problem or bottleneck is detected, for example when the delay exceeds a certain policy, a highly-granular hop-by-hop monitoring mechanism, such as [I-D.browne-sfc-nsh-kpi-stamp] or [I-D.brockners-inband-oam-data], can be triggered, allowing to analyze and localize the problem.

Timestamping is also useful for logging and for flow analytics. It is often useful to maintain the timestamp of the first and last packet of the flow. Furthermore, traffic mirroring and sampling often requires a timestamp to be attached to analyzed packets. Attaching the timestamp to the NSH Context Header provides an in-band common time reference that can be used for various network analytics applications.

4.2. Alternate Marking

A possible approach for passive performance monitoring is to use an alternate marking method [RFC8321]. This method requires data packets to carry a field that marks (colors) the traffic, and enables passive measurement of packet loss, delay, and delay variation. The value of this marking field is periodically toggled between two values.

When the timestamp is incorporated in the NSH Context Header, it can natively be used for alternate marking. For example, the least significant bit of the timestamp Seconds field can be used for this purpose, since the value of this bit is inherently toggled every second.

4.3. Consistent Updates

The timestamp can be used for taking policy decisions such as 'Perform action A if timestamp>=T_0'. This can be used for enforcing time-of-day policies or periodic policies in service functions. Furthermore, timestamp-based policies can be used for enforcing consistent network updates, as discussed in [DPT].

5. Synchronization Considerations

Some of the applications that make use of the timestamp require the Classifer and SFs to be synchronized to a common time reference, for example using the Network Time Protocol [RFC5905], or the Precision Time Protocol [IEEE1588]. Although it is not a requirement to use a clock synchronization mechanism, it is expected that depending on the applications that use the timestamp, such synchronization mechanisms will be used in most deployments that use the timestamp allocation.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

The security considerations of NSH in general are discussed in [I-D.ietf-sfc-nsh]. The security considerations of in-band timestamping in the context of NSH is discussed in [I-D.browne-sfc-nsh-kpi-stamp], and the current section is based on that discussion.

The use of in-band timestamping, as defined in this document, can be used as a means for network reconnaissance. By passively eavesdropping to timestamped traffic, an attacker can gather

information about network delays and performance bottlenecks. A man-in-the-middle attacker can maliciously modify timestamps in order to attack applications that use the timestamp values, such as performance monitoring applications.

Since the timestamping mechanism relies on an underlying time synchronization protocol, by attacking the time protocol an attack can potentially compromise the integrity of the NSH timestamp. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [RFC7384].

8. References

8.1. Normative References

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-28 (work in progress), November 2017.

[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>.

8.2. Informative References

[DPT] Mizrahi, T., Moses, Y., "The Case for Data Plane Timestamping in SDN", IEEE INFOCOM Workshop on Software-Driven Flexible and Agile Networking (SWFAN), 2016.

[I-D.brockners-inband-oam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., and d. daniel.bernier@bell.ca, "Data Fields for In-situ OAM", draft-brockners-inband-oam-data-07 (work in progress), July 2017.

[I-D.browne-sfc-nsh-kpi-stamp]
Browne, R., Chilikin, A., and T. Mizrahi, "A Key Performance Indicators (KPI) Stamping for the Network Service Header (NSH)", draft-browne-sfc-nsh-kpi-stamp-05 (work in progress), August 2018.

[I-D.ietf-ntp-packet-timestamps]
Mizrahi, T., Fabini, J., and A. Morton, "Guidelines for Defining Packet Timestamps", draft-ietf-ntp-packet-timestamps-04 (work in progress), October 2018.

[IEEE1588]
IEEE, "IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", 2008.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <https://www.rfc-editor.org/info/rfc7384>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

Authors' Addresses

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com

Ilan Yerushalmi
Marvell
6 Hamada
Yokneam 2066721
Israel

Email: yilan@marvell.com

David Melman
Marvell
6 Hamada
Yokneam 2066721
Israel

Email: davidme@marvell.com

Rory Browne
Intel
Dromore House
Shannon, Co.Clare
Ireland

Email: rory.browne@intel.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 30, 2018

D. Purkayastha
A. Rahman
D. Trossen
InterDigital Communications, LLC
Z. Despotovic
R. Khalili
Huawei

October 27, 2017

Alternative Handling of Dynamic Chaining and Service Indirection
draft-purkayastha-sfc-service-indirection-01

Abstract

Many stringent requirements are imposed on today's network, such as low latency, high availability and reliability in order to support several use cases such as IoT, Gaming, Content distribution, Robotics etc. Networks need to be flexible and dynamic in terms of allocation of services and resources. Network Operators should be able to reconfigure the composition of a service and steer users towards new service end points as users move or resource availability changes. SFC allows network operators to easily create and reconfigure service function chains dynamically in response to changing network requirements. We discuss a use case where Service Function Chain can adapt or self-organize as demanded by the network condition without requiring SPI re-classification. This can be achieved, for example, by decoupling the service consumer and service endpoint by a new service function proposed in this draft. We describe few requirements for this service function to enable dynamic switching between consumer and end point.

Table of Contents

1. Introduction

The requirements on today's networks are very diverse, enabling multiple use cases such as IoT, Content Distribution, Gaming, Network functions such as Cloud RAN. Every use case imposes certain requirements on the network. These requirements vary from one extreme to other and often they are in a divergent direction. Network operator and service providers are pushing many functions towards the edge of the network in order to be closer to the users. This reduces latency and backhaul traffic, as user request can be processed locally.

It becomes more challenging for the network when user mobility as well as non-deterministic availability of compute and storage resources are considered. The impact is felt most in the edge of the network because as the users move, their point of attachment changes frequently, which results in (at least partially) relocating the service as well as the service endpoint. Furthermore, network functions are pushed more and more towards the edge, where compute and storage resources are constrained and availability is non-deterministic. Also, storage resources may need to be moved where the user concentration is more in case of content delivery applications.

We describe a few use cases in the next section and derive the requirements for composing new services and service path in a dynamic edge network. We address this dynamicity by introducing a special Service Function, called SRR (service request routing). We describe the problems associated with today's network and Layer 3 based approach to handle dynamicity in the network. We then discuss how such new Service Function with certain capabilities can handle the dynamicity better than these conventional methods. Note : State migration is not in the scope of our solution since this problem is a general one pertaining to re-chaining stateful SFs.

2. Use Case Description

2.1. Data Center

The data center use case draft [I-D.ietf-sfc-dc-use-cases] describes an East West traffic use case. This is the predominant traffic in data centers today. Server virtualization has led to the new paradigm where virtual machines can migrate from one server to another across the data center. This explosion in east-west traffic is leading to newer data center network fabric architectures that provide consistent latencies from one point in the fabric to another.

SFCs applied in an enterprise or service provider data center can be broadly categorized into two types:

- Access SFCs
- Application SFCs

Access SFCs are focused on servicing traffic entering and leaving the data center while Application SFCs are focused on servicing traffic destined to applications. Service providers deploy a single "Access SFC" and multiple "Application SFCs" for each tenant. Enterprise data center operators on the other hand may not have a need for Access SFCs depending on the size and requirements of the enterprise.

In carrier networks, operators may deploy multiple data centers dispersed geographically. Each data center may host different types of service functions. For example, latency sensitive or high usage service functions are deployed in regional data centers while other latency tolerant, low usage service functions are deployed in global or central data centers. In such deployments, SFCs may span multiple data centers and enable operators to deploy services in a flexible and inexpensive way.

It is clear that within the data center as well as in inter data center scenarios, users are serviced by multiple SFs distributed inside as well as outside a location. In this scenario, it is clear that Service function chains should be able to reselect, redirect traffic very fast. The draft identifies that Static service chains do not allow for modifying the SFCs as they require the ability to add SNs or remove SNs to scale up and down the service capacity. Likewise the ability to dynamically pick one among the many SN instance is not available.

2.2. ETSI MEC USE CASE

Take the following video orchestration service example from ETSI MEC Requirements document [ETSI_MEC]. The proposed use case of edge video orchestration suggests a scenario where visual content can be produced and consumed at the same location close to consumers in a densely populated and clearly limited area. Such a case could be a sports event or concert where a remarkable number of consumers are using their handheld devices to access user select tailored content. The overall video experience is combined from multiple sources, such as local recording devices, which may be fixed as well as mobile, and master video from central production server. The user is given an opportunity to select tailored views from a set of local video sources.

2.3. 3GPP

3GPP Rel. 15 introduces the notion of the service-based interface (SBI) as an alternative to the traditional call pattern invocation of network functions. This introduction targets the support for replication, e.g., driven by virtualized functions, as well as supporting alternative interactions, e.g., for different vertical market specific control planes, by making the discovery as well as composition of new interactions more flexible.

We believe that SFC is a suitable framework for the interconnection of such network functions through the new SBI. One of the aforementioned driving forces, namely the replication of functions aligns with our thinking in this draft in that indirections to new

vertical instances need to be dynamic in reacting to the appearance of new virtual instances or to changes in policies for the selection of specific instances by specific calling entities.

2.4. Use Case Analysis

In such a dynamic network environment, the capability to dynamically compose new services from available services as well as move a service instance in response to user mobility or resource availability is desirable. SFC allows network operators as well as service providers to compose new services by chaining individual service functions towards the composed new service. In a dynamic network environment where service functions move frequently because of user movement, load balancing or resource modification, service function chains and the service end points need to be created and recreated frequently. SFC, as defined in IETF, is capable of modifying the service chain dynamically in response to network conditions.

In order to route the service requests to service end points in a dynamic manner, we identify the following desirable features in a service function chain:

- Fast switching from one service instance to another by not relying on the DNS for service location resolution. Instead of DNS, the function should be able to identify the path, which will allow to reach the service end point.

- Direct path mobility, where the path between the requester and the responding service can be determined as being optimal (e.g., shortest path or direct path to a selected instance), is needed to avoid the use of anchor points and further reduce service-level latency

- Indirect service requests at the network level, transparent to the requesting client and without the involvement of the DNS. End user is not aware of the decision made by the SF.

- New methods for forwarding, such as path-based forwarding, direct path routing in mobility cases, path pinning for traffic steering and simplified service-specific peering towards the Internet.

3. NSH and Re-classification

[RFC7498] captures the problems associated with existing service deployments that are problematic. The problems are described below at a high level:

- Network topology: Network service deployment is tightly coupled with network topology thus reducing the flexibility in service delivery. It adds complexity in deploying network service when certain traffic types may need some service and other traffic types do not need the same service.

- Configuration complexity is the direct result of dependency on network topology.

- Limited availability of services

- Altering the order of a deployed chain is complex and cumbersome

- Coupling of service functions to topology may require service functions to support many transport encapsulations or for a transport gateway function to be present.

- In a dynamic environment like the Edge of a network service delivery, routing changes fast. It may be difficult to deliver service dynamically due to the risk and complexity of VLANs and/or routing modifications.

These factors provide motivation for a simplified and flexible service insertion model that addresses many of the current shortcomings and provides new, much needed functionality to enable service deployments in modern network environments. Service chaining accomplishes this by considering service functions as resources, with associated attributes, available for scheduled consumption. Selective traffic, subject to policy, may then be "steered" to the requisite service resources, along with any "extra" information referred to as metadata. This metadata is used for policy enforcement.

A basic form of service chaining may be realized using existing transport encapsulations. This method of chaining relies upon the tunneling of selected data between service functions. Although this form of service chaining achieves some level of abstraction from the underlying topology, it does not truly create a service plane. NSH [I-D.ietf-sfc-nsh] is a distinct identifiable plane that can be used across all transports to create a service chain and exchange metadata along the chain.

Fundamentally, however, the notion of "services" in SFC is tied into specific service function endpoints, which lie along a well-defined service function path (SFP) where the path is defined through lower layer transport encapsulations. If any such service function endpoint changes, the service chain needs to be adjusted; a procedure we outline in the following sub-section.

3.1. Dynamic service chain creation using NSH

We revisit the dynamic service chain creation capability of NSH. NSH defines a new service plane protocol [I-D.ietf-sfc-nsh]. A Network Service Header (NSH) contains service path information and optionally metadata that are added to a packet or frame and used to create a service plane. A control plane is required in order to exchange NSH values with participating nodes, and to provision the same nodes with requisite information such as service path ID to overlay mapping.

The Network Service Header has three parts, Base header, Service Path Header and Context Header. NSH Service Path Header is a 4-byte service path header follows the base header and defines two fields used to construct a service path:

- Service path identifier (SPI)
- Service index (SI)

The following figure depicts the service path header.

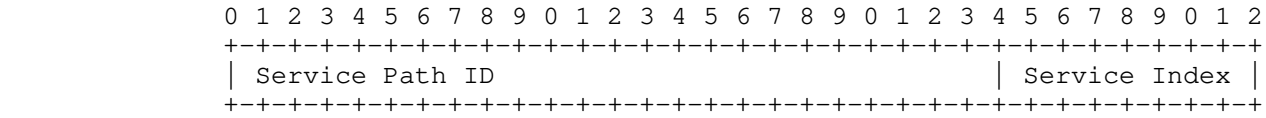


Figure 1: NSH Path Header

The service path identifier (SPI) is used to identify the service path that interconnects the needed service functions. It allows nodes to utilize the identifier to select the appropriate network transport protocol and forwarding techniques. The service index (SI) identifies the location of a packet within a service path. As packets traverse a service path, the SI is decremented post-service.

SPI represents the service path and altering the path identifier results in a change of a service path. A change in SPI value is a result of re-classification. It means a node in the service path determined, based on policy, that the initial classification was incorrect or incomplete. If the updated classification results in the necessity of a new service path, the node updates the SPI and SI fields accordingly. The new identifier is then used to select the appropriate overlay topology. This allows service functions to alter the path of a packet without having to participate in the network topology and its associated control plane(s). The method to determine that an existing classification is incorrect and how to determine the new classification is not defined.

4. Challenges with dynamic indirection

The emerging trend in today's network is to deploy network functions, services and applications at the edge of the network to support latency requirements, computational offload, traffic optimization etc. As users are moving, application or services being used by users, may need to be moved closer to the user's new location. This implies another instance of the service function may need to be instantiated close to the user's new location. It may result in re-establishing service path from the newly instantiated service function to other service instances. It is also possible that the newly instantiated service function may be redirected to a new service end point (e.g. Application Server) for various reasons, such as incomplete content, proximity to data store, load balancing etc. In another scenario, a single instance of the service function may not handle all users. A single service function may be instantiated more than once to balance user load. As the number of instances increase and along with mobility, the complexity of service routing increases. It is anticipated that there may be a constant action of function chaining, re-chaining occurring in the network.

The challenge of dynamic indirection may be better described by analyzing the working of CDNs, which dynamically (re-)direct user-initiated requests towards the most appropriate content instance. This task becomes more difficult if granularity of the instance placement increases. For instance, in case of a CDN being realized close to end users, specifically in edge of the network, the specific content instance might need to be selected dynamically. After initial selection, the instance may change during service execution.

In a conventional network, an instance of a service is found and selected using DNS. The subsequent service request is then routed through the network between the client and the service. If the user is doing a DNS lookup to access content served by a CDN then the DNS service will maintain a list of IP addresses that can be returned for a given domain name and will try to return an IP address of a node geographically close to the client. Should the service provider want to replace an instance of their service with another one at a different IP address (and potentially a different physical location for various reasons such as load balancing, reliability etc.) then the DNS tables must be updated, i.e., the service needs to be (re-)registered quickly. This is done by updating the local authoritative DNS server which then propagates the new mapping to DNS services across the world. DNS propagation can take up to 48 hours so fast and dynamic switching from one service instance to another is not possible in conventional networks. When relying on many surrogate service endpoints to exist in the edge network, there is a clear issue of certain resources not being available in one surrogate

instance while existing in another so that changes in redirection might be desirable, while also changes in local load drive the need for such change in redirection.

The other issue in conventional network lies with mobility management procedure. These procedures use an anchor point, which terminates a session at the network edge. As user moves around, traffic is redirected from the anchor point to the new point of attachment. Relying on typical mobility management approaches found in IP networks, usually leads to inefficient 'triangular' routing of requests through this common 'anchor' point. This triangular routing increases the latency in reaching the new service function or service end points as users move.

Traffic steering is a common procedure in managed networks, particularly at the edge, due to desired subscriber-centric traffic policies (e.g., related to pricing structures), resource requirements (e.g., related to using particular paths in the network) or mobility (e.g., users moving in a cellular network). Today's methods for traffic steering include anchor-based mobility management as well as traffic classification, for instance, in packet gateways of cellular systems (using, e.g., deep packet inspection as well as port and address classification). While the former leads to inefficient 'triangular' traffic forwarding, the latter often requires additional state in the forwarders to differentiate traffic from one user to another.

The analysis of CDN network shows that dynamic indirection is a necessary requirement, which needs to be supported by the networks. The goal for this indirection is to provide user applications lowest possible latency. But as discussed above, relying on today's technique does not help in guaranteeing same latency to user applications. On the other hand, there is a high possibility that latency may increase if we rely on Layer 3 based service redirection techniques.

SFC handles indirection through the use of SPI. A packet needs to be reclassified and the intermediate node changes the SPI. Following are the typical steps that happens in order to implement the indirection.

- A packet arrives at a particular node
- The node contacts the policy manager
- Identifies the current classification is incorrect
- Reclassifies the packet, i.e. change the SPI

- Inserts the packet in the pipe, possibly towards the SFF

The indirection mechanism in SFC involves certain steps to process policy information and change the SPI in the packet header, making it suitable to handle dynamic indirection requirements. Our proposed SF in this document provides an additional method to handle dynamic indirection of service requests, not relying on the reclassification mechanism. Combining these two techniques may provide flexibility and improvement over single method.

5. Desired Features

In order to route the service requests to service end points in a dynamic manner, we identify the following desirable features:

- Fast switching from one service instance to another by not relying on DNS for service location resolution. Instead of DNS, the function should be able to identify the path, which will allow to reach the service end point.

- Direct path mobility, where the path between the requester and the responding service can be determined as being optimal (e.g., shortest path or direct path to a selected instance), is needed to avoid the use of anchor points and reduce service-level latency

- Indirect service requests at the network level, transparent to the requesting client and without the involvement of the DNS. End user is not aware of the decision made by the SF.

- New methods for forwarding, such as path-based forwarding, direct path routing in mobility cases, path pinning for traffic steering and simplified service-specific peering towards the Internet.

6. Service Request Routing (SRR) Service Function

6.1. Overview

The following diagram shows the application of the new proposed SRR service function in an example of media clients connecting to media servers. There may be more than one media functions to support CDN like architecture, Surrogate servers to handle mobility and load balancing.

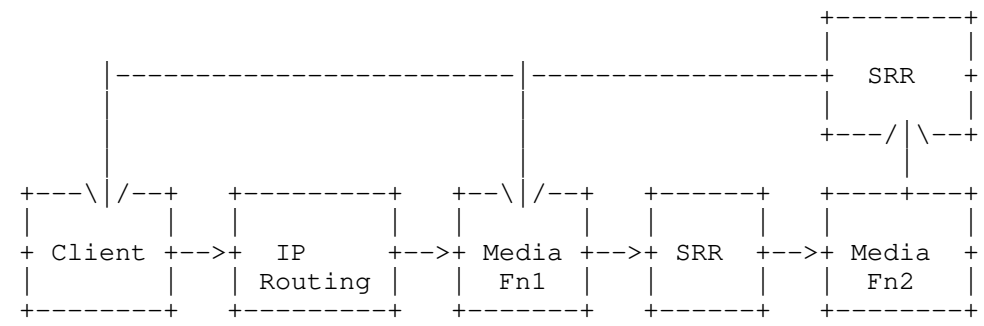


Figure 2: General SFC with SRR Flexible Chaining, Initiated via IP Routed Client Connection

The clients are connected to media functions through frontend routed network, e.g., relying on standard IP routing, while media functions are chained via the new proposed service request routing (SRR) function. Alternatively, we also envision to utilize the SRR function directly between client SF and media function SF, as outlined in the figure below

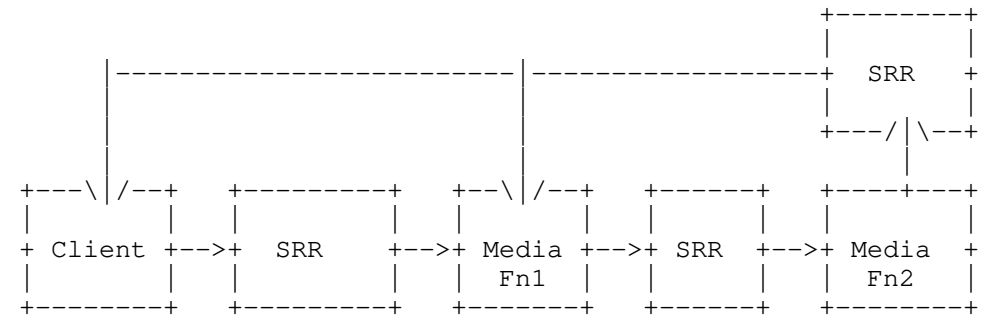


Figure 3: General SFC with SRR Flexible Chaining, Initiated between via SRR Chained Client

For our considerations, we assume that each SF is realized by at least one or more service function endpoints (SFEs). Hence, instead of looking at "chaining" as a concept that connects specific SFEs along a well-defined SFP, we propose to look at "chaining" at the level of "named" service functions rather than their specific endpoints. With this in mind, the SRR service function lifts the relationship between the connecting SFs to the level of "logical" service functions rather than their specific realizing endpoints.

Instead of relying on dynamic re-chaining in case of any dynamically changing relationship between specific SFEs, the SRR provides the selection of suitable SFEs while maintaining the logical relationship between the SFs. In Section 6.3, we will present the necessary extensions to the SFP concept to support this higher abstraction of "chaining" via "named" logical SFs. The SRR introduces the flexibility in routing service requests from client to specific SFEs. In the edge network, where users are moving and service end points may also change, having flexibility to decide and steer service requests directly helps in guaranteeing the same latency to user applications. Clearly, that is achieved by reducing the switching time from SF to another. As service end point changes, the routing functions makes instantaneous decision to route the request to the appropriate media server.

The possible improvements of using SRR within an SFC framework are listed below:

- Fast (between 10 and 20ms) switching times from one service instance to another by not relying on the DNS for service discovery and directly routing service requests at the level of the transport network.

- The capability to indirect service requests at the network level will help in reducing latency, when service end points change. E.g. when a service request is being sent to one surrogate instance but results in a HTTP 404 or 5xx error response, the original request is redirected to another alternative surrogate with minimal latency, i.e., right at the destination of said failed service request. Nesting these operations effectively leads to a net-level 'search' among all available surrogate instances until the search is exhausted (with a negative result) or the resource is found.

- New methods for forwarding, such as path-based forwarding, will enable direct path routing in mobility cases, path pinning for traffic steering and simplified service-specific peering towards the Internet. Such capability would allow for localizing traffic, reduce latency and costs.

6.2. Notion of HTTP-Based Transport

As a first proposed extension to the SFC framework, we introduce the notion of a "HTTP-based transport" utilizing URLs as addressing scheme. With that, we can create SFPs as shown in Fig 4, "i.e., 192.168.x.x -> www.foo.com -> 192.168.x.x -> www.foo2.com -> 192.168.x.x -> ... -> www.fooN.com." It is this "name-based" relationship that we see possibly realized through specific

replicated instances, where in turn the routing towards those specific instances is realized by the SRR.

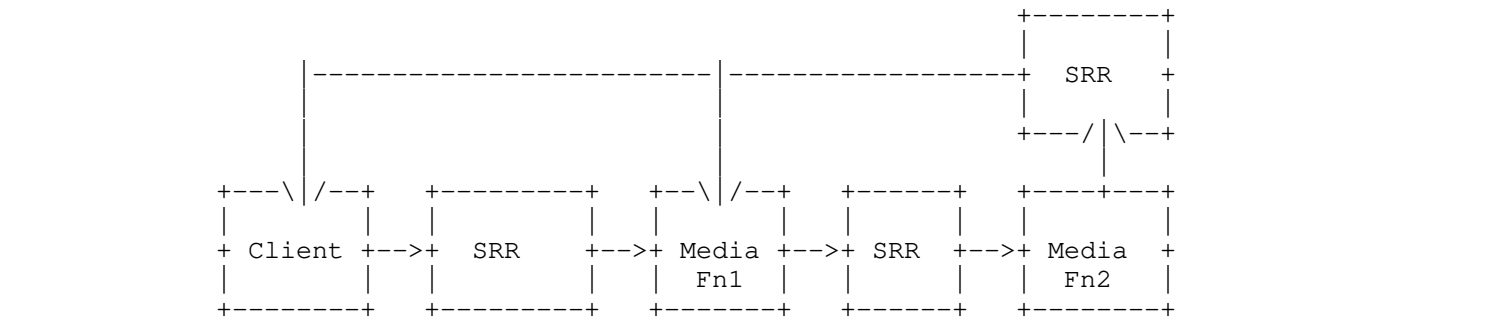


Figure 4: SFP with new HTTP-based Transport option

In a pure SFC architectural framework, Classifier function can may interact with SRR to obtain an SE (Service Encapsulation). E.g. the Classifier function may look into the network locator map in Fig 4 and determine the next SF is www.foo.com. It provides this information to SRR to obtain the next hop information. SRR returns the SE for next hop, which can be a "bitfield" information that is being used in the overlay routing for this part of the SFP. The Classifier function uses this SE to route the incoming packet directly at the transport network level.

6.3. Details of SRR Function

Assuming such introduction of an HTTP-level transport notion, the SRR function can be decomposed further as shown in Fig 5.

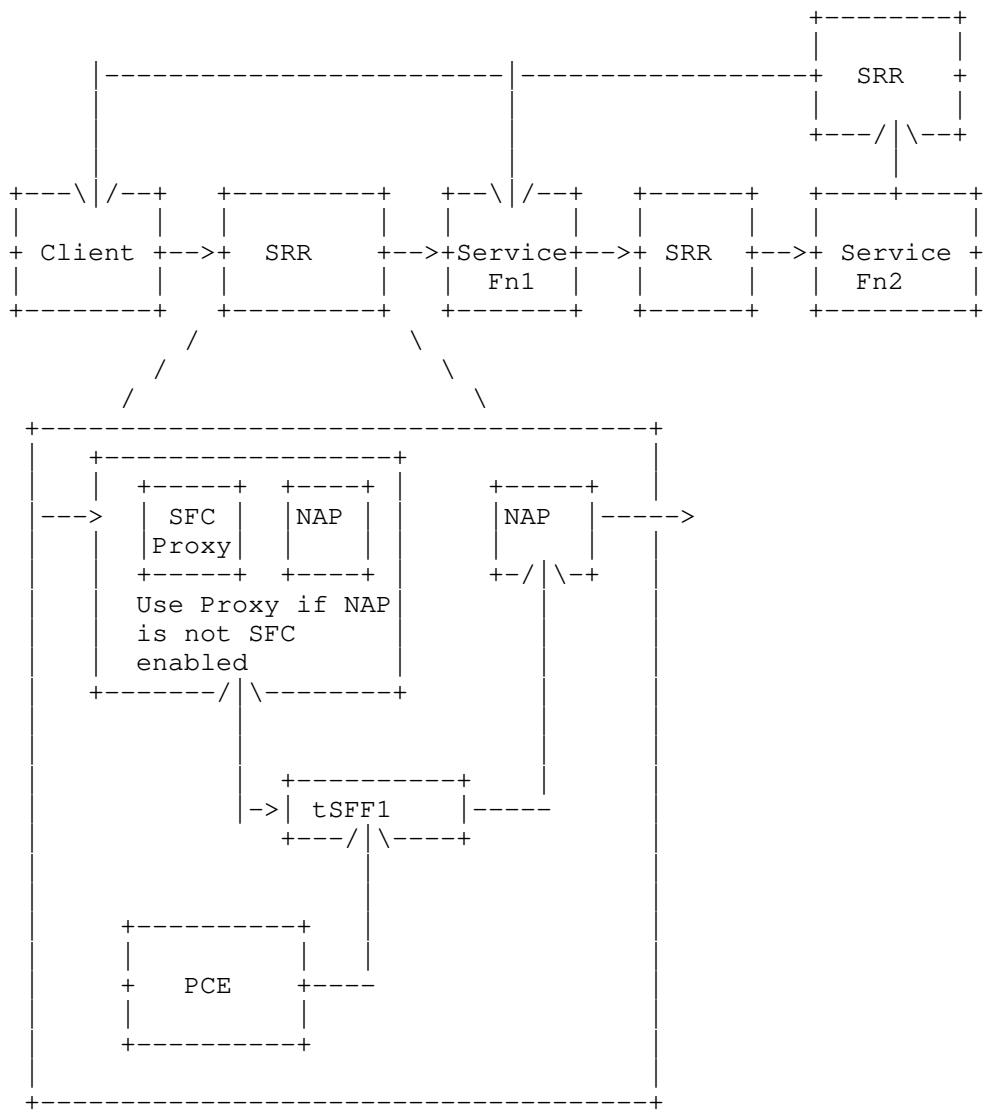


Figure 5: SRR decomposition

Another option for the two functions routing via the SRR could be entirely link-local, i.e., there's another simple tSFF2 between client and SRR as well as SF1 and SRR that is simply a link-local transport. The following figure describes this alternate option.

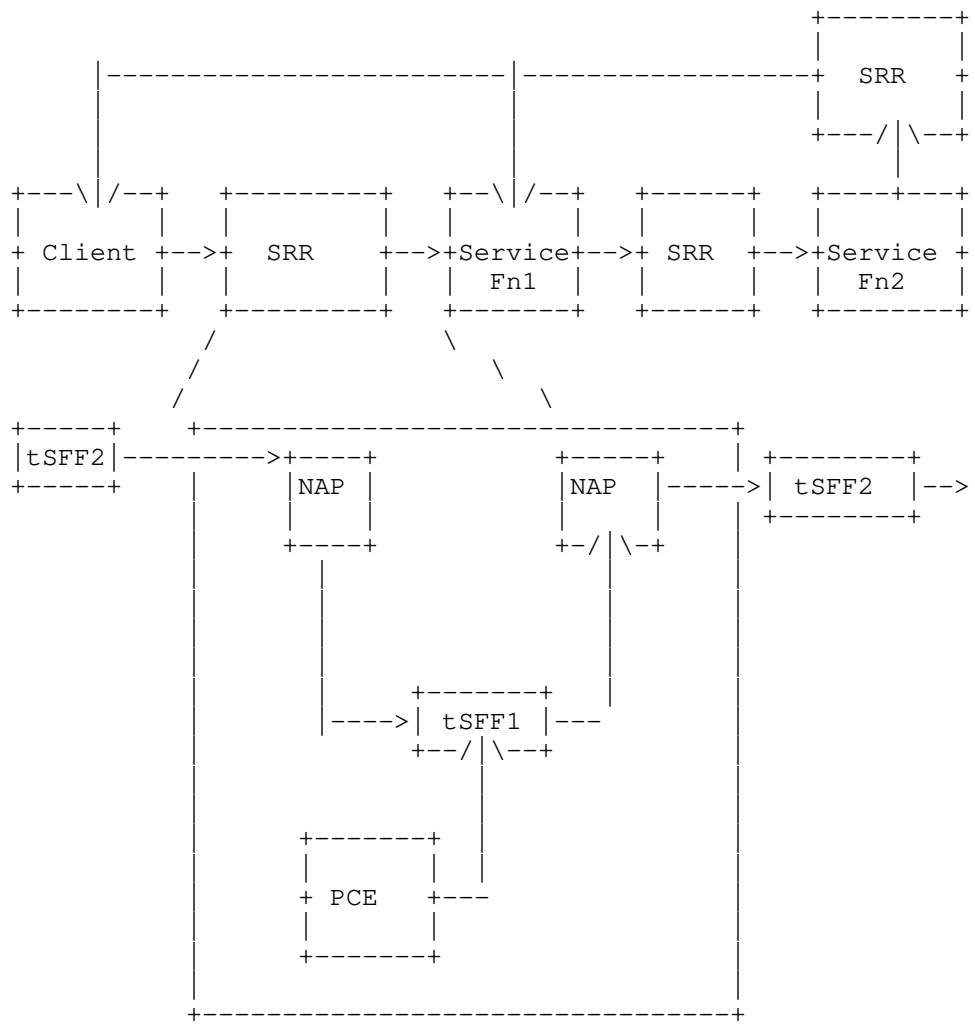


Figure 6: SRR decomposition using link-local client/function communication

The SRR function may be composed of the following functions:

- NAP at the ingress, terminates on the client side Layer 3 and above protocols, such as TCP

- NAP at the egress, terminates any transport protocol on the network outgoing (server) side

- PCE, Path Computation Element Policy control and Enforcement function is responsible for selecting the correct next SF, also possibly realizing path policy enforcement. The result of the selection is a path identifier which is delivered to the ingress NAP upon initial path computation request (i.e., when sending a request to a specific URL on the SFP for the first time). The path identifier is utilized for any future request for a given URL-based SF. In case of another SF instance becoming available, indicated to the PCE through a registration procedure, the PCE will instruct all ingress NAPs to invalidate path identifiers to the specific URL of the SF, resulting in an initial path computation request at the next SF request forwarding. Through this, the newly registered SF instance might be utilized if the policy-governed path computation will select said SF instance.

- Transport-derived SFF (tSFF1): the communication between ingress/ egress NAPs as well as NAPs to PCE is realized via a transport-derived SFF. We outline here three possible tSFFs

- SDN-based: The Transport Derived SFFThis option (tSFF), utilizes path-based forwarding utilizing through SDN-based wildcard matching fields, supported according to with OF1.2+ [Reed2016]. It can be embedded into slicing approach of underlying transport infrastructure by leaving typical slicing fields available (e.g., VLAN tags). The forwarding utilizes the Ethernet frame format at Layer 2, representing the topological links of a specific forwarding path in the transport network as unique bits in a fixed size bit array. For the latter, the approach utilizes the IPv6 source and destination fields for storing the bit array information (in a simple version for this forwarding, this limits the topology to 256 links but extensions schemes are possible, which are left out of this document at this stage). AS mentioned, tThe SDN forwarding decision action is a simple wildcard matching, supported withby OF1.2+, with the wildcard representing the unique bit of a switch-specific output port. With that, the switch needs to consider as many forwarding rules as switch local output ports - see [Reed2016] for more information. Fig. xx illustrate this forwarding solution, including the ability to create ad-hoc multicast relations by simply ORing individual bitarrays representing unicast paths.

- Another approach is outlined in [I-D.ietf-bier-use-cases] where the SFF is suggested to be realized via a BIER overlay, in turn realized over a BIER-compliant underlay, such as MPLS. BIER utilizes a similar bit array approach for representing a

forwarding path in the overlay network but unlike [Reed2016], the bit fields indicate the egress BIER-compliant router that the packet is supposed to reach.

- As yet another alternative, the tSFF may utilize a flow aggregation approach, outlined in [Khalili2016], called edge switch classification (ESC). In this approach, a path from an ingress to egress NAP is described as a so-called edge classification vector (ECV), which combines information on the aggregated flow (following [Khalili2016]) and the switch-local endpoint. The representation has similar bitarray characteristics as the previous two approaches

- NOTE: with the ingress and egress NAPs terminating SF Layer 3 connections and the utilization of bitarray-based tSFFs, the transmission of packets can effectively take place as an ad-hoc Layer multicast while the SFC itself is denoted as an n-times unicast SFC. As an example, consider the chaining of a set of n clients to a single video server. Each sub-SFC from an individual client to the video server will semantically result in a unicast response from the server back to the client (e.g., carrying the video chunk for a MPEG DASH-based video stream). When combining the sub-SFCs to the single SFC with n times unicast relations to the server, the SRR will deliver the responses from the server via one or more multicast responses to one or more clients. The size of the individual multicast groups will depend on the synchronicity of the client requests (and therefore on the synchronicity of the server responses). Note that the multicast relations here are ad-hoc created by ORing the bitarrays representing the specific clients to which the responses are meant to be sent. This is illustrated in the figure below. The HTTP multicast use case is being presented in the BIER use case draft [I-D.ietf-bier-use-cases]albeit without specific a SFC relation.

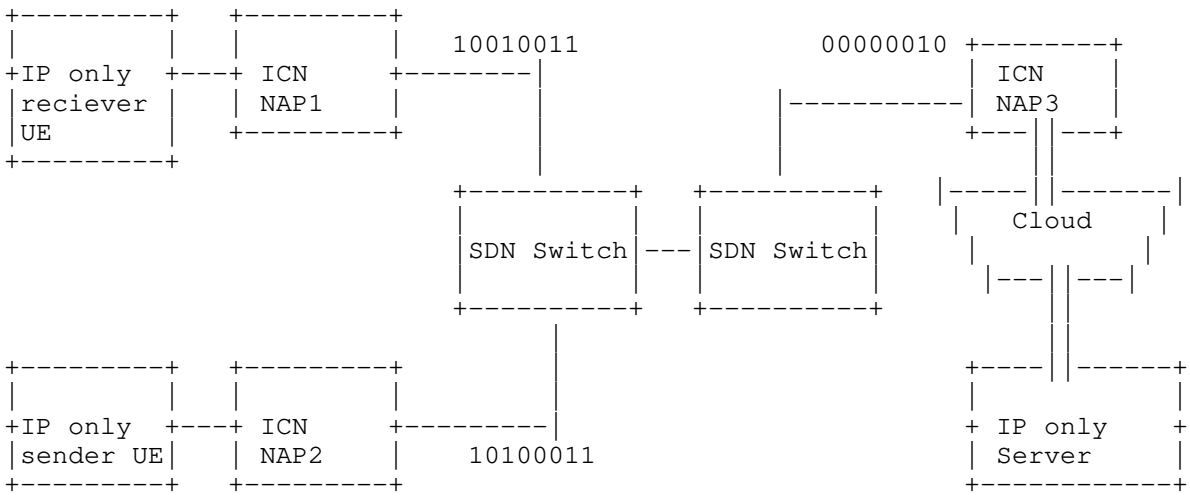


Figure 7: Illustration of Bitfield-based Forwarding using SDN

7. Protocol Consideration

For the operations outlined in the previous section, we foresee the following protocol changes are required:

- NAP-to-NAP protocol for HTTP: HTTP based message exchange between client and server NAPs
- NAP-PCE protocol: Used for path computation, obtaining routing information as well as provide path updates
- Overlay transport protocol: Used for transport-level exchange over any underlay network
- Registration protocol: Used to register FQDN service endpoints
- Content certificate distribution protocol: Used for HTTPS support

8. IANA Considerations

This document requests no IANA actions.

9. Security Considerations

TBD.

10. Informative References

[ETSI_MEC]
ETSI, "Mobile Edge Computing (MEC), Technical Requirements", GS MEC 002 1.1.1, March 2016, <http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf>.

[I-D.ietf-bier-use-cases]
Kumar, N., Asati, R., Chen, M., Xu, X., Dolganow, A., Przygienda, T., arkadiy.gulko@thomsonreuters.com, a., Robinson, D., Arya, V., and C. Bestler, "BIER Use Cases", draft-ietf-bier-use-cases-05 (work in progress), July 2017.

[I-D.ietf-sfc-dc-use-cases]
Kumar, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", draft-ietf-sfc-dc-use-cases-06 (work in progress), February 2017.

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-27 (work in progress), October 2017.

[Khalili2016]
Khalili, R., Poe, W., Despotovic, Z., and A. Hecker, "Reducing State of SDN Switches in Mobile Core Networks by Flow Rule Aggregation", ICCCN, August, 2016.

[Reed2016]
Reed, M., Al-Naday, M., Thomas, N., Trossen, D., and S. Spirou, "Reducing State of SDN Switches in Mobile Core Networks by Flow Rule Aggregation", ICC 2016, 2016.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

Authors' Addresses

Debashish Purkayastha
InterDigital Communications, LLC
Conshohocken
USA

Email: Debashish.Purkayastha@InterDigital.com

Akbar Rahman
InterDigital Communications, LLC
Montreal
Canada

Email: Akbar.Rahman@InterDigital.com

Dirk Trossen
InterDigital Communications, LLC
64 Great Eastern Street, 1st Floor
London EC2A 3QR
United Kingdom

Email: Dirk.Trossen@InterDigital.com
URI: http://www.InterDigital.com/

Zoran Despotovic
Huawei

Email: Zoran.Despotovic@huawei.com
URI: http://www.huawei.com/

Ramin Khalili
Huawei

Email: Ramin.khalili@huawei.com
URI: http://www.huawei.com/

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 2, 2018

D. Purkayastha
A. Rahman
D. Trossen
InterDigital Communications, LLC
Z. Despotovic
R. Khalili
Huawei
March 1, 2018

Alternative Handling of Dynamic Chaining and Service Indirection
draft-purkayastha-sfc-service-indirection-02

Abstract

Many stringent requirements are imposed on today's network, such as low latency, high availability and reliability in order to support several use cases such as IoT, Gaming, Content distribution, Robotics etc. Networks need to be flexible and dynamic in terms of allocation of services and resources. Network Operators should be able to reconfigure the composition of a service and steer users towards new service end points as user move or resource availability changes. SFC allows network operators to easily create and reconfigure service function chains dynamically in response to changing network requirements. We discuss a use case where Service Function Chain can adapt or self-organize as demanded by the network condition without requiring SPI re-classification. This can be achieved, for example, by decoupling the service consumer and service endpoint by a new service function proposed in this draft. We describe few requirements for this service function to enable dynamic switching between consumer and end point.

Table of Contents

1. Introduction

The requirements on today's networks are very diverse, enabling multiple use cases such as IoT, Content Distribution, Gaming, Network functions such as Cloud RAN. Every use case imposes certain requirements on the network. These requirements vary from one extreme to other and often they are in a divergent direction. Network operator and service providers are pushing many functions towards the edge of the network in order to be closer to the users.

This reduces latency and backhaul traffic, as user request can be processed locally.

It becomes more challenging when network congestion, user mobility as well as non-deterministic availability of compute and storage resources are considered. The impact is felt most in the edge of the network because as the users move, their point of attachment changes frequently, which results in (at least partially) relocating the service as well as the service endpoint. Furthermore, network functions are pushed more and more towards the edge, where network, compute and storage resources are constrained and availability is non-deterministic. Constrained network resources may lead into congestion in the network. Also, storage resources may need to be moved where the user concentration is more in case of content delivery applications.

We describe few use cases in the next section and derive the requirement for composing new services and service path in a dynamic edge network. We address this dynamicity by introducing a special Service Function, called SRR (service request routing). We describe the problems associated with today's network and Layer 3 based approach to handle dynamicity in the network. We then discuss how such new Service Function with certain capabilities can handle the dynamicity better than these conventional methods.

2. Use Case Description

2.1. Data Center

The data center use case draft [I-D.ietf-sfc-dc-use-cases] describes an East West traffic use case. This is the predominant traffic in data centers today. Server virtualization has led to the new paradigm where virtual machines can migrate from one server to another across the data center. This explosion in east-west traffic is leading to newer data center network fabric architectures that provide consistent latencies from one point in the fabric to another.

SFCs applied in an enterprise or service provider data center can be broadly categorized into two types:

- Access SFCs
- Application SFCs

Access SFCs are focused on servicing traffic entering and leaving the data center while Application SFCs are focused on servicing traffic destined to applications. Service providers deploy a single "Access SFC" and multiple "Application SFCs" for each tenant. Enterprise

data center operators on the other hand may not have a need for Access SFCs depending on the size and requirements of the enterprise.

In carrier networks, operators may deploy multiple data centers dispersed geographically. Each data center may host different types of service functions. For example, latency sensitive or high usage service functions are deployed in regional data centers while other latency tolerant, low usage service functions are deployed in global or central data centers. In such deployments, SFCs may span multiple data centers and enable operators to deploy services in a flexible and inexpensive way.

It is clear that within the data center as well as in inter data center scenarios, users are serviced by multiple SFs distributed inside as well as outside a location. In this scenario, it is clear that Service function chains should be able to reselect, redirect traffic very fast. The draft identifies that Static service chains do not allow for modifying the SFCs as they require the ability to add SNs or remove SNs to scale up and down the service capacity. Likewise the ability to dynamically pick one among the many SN instance is not available.

2.2. Third party cloud service provider

This use case is related to an emerging business model, where computational resources for edge cloud service are provided by alternative facility providers that are non-traditional network operators. This is due to the situation for many specific localized use cases, where network operators may not have necessary real estate available. They may even not be willing to spend on CAPEX and OPEX for said point-of-presence, because there is no clear path for sustainable cost recovery [UKNIC].

The industry is witnessing the emergence of real estate owners such as building asset or management companies, cell tower owners, railway companies or other facility owners willing to deploy edge cloud resources. The facility provider, e.g. cell tower owner or building management company, deploys edge computing resources throughout their installation in the country. They have their own operation and management software, which is capable of resource deployment, scale up or scale down resources, deploy edge applications from third party service providers . They are capable of offering service to more than one network operator at a specific location, thus acting as a "neutral host". The facility provider, which owns cloud resources and provides application services, is referred to as "Third party Edge Owner (TEO)".

There is more than one stakeholder in this ecosystem, E.g. Network Service Provider, Real estate owner, Cloud capability (compute and storage resource) provider, Application/service provider. An entity can assume more than one role. From network operators point of view there may be "Cloud provider" or "Cloud service provider" depending on the roles assumed by external entity.

"Cloud Providers" provide cloud resources (compute and storage) to network operators. Network operators rent those resources and manage MEC host by themselves. Network operator can set up application traffic rules, so that traffic can be processed, by that host.

"Cloud Service Providers" not only make resources available to network operators or service providers, but also provides management and hosting service. They can host edge applications on behalf of application service providers and sets up user plane traffic to be steered towards the edge application.

Cloud Service Providers, as well as many organizations that need to share and analyze a quickly growing amount of data, such as retailers, manufacturers, telcos, financial services firms, and many more, are turning to localized Micro Data Centers(MDC) installed on the factory floor, in the telco central office, the back of a retail outlet, etc. The solution applies to a broad base of applications that require low latency, high bandwidth, or both.

As Micro Date centers are deployed at the edge of the network, common deployment options are:

- Micro Data Centers are deployed on L2 in the edge of the network
- Instead of single internet Point Of Presence (POP) deployment, multiple internet POP deployment is desirable to localize data
- Service is composed out of these multiple POP deployment of MDC, where data exchange and collaboration is expected among these MDCs
- Due to mobility, changes in network condition (e.g. congestion, load), service composition may change frequently to support promised quality of experience

2.3. ETSI MEC USE CASE

Take the following video orchestration service example from ETSI MEC Requirements document [ETSI_MEC]. The proposed use case of edge video orchestration suggests a scenario where visual content can be produced and consumed at the same location close to consumers in a densely populated and clearly limited area. Such a case could be a

sports event or concert where a remarkable number of consumers are using their handheld devices to access user select tailored content. The overall video experience is combined from multiple sources, such as local recording devices, which may be fixed as well as mobile, and master video from central production server. The user is given an opportunity to select tailored views from a set of local video sources.

2.4. 3GPP

3GPP Rel. 15 introduces the notion of the service-based interface (SBI) as an alternative to the traditional call pattern invocation of network functions. This introduction targets the support for replication, e.g., driven by virtualized functions, as well as supporting alternative interactions, e.g., for different vertical market specific control planes, by making the discovery as well as composition of new interactions more flexible.

We believe that SFC is a suitable framework for the interconnection of such network functions through the new SBI. One of the aforementioned driving forces, namely the replication of functions aligns with our thinking in this draft in that indirections to new vertical instances need to be dynamic in reacting to the appearance of new virtual instances or to changes in policies for the selection of specific instances by specific calling entities.

2.5. Use Case Analysis

SFC allows network operators as well as service providers to compose new services by chaining individual service functions.

In a dynamic network environment, like the edge of a network, the capability to dynamically compose new services from available services as well as move a service instance is desirable. Dynamic composition and relocation of services may be attributed to:

- Congestion in the network: Due to constrained network resources, increase in the network load may create congestion in the network, resulting in a congested Service Function Path. Service functions may detect congestion and reconfigure the Service Function Path to avoid it.

- In response to latency: in a dynamic network environment and with the need for ultra-low latency communication, instantiation of new service function endpoints might be the only remedy to combat the increase of latency caused, e.g., by increased load on a previous endpoint or mobility of the user and therefore increasing the 'distance' to the service function endpoint. Keeping the service

function endpoint 'close' to the user allows for reducing latency, segregating communication in localized islands of service interaction.

- In response to user mobility: In a dynamic network environment where service functions move frequently because of user movement, load balancing or resource modification, service function chains and the service end points need to be created and recreated frequently

- Resource availability.: Availability of compute and storage resources varies with network load, number and type of applications running etc. In the edge of the network, due to sudden increase of users, compute load may increase. In this situation applications, running on the compute resources may be moved to another location where more resources are available.

In SFC, there is a notion of logical chaining of SFs and chaining of actual physical locations, known as Rendered Service Path (RSP). RSP provides a static binding of SFs to their physical location. In order to create a chain in dynamic fashion, late binding of SFs and physical location may be desired. SFC is capable of modifying the service chain to certain extent in response to network conditions, but not a complete solution has been described

In order to route the service requests to service end points in a dynamic manner, we identify the following desirable features in a service function chain:

- Capability to trigger service chain reconfiguration based on network information such as congestion indication, mobility, degradation of user experience etc. Service Functions should be able to process such network information, identify which section of the chain needs to be reconfigured and take action

- Fast switching from one service instance to another by not relying on the DNS for service location resolution. Instead of DNS, the function should be able to identify the path, which will allow to reach the service end point.

- Direct path mobility, where the path between the requester and the responding service can be determined as being optimal (e.g., shortest path or direct path to a selected instance), is needed to avoid the use of anchor points and further reduce service-level latency

- Indirect service requests at the network level, transparent to the requesting client and without the involvement of the DNS. End user is not aware of the decision made by the SF.

- New methods for forwarding, such as path-based forwarding, direct path routing in mobility cases, path pinning for traffic steering and simplified service-specific peering towards the Internet.

3. NSH and Re-classification

[RFC7498] captures the problems associated with existing service deployments that are problematic. The problems are described below at a high level:

- Network topology: Network service deployment is tightly coupled with network topology thus reducing the flexibility in service delivery. It adds complexity in deploying network service when certain traffic types may need some service and other traffic types do not need the same service.

- Configuration complexity is the direct result of dependency on network topology.

- Limited availability of services

- Altering the order of a deployed chain is complex and cumbersome

- Coupling of service functions to topology may require service functions to support many transport encapsulations or for a transport gateway function to be present.

- In a dynamic environment like the Edge of a network service delivery, routing changes fast. It may be difficult to deliver service dynamically due to the risk and complexity of VLANs and/or routing modifications.

These factors provide motivation for a simplified and flexible service insertion model that addresses many of the current shortcomings and provides new, much needed functionality to enable service deployments in modern network environments. Service chaining accomplishes this by considering service functions as resources, with associated attributes, available for scheduled consumption. Selective traffic, subject to policy, may then be "steered" to the requisite service resources, along with any "extra" information referred to as metadata. This metadata is used for policy enforcement.

A basic form of service chaining may be realized using existing transport encapsulations. This method of chaining relies upon the tunneling of selected data between service functions. Although this form of service chaining achieves some level of abstraction from the underlying topology, it does not truly create a service plane. NSH [RFC8300] is a distinct identifiable plane that can be used across all transports to create a service chain and exchange metadata along the chain.

Fundamentally, however, the notion of "services" in SFC is tied into specific service function endpoints, which lie along a well-defined service function path (SFP) where the path is defined through lower layer transport encapsulations. If any such service function endpoint changes, the service chain needs to be adjusted; a procedure we outline in the following sub-section.

3.1. Dynamic service chain creation using NSH

We revisit the dynamic service chain creation capability of NSH. NSH defines a new service plane protocol [RFC8300]. A Network Service Header (NSH) contains service path information and optionally metadata that are added to a packet or frame and used to create a service plane. A control plane is required in order to exchange NSH values with participating nodes, and to provision the same nodes with requisite information such as service path ID to overlay mapping.

The Network Service Header has three parts, Base header, Service Path Header and Context Header. NSH Service Path Header is a 4-byte service path header follows the base header and defines two fields used to construct a service path:

- Service path identifier (SPI)
- Service index (SI)

The following figure depicts the service path header.

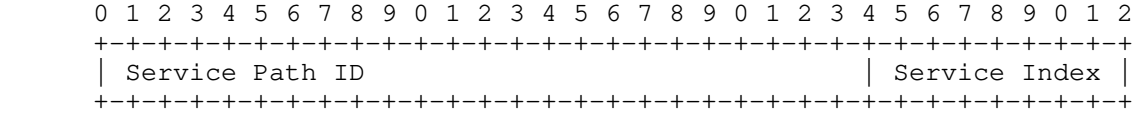


Figure 1: NSH Path Header

The service path identifier (SPI) is used to identify the service path that interconnects the needed service functions. It allows nodes to utilize the identifier to select the appropriate network

transport protocol and forwarding techniques. The service index (SI) identifies the location of a packet within a service path. As packets traverse a service path, the SI is decremented post-service.

SPI represents the service path and altering the path identifier results in a change of a service path. A change in SPI value is a result of re-classification. It means a node in the service path determined, based on policy, that the initial classification was incorrect or incomplete. If the updated classification results in the necessity of a new service path, the node updates the SPI and SI fields accordingly. The new identifier is then used to select the appropriate overlay topology. This allows service functions to alter the path of a packet without having to participate in the network topology and its associated control plane(s). The method to determine that an existing classification is incorrect and how to determine the new classification is not defined.

4. Challenges with dynamic indirection

The emerging trend in today's network is to deploy network functions, services and applications at the edge of the network to support latency requirements, computational offload, traffic optimization etc. As users are moving, application or services being used by users, may need to be moved closer to the user's new location. This implies another instance of the service function may need to be instantiated close to the user's new location. It may result in re-establishing service path from the newly instantiated service function to other service instances. It is also possible that the newly instantiated service function may be redirected to a new service end point (e.g. Application Server) for various reasons, such as incomplete content, proximity to data store, load balancing etc. In another scenario, a single instance of the service function may not handle all users due to latency or load constraints. A single service function may be instantiated more than once to balance user load. As the number of instances increase and along with mobility, the complexity of service routing increases. It is anticipated that there may be a constant action of function chaining, re-chaining occurring in the network.

The challenge of dynamic indirection may be better described by analyzing the working of CDNs, which dynamically (re-)direct user-initiated requests towards the most appropriate content instance. This task becomes more difficult if granularity of the instance placement increases. For instance, in case of a CDN being realized close to end users, specifically in edge of the network, the specific content instance might need to be selected dynamically. After initial selection, the instance may change during service execution.

In a conventional network, an instance of a service is found and selected using DNS. The subsequent service request is then routed through the network between the client and the service. If the user is doing a DNS lookup to access content served by a CDN then the DNS service will maintain a list of IP addresses that can be returned for a given domain name and will try to return an IP address of a node geographically close to the client. Should the service provider want to replace an instance of their service with another one at a different IP address (and potentially a different physical location for various reasons such as load balancing, reliability etc.) then the DNS tables must be updated, i.e., the service needs to be (re-)registered quickly. This is done by updating the local authoritative DNS server which then propagates the new mapping to DNS services across the world. DNS propagation can take up to 48 hours so fast and dynamic switching from one service instance to another is not possible in conventional networks; even in more localized scenarios, the propagation of DNS updates might still be insufficient. When relying on many surrogate service endpoints to exist in the edge network, there is a clear issue of certain resources not being available in one surrogate instance while existing in another so that changes in redirection might be desirable, while also changes in local load drive the need for such change in redirection. With the emergence of container-based virtualization platforms, service function endpoints can be established in a matter of seconds and we therefore believe that the 'reachability' of such said service instance, i.e., the possibility of route service requests to it from a client that was previously served elsewhere, must follow a similar timeline, i.e., a few seconds or even less.

The other issue in conventional network lies with mobility management procedure. These procedures use an anchor point, which terminates a session at the network edge. As user moves around, traffic is redirected from the anchor point to the new point of attachment. Relying on typical mobility management approaches found in IP networks, usually leads to inefficient 'triangular' routing of requests through this common 'anchor' point. This triangular routing increases the latency in reaching the new service function or service end points as users move.

Traffic steering is a common procedure in managed networks, particularly at the edge, due to desired subscriber-centric traffic policies (e.g., related to pricing structures), resource requirements (e.g., related to using particular paths in the network) or mobility (e.g., users moving in a cellular network). Today's methods for traffic steering include anchor-based mobility management as well as traffic classification, for instance, in packet gateways of cellular systems (using, e.g., deep packet inspection as well as port and

address classification). While the former leads to inefficient 'triangular' traffic forwarding, the latter often requires additional state in the forwarders to differentiate traffic from one user to another.

The analysis of CDN network shows that dynamic indirection is a necessary requirement, which needs to be supported by the networks. The goal for this indirection is to provide user applications lowest possible latency. But as discussed above, relying on today's technique does not help in guaranteeing same latency to user applications. On the other hand, there is a high possibility that latency may increase if we rely on Layer 3 based service redirection techniques.

SFC handles indirection through the use of SPI. A packet needs to be reclassified and the intermediate node changes the SPI. Following are the typical steps that happens in order to implement the indirection.

- A packet arrives at a particular node
- The node contacts the policy manager
- Identifies the current classification is incorrect
- Reclassifies the packet, i.e. change the SPI
- Inserts the packet in the pipe, possibly towards the SFF

The indirection mechanism in SFC involves certain steps to process policy information and change the SPI in the packet header, making it suitable to handle dynamic indirection requirements. Our proposed SF in this document provides an additional method to handle dynamic indirection of service requests, not relying on the reclassification mechanism. Combining these two techniques may provide flexibility and improvement over single method.

5. HTTP as a transport

With the extensive use of "web technology", "distributed services" and availability of heterogeneous network, HTTP has effectively transitioned into the common transport for name-based E2E communication across the web. In the context of SFC and SF, HTTP requests and response are considered as the "Service Request (SR)". This use case describes how these SRs are directed towards correct SF in a fast and dynamic way. The routing and indirection of SRs are abstracted at HTTP level, instead of the traditional approach where routing decision for a service request is made at Layer 3.

If we abstract HTTP as a transport, HTTP requests, such as GET, PUT and POST can be routed based on the URI associated with the request, with the URI being simply the name of a resource or the invocation point for a service transaction. Based on the name of the resource requested, the appropriate HTTP request can be routed to the suitable service endpoint. If Service Functions (SF) could be identified using URI or name, HTTP requests to an SF would be routed or directed using name based routing. With that, the redirection to the most suitable service instance is purely done based on named services with HTTP being a specific (application layer) transport service.

The ongoing EU H2020 efforts like FLAME [H2020FLAME] are driven by city-scale many-POP deployments of compute infrastructure, all SDN-connected and OpenStack managed. Localized media use cases drive the need for name-based (HTTP as the main transport protocol here) service instances being chained with the relationship between specific virtual instances being controlled at the underlying routing/switching level.

The notion of 'HTTP as-a transport', utilizing URLs as addressing scheme, can be used to create SFP as shown in Fig 2., i.e., 192.168.x.x -> www.example.com -> 192.168.x.x -> www.example2.com -> 192.168.x.x -> ... -> www.exampleN.com. It is this 'name-based' relationship that we see possibly realized through specific replicated instances, where in turn the routing towards those specific instances is realized by the SRR.

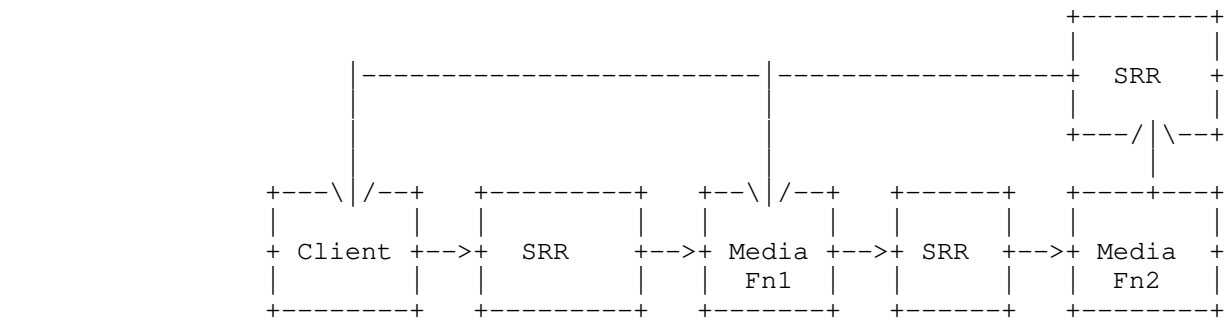


Figure 2: SFP with new HTTP-based Transport option

In a pure SFC architectural framework, Classifier function may interact with SRR to obtain an SE (Service Encapsulation). E.g. the

Classifier function may look into the network locator map in Fig 2 and determine the next SF is www.example.com. It provides this information to SRR to obtain the next hop information. SRR returns the SE for next hop, which can be a "bitfield" information that is being used in the overlay routing for this part of the SFP. The Classifier function uses this SE to route the incoming packet directly at the transport network level.

6. Service Request Routing (SRR) Service Function

6.1. Overview

The following diagram shows the application of the new proposed SRR service function in an example of media clients connecting to media servers. There may be more than one media functions to support CDN like architecture, Surrogate servers to handle mobility and load balancing.

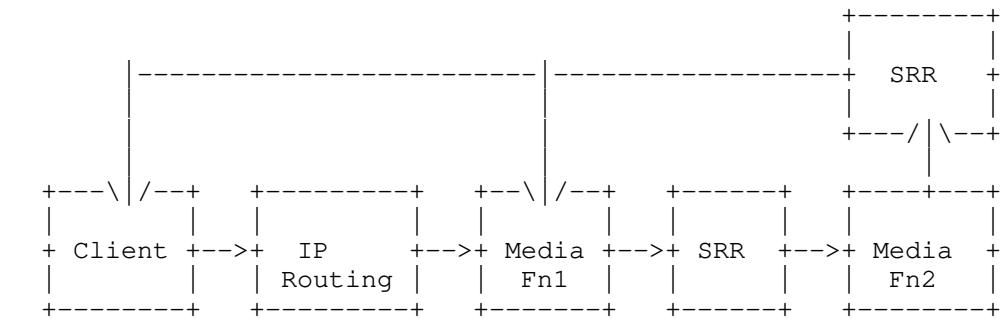


Figure 3: General SFC with SRR Flexible Chaining, initiated via IP Routed Client Connection

The clients are connected to media functions through frontend routed network, e.g., relying on standard IP routing, while media functions are chained via the new proposed service request routing (SRR) function. Alternatively, we also envision to utilize the SRR function directly between client SF and media function SF, as outlined in the figure below

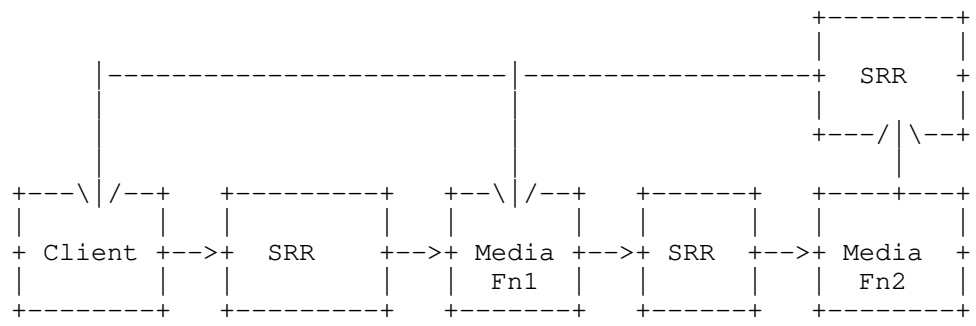


Figure 4: General SFC with SRR Flexible Chaining, initiated via SRR Chained Client

For our considerations, we assume that each SF is realized by at least one or more service function endpoints (SFEs). Hence, instead of looking at "chaining" as a concept that connects specific SFEs along a well-defined SFP, we propose to look at "chaining" at the level of "named" service functions rather than their specific endpoint instances. With this in mind, the SRR service function lifts the relationship between the connecting SFs to the level of "logical" service functions rather than their specific realizing endpoints. Instead of relying on dynamic re-chaining in case of any dynamically changing relationship between specific SFEs, the SRR provides the selection of suitable SFEs while maintaining the logical relationship between the SFs. In Section 6.3, we will present the necessary extensions to the SFP concept to support this higher abstraction of "chaining" via "named" logical SFs. The SRR introduces the flexibility in routing service requests from client to specific SFEs. In the edge network, where users are moving and service end points may also change, having flexibility to decide and steer service requests directly helps in guaranteeing the same latency to user applications. Clearly, that is achieved by reducing the switching time from SF to another. As service end point changes, the routing functions makes instantaneous decision to route the request to the appropriate media server.

The SRR introduces the flexibility in routing service requests from client to specific SFEs in response to conditions such as congestion in the network, user mobility etc. In the edge network, where users are moving and service end points may also change, having flexibility to decide and steer service requests directly helps in guaranteeing the same latency to user applications. The edge of the network maybe congested due to limited network resources. The SRR may be able to determine network congestion and quickly route service requests to other Service End point, which is not experiencing congestion. In

addition, application-layer control functions might utilize latency measurements to ensure that suitable service instances are being created during runtime of the scenario such as to ensure that service function endpoints are available 'nearby' (possibly) moving so as to keep a desired latency under a desired value.

Clearly, that is achieved by reducing the switching time from one SF endpoint to another. As the service end point changes, the routing functions makes instantaneous decision to route the request to the appropriate media server.

The possible improvements of using SRR within an SFC framework are listed below:

- Fast (between 10 and 20ms) switching times from one service instance to another by not relying on the DNS for service discovery and directly routing service requests at the level of the transport network.

- The capability to indirect service requests at the network level will help in reducing latency, when service end points change. E.g. when a service request is being sent to one surrogate instance but results in a HTTP 404 or 5xx error response, the original request is redirected to another alternative surrogate with minimal latency, i.e., right at the destination of said failed service request. Nesting these operations effectively leads to a net-level 'search' among all available surrogate instances until the search is exhausted (with a negative result) or the resource is found.

- New methods for forwarding, such as path-based forwarding, will enable direct path routing in mobility cases, path pinning for traffic steering and simplified service-specific peering towards the Internet. Such capability would allow for localizing traffic, reduce latency and costs.

6.2. Details of SRR Function

Assuming such introduction of an HTTP-level transport notion, the SRR function can be decomposed further as shown in Fig 5.

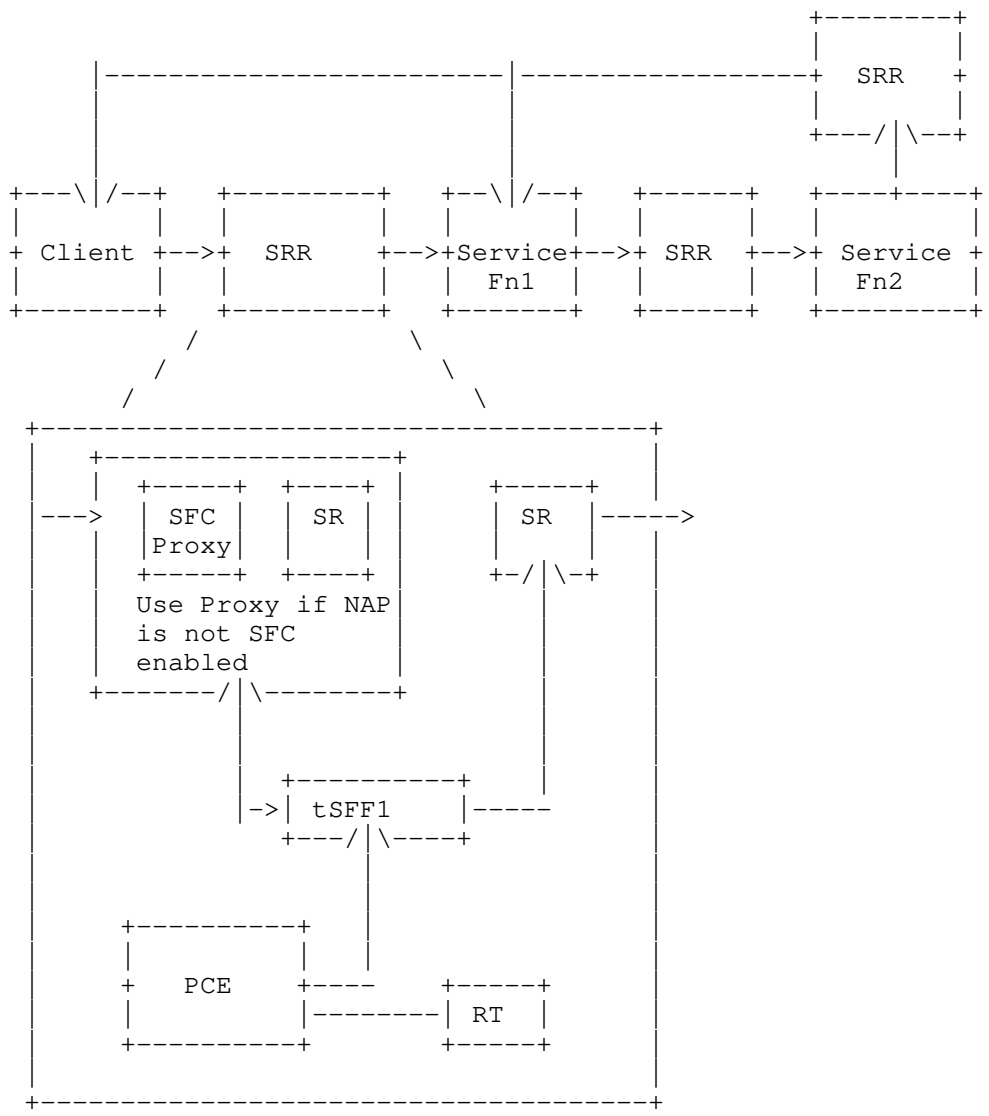


Figure 5: SRR decomposition

Another option for the two functions routing via the SRR could be entirely link-local, i.e., there's another simple tSFF2 between client and SRR as well as SF1 and SRR that is simply a link-local transport. The following figure describes this alternate option.

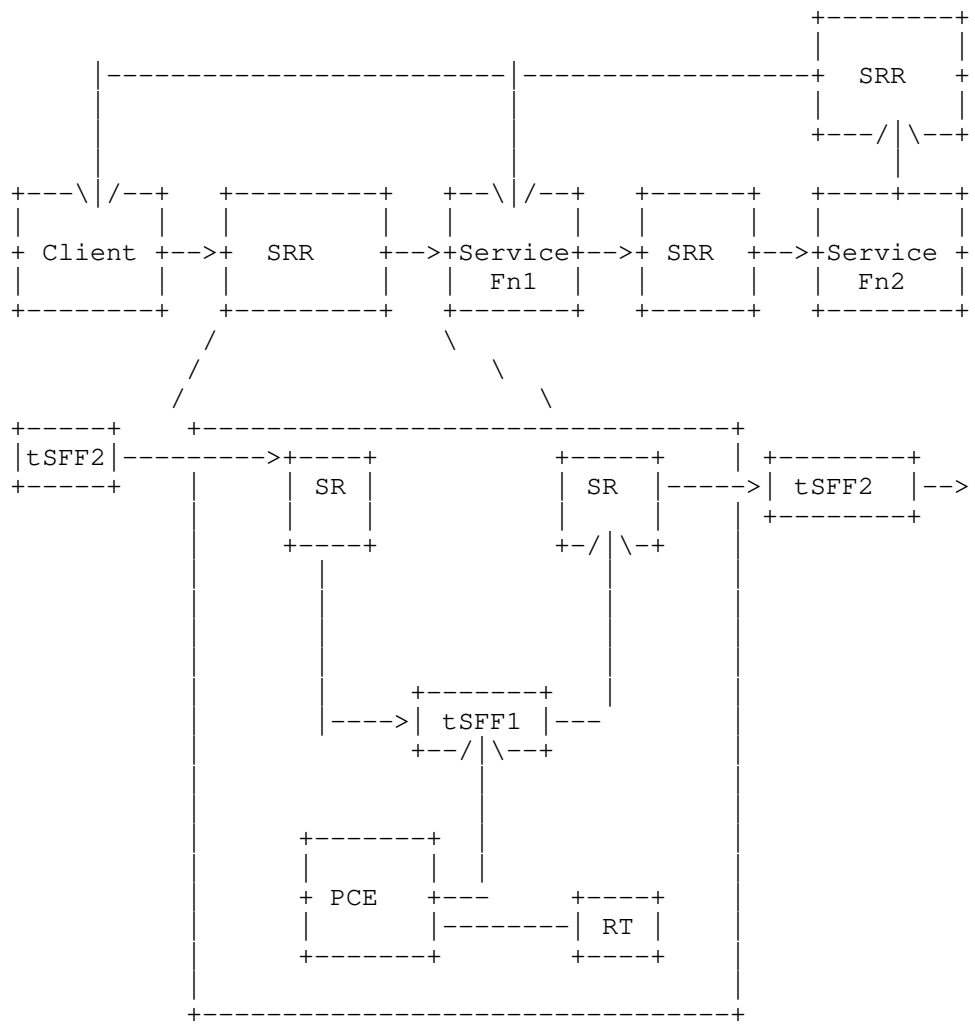


Figure 6: SRR decomposition using link-local client/function communication

The SRR function may be composed of the following functions:

- Service Router(SR) at the ingress, terminates on the client side Layer 3 and above protocols, such as TCP

- Service Router(SR) at the egress, terminates any transport protocol on the outgoing (server) side

- PCE, Path Computation Element function is responsible for selecting the correct next SF, also possibly realizing path policy enforcement. The result of the selection is a path identifier which is delivered to the ingress SR upon initial path computation request (i.e., when sending a request to a specific URL on the SFP for the first time). The path identifier is utilized for any future request for a given URL-based SF. In case of another SF instance becoming available, indicated to the PCE through a registration procedure, the PCE will instruct all ingress SRs to invalidate path identifiers to the specific URL of the SF, resulting in an initial path computation request at the next SF request forwarding. Through this, the newly registered SF instance might be utilized if the policy-governed path computation will select said SF instance.

- Reclassification Trigger Handler (RT) : Network measurement information, such as latency, packet loss or network congestion, etc. could be processed by the handler. This may trigger reconfiguration of the specific service function endpoint chain over which the SFC is being executed. The handler forwards the information about the chain reconfiguration to PCE.

- Transport-derived SFF (tSFF1): the communication between ingress/ egress SRs as well as SRs to PCE is realized via a transport-derived SFF. We outline here three possible tSFFs

  * SDN-based: This option utilizes path-based forwarding through SDN-based wildcard matching fields, supported with OF1.2+[Reed2016]. It can be embedded into slicing approach of underlying transport infrastructure by leaving typical slicing fields available (e.g., VLAN tags). The forwarding utilizes the Ethernet frame format at Layer 2, representing the topological links of a specific forwarding path in the transport network as unique bits in a fixed size bit array. For the latter, the approach utilizes the IPv6 source and destination fields for storing the bit array information (in a simple version for this forwarding, this limits the topology to 256 links but extensions schemes are possible, which are left out of this document at this stage). AS mentioned, the SDN forwarding decision action is a simple wildcard matching, supported with OF1.2+, with the wildcard representing the unique bit of a switch-specific output port. With that, the switch needs to consider as many forwarding rules as switch local output ports - see [Reed2016] for more information. Fig. xx illustrate this forwarding solution, including the ability

to create ad-hoc multicast relations by simply ORing individual bitarrays representing unicast paths.

* Another approach is outlined in [I-D.ietf-bier-use-cases] where the SFF is suggested to be realized via a BIER overlay, in turn realized over a BIER-compliant underlay, such as MPLS. BIER utilizes a similar bit array approach for representing a forwarding path in the overlay network but unlike [Reed2016], the bit fields indicate the egress BIER-compliant router that the packet is supposed to reach.

* As yet another alternative, the tSFF may utilize a flow aggregation approach, outlined in [Khalili2016], called edge switch classification (ESC). In this approach, a path from an ingress to egress SR is described as a so-called edge classification vector (ECV), which combines information on the aggregated flow (following [Khalili2016]) and the switch-local endpoint. The representation has similar bitarray characteristics as the previous two approaches

o NOTE: with the ingress and egress SRs terminating SF Layer 3 connections and the utilization of bitarray-based tSFFs, the transmission of packets can effectively take place as an ad-hoc Layer multicast while the SFC itself is denoted as an n-times unicast SFC. As an example, consider the chaining of a set of n clients to a single video server. Each sub-SFC from an individual client to the video server will semantically result in a unicast response from the server back to the client (e.g., carrying the video chunk for a MPEG DASH-based video stream). When combining the sub-SFCs to the single SFC with n times unicast relations to the server, the SRR will deliver the responses from the server via one or more multicast responses to one or more clients. The size of the individual multicast groups will depend on the synchronicity of the client requests (and therefore on the synchronicity of the server responses). Note that the multicast relations here are ad-hoc created by ORing the bitarrays representing the specific clients to which the responses are meant to be sent. This is illustrated in the figure below. The HTTP multicast use case is being presented in the BIER use case draft [I-D.ietf-bier-use-cases]albeit without specific a SFC relation.

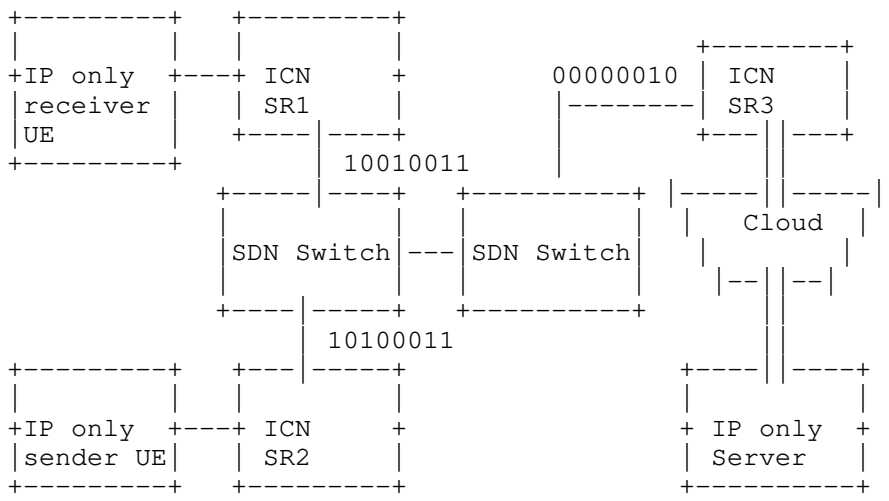


Figure 7: Illustration of Bitfield-based Forwarding using SDN

7. Protocol Consideration

For the operations outlined in the previous section, we foresee the following protocol changes are required:

- SR-to-SR protocol for HTTP: HTTP based message exchange between client and server SRs
- SR-PCE protocol: Used for path computation, obtaining routing information as well as provide path updates
- Registration protocol: Used to register FQDN service endpoints

8. Next Steps

Feedback from the SFC WG on the validity of this solution and its scope within the SFC WG. If such alternative to the re-classification for service indirection is seen beneficial as well as fitting with the charter of the WG, the next steps would be to update the draft to outline potential protocol solutions required for the realization of such SRR SF.

9. IANA Considerations

This document requests no IANA actions.

10. Security Considerations

TBD.

11. Informative References

[ETSI_MEC]
ETSI, "Mobile Edge Computing (MEC), Technical Requirements", GS MEC 002 1.1.1, March 2016, <http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf>.

[H2020FLAME]
EU, "EU H2020 FLAME PROJECT", , March 2016, <https://www.ict-flame.eu/>.

[I-D.ietf-bier-use-cases]
Kumar, N., Asati, R., Chen, M., Xu, X., Dolganow, A., Przygienda, T., Gulko, A., Robinson, D., Arya, V., and C. Bestler, "BIER Use Cases", draft-ietf-bier-use-cases-06 (work in progress), January 2018.

[I-D.ietf-sfc-dc-use-cases]
Kumar, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", draft-ietf-sfc-dc-use-cases-06 (work in progress), February 2017.

[Khalili2016]
Khalili, R., Poe, W., Despotovic, Z., and A. Hecker, "Reducing State of SDN Switches in Mobile Core Networks by Flow Rule Aggregation", ICCCN, August, 2016.

[Reed2016]
Reed, M., Al-Naday, M., Thomas, N., Trossen, D., and S. Spirou, "Reducing State of SDN Switches in Mobile Core Networks by Flow Rule Aggregation", ICC 2016, 2016.

[RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <https://www.rfc-editor.org/info/rfc7498>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

[UKNIC] UK NIC, "5G Infrastructure Requirements in the UK", Final Report 3.0, December 2016, <https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/577940/5G_Infrastructure_requirements_for_the_UK_-_LS_Telcom_report_for_the_NIC.pdf>.

Authors' Addresses

Debashish Purkayastha
InterDigital Communications, LLC
Conshohocken
USA

Email: Debashish.Purkayastha@InterDigital.com

Akbar Rahman
InterDigital Communications, LLC
Montreal
Canada

Email: Akbar.Rahman@InterDigital.com

Dirk Trossen
InterDigital Communications, LLC
64 Great Eastern Street, 1st Floor
London EC2A 3QR
United Kingdom

Email: Dirk.Trossen@InterDigital.com
URI: http://www.InterDigital.com/

Zoran Despotovic
Huawei

Email: Zoran.Despotovic@huawei.com
URI: http://www.huawei.com/

Ramin Khalili
Huawei

Email: Ramin.khalili@huawei.com
URI: http://www.huawei.com/

SFC WG
Internet-Draft
Intended status: Standards Track
Expires: March 25, 2018

G. Mirsky
ZTE Corp.
W. Meng
ZTE Corporation
B. Khasnabish
ZTE TX, Inc.
C. Wang
September 21, 2017

Multi-Layer Active OAM for Service Function Chains in Networks
draft-wang-sfc-multi-layer-oam-10

Abstract

A multi-layer approach to the task of Operation, Administration and Maintenance (OAM) of Service Function Chains (SFCs) in networks is presented. Based on the requirements towards active OAM for SFC, a multi-layer model is introduced. A mechanism to detect and localize defects using the multi-layer model is also described.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 25, 2018.

Copyright Notice

Table of Contents

1. Introduction

[RFC7665] defines components necessary to implement Service Function Chain (SFC). These include a classifier which performs classification of incoming packets. A Service Function Forwarder (SFF) is responsible for forwarding traffic to one or more connected Service Functions (SFs) according to the information carried in the SFC encapsulation. SFF also handles traffic coming back from the SF and transports the data packets to the next SFF. And the SFF serves as termination element of the Service Function Path (SFP). SF is responsible for specific treatment of received packets.

Resulting from that SFC is constructed by a number of these components, there are different views from different levels of the

SFC. One is the SFC, fully abstract entity, that defines an ordered set of SFs that must be applied to packets selected as a result of classification. But SFC doesn't define exact mapping between SFFs and SFs. Thus there exists another semi-abstract entity referred as SFP. SFP is the instantiation of the SFC in the network and provides a level of indirection between the fully abstract SFC and a fully specified ordered list of SFFs and SFs identities that the packet will visit when it traverses the SFC. The latter entity is being referred as Rendered Service Path (RSP). The main difference between SFP and RSP is that in the former the authority to select the SFF/SF has been delegated to the network.

This document proposes the multi-layer model of SFC active Operation, Administration and Maintenance (OAM), per [RFC7799] definition of active OAM, lists requirements to improve the troubleshooting efficiency and defines SFC Echo request and Echo reply that enables on-demand Continuity Check, Connectivity Verification among other operations over SFC in networks.

2. Conventions

2.1. Requirements Language

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

2.2. Terminology

Unless explicitly specified in this document, active OAM in SFC and SFC OAM are being used interchangeably.

e2e: End-to-End

FM: Fault Management

NSH: Network Service Header

OAM: Operations, Administration, and Maintenance

RDI: Remote Defect Indication

RSP: Rendered Service Path

SF: Service Function

SFC: Service Function Chain

SFF: Service Function Forwarder

SFP: Service Function Path

3. Multi-layer Model of SFC OAM

As described in [I-D.ietf-sfc-oam-framework], multiple layers come into play to realize the SFC, including the Service layer, the underlying Network layer, as well as the Link layer, which are depicted in Figure 1:

- The Service layer consists of classifiers and/or service functions/SFs.
- Network and Transport layers leverage various overlay network technologies interconnecting SFs to establish SFP.
- The Link layer is technology specific and reflects the technology used in the underlay network.

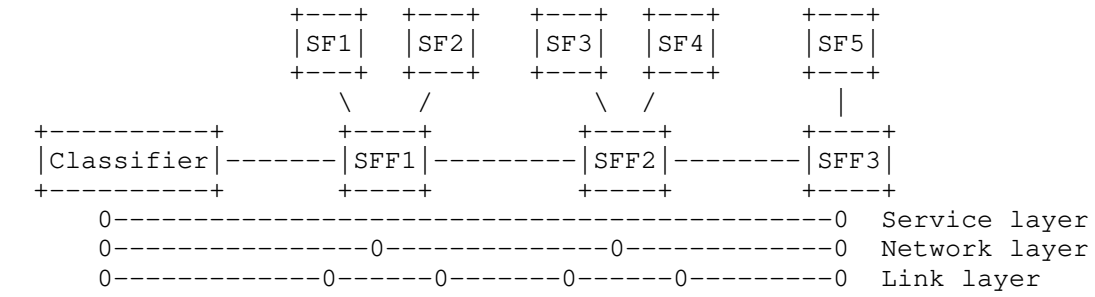


Figure 1: SFC OAM Multi-Layer model

4. Requirements for Multi-layer Model of Active OAM

To perform the OAM task of fault management (FM) in an SFC, that includes failure detection, defect characterization and localization, this document defines the multi-layer model of OAM, presented in Section 3, and set of requirements towards active OAM mechanisms to be used on an SFC.

In example presented in Figure 1 the service SFP1 may be realized through two RSPs, RSP1(SF1--SF3--SF5) and RSP2(SF2--SF4--SF6). To perform end-to-end (e2e) FM SFC OAM:

REQ#1: Packets of active OAM in SFC SHOULD be fate sharing with data traffic, i.e. in-band with the monitored traffic, i.e. follow exactly the same RSP, in forward direction, i.e. from ingress toward egress end point(s) of the OAM test.

REQ#2: SFC OAM MUST support pro-active monitoring of any element in the SFC availability.

The egress, SFF3 in example in Figure 1, is the entity that detects the failure of the SFC. It must be able to signal the new defect state to the ingress, i.e. SFF1. Hence the following requirement:

REQ#3: SFC OAM MUST support Remote Defect Indication (RDI) notification by egress to the ingress, i.e. source of continuity checking.

REQ#4: SFC OAM MUST support connectivity verification. Definition of mis-connectivity defect entry and exit criteria are outside the scope of this document.

Once the SFF1 detects the defect objective of OAM switches from failure detection to defect characterization and localization.

REQ#5: SFC OAM MUST support fault localization of Loss of Continuity check in the SFC.

REQ#6: SFC OAM MUST support tracing an SFP in order to realize the RSP.

It is practical, as presented in Figure 1, that several SFs share the same SFF. In such case SFP1 may be realized over two RSPs, RSP1(SF1--SF3--SF5) and RSP2(SF2--SF4--SF6).

REQ#7: SFC OAM MUST have the ability to discover and exercise all available RSPs in the transport network.

In process of localizing the SFC failure separating SFC OAM layers is very attractive and efficient approach. To achieve that continuity among SFFs that are part of the same SFP should be verified. Once SFFs reachability along the particular SFP has been confirmed task of defect localization may focus on SF reachability verification. Because reachability of SFFs has already been verified, SFF local to the SF may be used as source.

REQ#8: SFC OAM MUST be able to trigger on-demand FM with responses being directed towards initiator of such proxy request.

5. Active OAM Identification in SFC NSH

The multi-layer model OAM that confirms to the above listed requirements enables active OAM protocols that are capable to perform efficient defect localization on an SFC. [I-D.ietf-sfc-nsh] does not provide definition for identification of an SFC active OAM packet. This document defines that active OAM packet on SFC MUST have OAM bit set and MUST have the value on the Next Protocol field set to OAM (TBA1) according to Section 9.1.

It is very unlikely that a single protocol will address all the requirements listed in Section 4. Protocols may be identified by destination UDP port number if IP/UDP encapsulation used. But extra IP/UDP headers, especially in case of IPv6, add noticeable overhead. This document defines Active OAM Header Figure 2 to demultiplex active OAM protocols on an SFC.

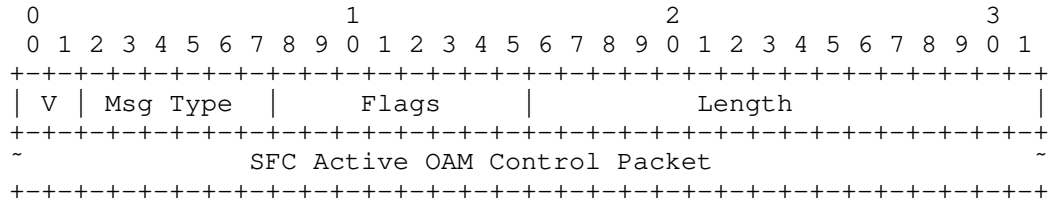


Figure 2: SFC Active OAM Header

V - two bits long field indicates the current version of the SFC active OAM header. The current value is 0.

Msg Type - six bits long field identifies OAM protocol, e.g. Echo Request/Reply or BFD.

Flags - eight bits long field carries bit flags that define optional capability and thus processing of the SFC active OAM control packet, e.g. optional timestamping.

Length - two octets long field that is length of the SFC active OAM control packet in octets.

6. SFC OAM multi-layer model

Figure 3 presents a use case of applying the proposed SFC OAM multi-layer model. In this scenario operator needs to discover SFFs and SFs of the same SFC. The Layer 1 includes the SFFs that are part of the SFP. The Layer 2 - the SFs along the RSP. When trying to do SFC OAM, classifier or service nodes select and confirm which SFC OAM layering they plan to do, then encapsulate the layering information

in the SFC OAM packets, and send the SFC OAM packets along the service function paths to the destination. When receiving the SFC OAM packets, service nodes analyze the layering information and then decide whether sending these packets to next SFFs directly without being processed by SFs for Layer 1 process or sending to SFs for Layer 2 process.

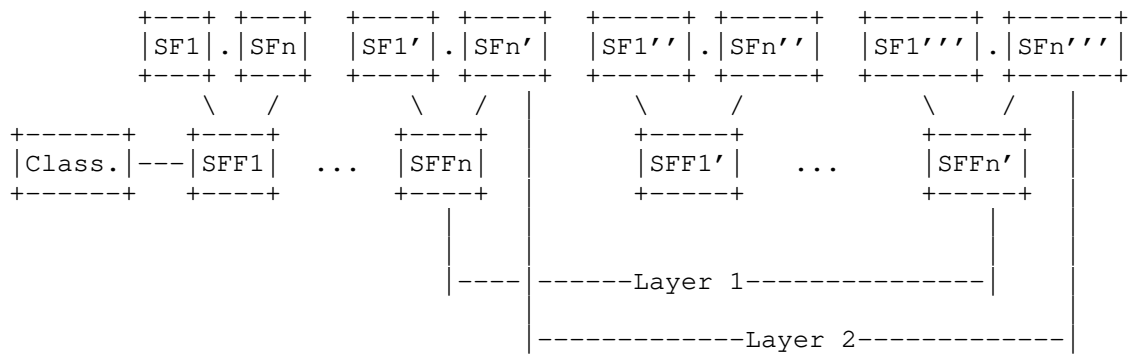


Figure 3: SFC OAM multi-layering model

7. Echo Request/Echo Reply for SFC in Networks

Echo Request/Reply is well-known active OAM mechanism that is extensively used to detect inconsistencies between states in control plane and data plane, localize defects in the data plane. The format of the Echo request/Echo reply control packet is to support ping and traceroute functionality in SFC in networks Figure 4 resembles the format of MPLS LSP Ping [RFC8029] with some exceptions.

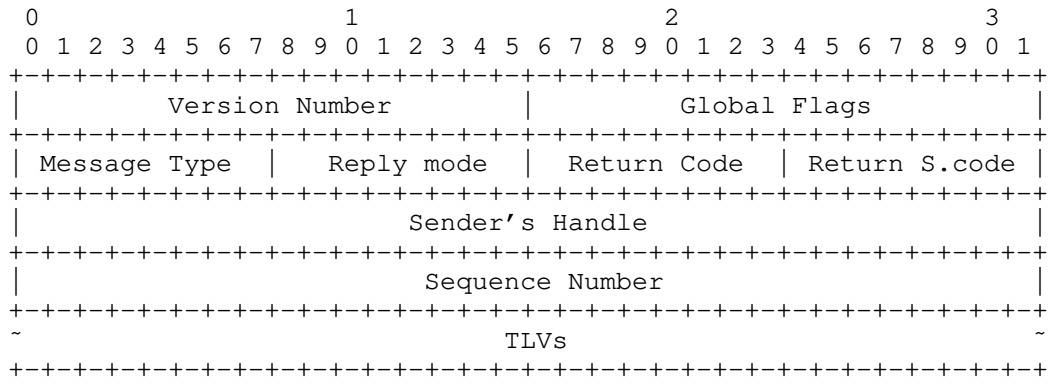


Figure 4: SFC Echo Request/Reply format

The interpretation of the fields is as following:

The Version reflects the current version. The version number is to be incremented whenever a change is made that affects the ability of an implementation to correctly parse or process control packet.

The Global Flags is a bit vector field

The Message Type filed reflects the type of the packet. Value TBA3 identifies echo request and TBA4 - echo reply

The Reply Mode defines the type of the return path requested by the sender of the echo request.

Return Codes and Subcodes can be used to inform the sender about result of processing its request.

The Sender's Handle is filled in by the sender, and returned unchanged by the receiver in the echo reply.

The Sequence Number is assigned by the sender and can be (for example) used to detect missed replies.

TLVs (Type-Length-Value tuples) have the two octets long Type field, two octets long Length field that is length of the Value field in octets.

7.1. SFC Echo Request Transmission

SFC echo request control packet MUST use the appropriate encapsulation of the monitored SFP. If Network Service Header (NSH) is used, echo request MUST set O bit, as defined in [I-D.ietf-sfc-nsh]. SFC NSH MUST be immediately followed by the SFC Active OAM Header defined in Section 5. Message Type field in the SFC Active OAM Header MUST be set to SFC Echo Request/Echo Reply value (TBA2) per Section 9.2.

Value of the Reply Mode field MAY be set to:

- Do Not Reply (TBA5) if one-way monitoring is desired. If echo request is used to measure synthetic packet loss, the receiver may report loss measurement results to a remote node.

- Reply via an IPv4/IPv6 UDP Packet (TBA6) value likely will be the most used.

- Reply via Application Level Control Channel (TBA7) value if the SFP may have bi-directional paths.

- Reply via Specified Path (TBA7) value in order to enforce use of the particular return path specified in the included TLV to verify bi-directional continuity and also increase robustness of the monitoring by selecting more stable path.

7.2. SFC Echo Request Reception

7.3. SFC Echo Reply Transmission

The Reply Mode field directs whether and how the echo reply message should be sent. The sender of the echo request MAY use TLVs to request that corresponding echo reply be sent using the specified path. Value TBA3 is referred as "Do not reply" mode and suppresses transmission of echo reply packet. Default value (TBA6) for the Reply mode field requests the responder to send the echo reply packet out-of-band as IPv4 or IPv6 UDP packet.

Responder to the SFC echo request sends the echo reply over IP network if the Reply mode is Reply via an IPv4/IPv6 UDP Packet. Because SFC NSH does not identify the ingress of the SFP the echo request MUST include this information that to be used as IP destination address for IP/UDP encapsulation of the SFC echo reply. Sender of the SFC echo request MUST include SFC Source TLV Figure 5.

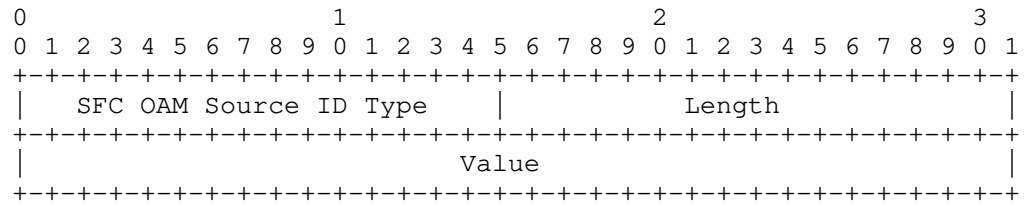


Figure 5: SFC Source TLV

where

SFC OAM Source Id Type is two octets in length and has the value of TBA9 Section 9.6.

Length is two octets long field and the values is equal to the length of the Value field.

Value field contains IP address of the sender of the SFC OAM control message, IPv4 or IPv6.

The UDP destination port for SFC Echo Reply TBA10 will be allocated by IANA Section 9.7.

7.4. Overlay Echo Reply Reception

8. Security Considerations

Overlay Echo Request/Reply operates withing the domain of the overlay network and thus inherits any security considerations that apply to the use of that overlay technology and, consequently, underlay data plane. Also, the security needs for SFC echo request/reply are similar to those of ICMP ping [RFC0792], [RFC4443] and MPLS LSP ping [RFC8029].

There are at least three approaches of attacking a node in the overlay network using the mechanisms defined in the document. One is a Denial-of-Service attack, by sending SFC ping to overload an element of the SFC. The second may use spoofing, hijacking, replying, or otherwise tampering with SFC echo requests and/or replies to misrepresent, alter operator's view of the state of the SFC. The third is an unauthorized source using an SFC echo request/ reply to obtain information about the SFC and/or its elements, e.g. SFF or SF.

To mitigate potential Denial-of-Service attacks, it is RECOMMENDED that implementations throttle the SFC ping traffic going to the control plane.

Reply and spoofing attacks involving faking or replying SFC echo reply messages would have to match the Sender's Handle and Sequence Number of an outstanding SFC echo request message which is highly unlikely. Thus the non-matching reply would be discarded.

To protect against unauthorized sources trying to obtain information about the overlay and/or underlay an implementation MAY check that the source of the echo request is indeed part of the SFP.

9. IANA Considerations

9.1. SFC Active OAM Protocol

IANA is requested to assign new type from the SFC Next Protocol registry as follows:

Value	Description	Reference
TBA1	SFC Active OAM	This document

Table 1: SFC Active OAM Protocol

9.2. SFC Active OAM Message Type

IANA is requested to create new registry called "SFC Active OAM Message Type". All code points in the range 1 through 32767 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126] . Remaining code points are allocated according to the table Table 2:

Value	Description	Reference
0	Reserved	
1 - 32767	Reserved	IETF Consensus
32768 - 65530	Reserved	First Come First Served
65531 - 65534	Reserved	Private Use
65535	Reserved	

Table 2: SFC Active OAM Message Type

IANA is requested to assign new type from the SFC Active OAM Message Type registry as follows:

Value	Description	Reference
TBA2	SFC Echo Request/Echo Reply	This document

Table 3: SFC Echo Request/Echo Reply Type

9.3. SFC Echo Request/Echo Reply Parameters

IANA is requested to create new SFC Echo Request/Echo Reply Parameters registry.

9.4. SFC Echo Request/Echo Reply Message Types

IANA is requested to create in the SFC Echo Request/Echo Reply Parameters registry the new sub-registry Message Types. All code points in the range 1 through 191 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126] and assign values as follows:

Value	Description	Reference
0	Reserved	
TBA3	SFC Echo Request	This document
TBA4	SFC Echo Reply	This document
TBA4+1-191	Unassigned	IETF Review
192-251	Unassigned	First Come First Served
252-254	Unassigned	Private Use
255	Reserved	

Table 4: SFC Echo Request/Echo Reply Message Types

9.5. SFC Echo Reply Modes

IANA is requested to create in the SFC Echo Request/Echo Reply Parameters registry the new sub-registry Reply Modes All code points in the range 1 through 191 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126] and assign values as follows:

Value	Description	Reference
0	Reserved	
TBA5	Do Not Reply	This document
TBA6	Reply via an IPv4/IPv6 UDP Packet	This document
TBA7	Reply via Application Level Control Channel	This document
TBA8	Reply via Specified Path	This document
TBA8+1-191	Unassigned	IETF Review
192-251	Unassigned	First Come First Served
252-254	Unassigned	Private Use
255	Reserved	

Table 5: SFC Echo Reply Modes

9.6. SFC TLV Type

IANA is requested to create SFC OAM TLV Type registry. All code points in the range 1 through 32759 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 32760 through 65279 in this registry shall be allocated according to the "First Come First Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to the Table 6:

Value	Description	Reference
0	Reserved	This document
1- 32759	Unassigned	IETF Review
32760 - 65279	Unassigned	First Come First Served
65280 - 65519	Experimental	This document
65520 - 65534	Private Use	This document
65535	Reserved	This document

Table 6: SFC TLV Type Registry

This document defines the following new value in SFC OAM TLV Type registry:

Value	Description	Reference
TBA9	Source IP Address	This document

Table 7: SFC OAM Source IP Address Type

9.7. SFC OAM UDP Port

IANA is requested to allocate UDP port number according to

Service Name	Port Number	Transport Protocol	Description	Semantics Definition	Reference
SFC OAM	TBA10	UDP	SFC OAM	Section 7.3	This document

Table 8: SFC OAM Port

10. References

10.1. Normative References

[I-D.ietf-sfc-nsh]
Quinn, P., Elzur, U., and C. Pignataro, "Network Service Header (NSH)", draft-ietf-sfc-nsh-21 (work in progress), September 2017.

[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>.

[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>.

10.2. Informative References

[I-D.ietf-sfc-oam-framework]
Aldrin, S., Pignataro, C., Kumar, N., Akiya, N., Krishnan, R., and A. Ghanwani, "Service Function Chaining (SFC) Operation, Administration and Maintenance (OAM) Framework", draft-ietf-sfc-oam-framework-03 (work in progress), September 2017.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

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

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Wei Meng
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: meng.wei2@zte.com.cn,vally.meng@gmail.com

Bhumip Khasnabish
ZTE TX, Inc.
55 Madison Avenue, Suite 160
Morristown, New Jersey 07960
USA

Email: bhumip.khasnabish@ztetx.com

Cui Wang

Email: lindawangjoy@gmail.com

SFC WG
Internet-Draft
Updates: 8300 (if approved)
Intended status: Standards Track
Expires: April 10, 2019

G. Mirsky
ZTE Corp.
W. Meng
ZTE Corporation
B. Khasnabish
ZTE TX, Inc.
C. Wang
October 7, 2018

Active OAM for Service Function Chains in Networks
draft-wang-sfc-multi-layer-oam-12

Abstract

A set of requirements for active Operation, Administration and Maintenance (OAM) of Service Function Chains (SFCs) in networks is presented. Based on these requirements an encapsulation of active OAM message in SFC and a mechanism to detect and localize defects described. Also, this document updates RFC 8300 in the definition of O (OAM) bit in the Network Service Header (NSH) and defines how the active OAM message identified in SFC NSH.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 10, 2019.

Copyright Notice

Table of Contents

1. Introduction

[RFC7665] defines components necessary to implement Service Function Chain (SFC). These include a classifier which performs the classification of incoming packets. A Service Function Forwarder (SFF) is responsible for forwarding traffic to one or more connected Service Functions (SFs) according to the information carried in the SFC encapsulation. SFF also handles traffic coming back from the SF and transports the data packets to the next SFF. And the SFF serves as termination element of the Service Function Path (SFP). SF is responsible for the specific treatment of received packets.

Resulting from that SFC is constructed by a number of these components, there are different views from different levels of the SFC. One is the SFC, entirely abstract entity, which defines an ordered set of SFs that must be applied to packets selected as a result of classification. But SFC doesn't specify the exact mapping between SFFs and SFs. Thus there exists another semi-abstract entity referred to as SFP. SFP is the instantiation of the SFC in the network and provides a level of indirection between the entirely abstract SFC and a fully specified ordered list of SFFs and SFs identities that the packet will visit when it traverses the SFC. The latter entity is being referred to as Rendered Service Path (RSP). The main difference between SFP and RSP is that in the former the authority to select the SFF/SF has been delegated to the network.

This document defines how active Operation, Administration and Maintenance (OAM), per [RFC7799] definition of active OAM, identified in Network Service Header (NSH) SFC, lists requirements to improve the troubleshooting efficiency, and defines SFC Echo request and Echo reply that enables on-demand Continuity Check, Connectivity Verification among other operations over SFC in networks. Also, this document updates Section 2.2 of [RFC8300] in part of the definition of O bit in the (NSH).

2. Conventions

2.1. Requirements Language

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

2.2. Terminology

Unless explicitly specified in this document, active OAM in SFC and SFC OAM are being used interchangeably.

e2e: End-to-End

FM: Fault Management

NSH: Network Service Header

OAM: Operations, Administration, and Maintenance

PRNG: Pseudorandom number generator

RDI: Remote Defect Indication

RSP: Rendered Service Path

SF: Service Function

SFC: Service Function Chain

SFF: Service Function Forwarder

SFP: Service Function Path

3. Requirements for Active OAM in SFC Network

To perform the OAM task of fault management (FM) in an SFC, that includes failure detection, defect characterization and localization, this document defines the set of requirements for active OAM mechanisms to be used on an SFC.

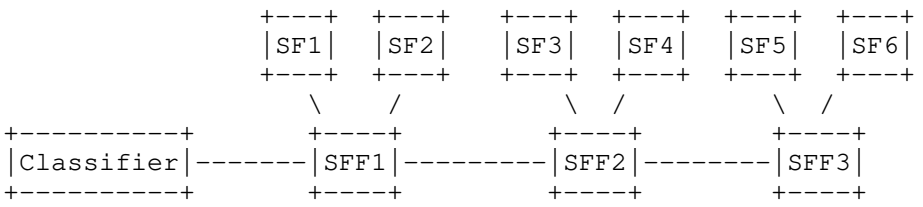


Figure 1: SFC reference model

In the example presented in Figure 1, the service SFP1 may be realized through two RSPs, RSP1(SF1--SF3--SF5) and RSP2(SF2--SF4--SF5). To perform end-to-end (e2e) FM SFC OAM:

REQ#1: Packets of active OAM in SFC SHOULD be fate sharing with data traffic, i.e., in-band with the monitored traffic follow the same RSP, in the forward direction from ingress toward egress endpoint(s) of the OAM test.

REQ#2: SFC OAM MUST support pro-active monitoring of any element in the SFC availability.

The egress, SFF3 in the example in Figure 1, is the entity that detects the failure of the SFC. It must be able to signal the new defect state to the ingress SFF1. Hence the following requirement:

REQ#3: SFC OAM MUST support Remote Defect Indication (RDI) notification by the egress to the ingress.

REQ#4: SFC OAM MUST support connectivity verification. Definition of the misconnection defect, entry and exit criteria are outside the scope of this document.

Once the SFF1 detects the defect objective of OAM switches from failure detection to defect characterization and localization.

REQ#5: SFC OAM MUST support fault localization of Loss of Continuity check in the SFC.

REQ#6: SFC OAM MUST support tracing an SFP to realize the RSP.

It is practical, as presented in Figure 1, that several SFs share the same SFF. In such case, SFP1 may be realized over two RSPs, RSP1(SF1--SF3--SF5) and RSP2(SF2--SF4--SF6).

REQ#7: SFC OAM MUST have the ability to discover and exercise all available RSPs in the transport network.

In the process of localizing the SFC failure, separating SFC OAM layers is an efficient approach. To achieve that continuity among SFFs that are part of the same SFP should be verified. Once SFFs reachability along the particular SFP has been confirmed task of defect localization may focus on SF reachability verification. Because reachability of SFFs has already verified, SFF local to the SF may be used as a source of the test packets.

REQ#8: SFC OAM MUST be able to trigger on-demand FM with responses being directed towards initiator of such proxy request.

4. Active OAM Identification in SFC NSH

The interpretation of O bit flag in the NSH header is defined in [RFC8300] as:

O bit: Setting this bit indicates an OAM packet.

This document updates the definition of O bit as follows:

O bit: Setting this bit indicates an OAM command and/or data in the NSH Context Header or packet payload

Active SFC OAM defined as a combination of OAM commands and/or data included in a message that immediately follows the NSH. To identify the active OAM message the value on the Next Protocol field MUST be

set to Active SFC OAM (TBA1) according to Section 8.1. The rules of interpreting the values of O bit and the Next Protocol field are as follows:

- O bit set and the Next Protocol value is not one of identifying active or hybrid OAM protocol (per [RFC7799] definitions), e.g., defined in this specification Active SFC OAM - TLVs contain OAM command or data, and the type of payload determined by the Next Protocol field;

- O bit set and the Next Protocol value is one of identifying active or hybrid OAM protocol - the payload that immediately follows SFC NSH contains OAM command or data;

- O bit is clear - no OAM in TLV and the payload determined by the value of the Next Protocol field.

Several active OAM protocols will be needed to address all the requirements listed in Section 3. Destination UDP port number may identify protocols if IP/UDP encapsulation used. But extra IP/UDP headers, especially in the case of IPv6, add noticeable overhead. This document defines Active OAM Header Figure 2 to demultiplex active OAM protocols on an SFC.

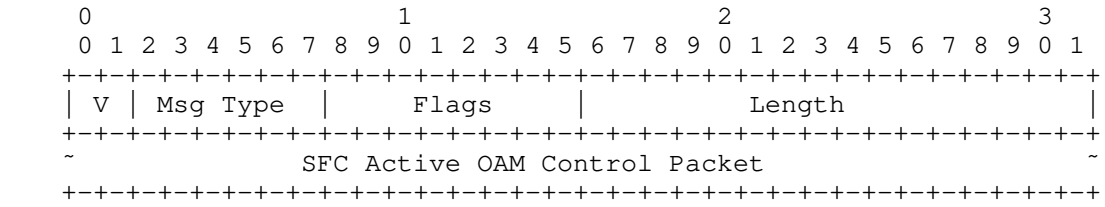


Figure 2: SFC Active OAM Header

V - two bits long field indicates the current version of the SFC active OAM header. The current value is 0.

Msg Type - six bits long field identifies OAM protocol, e.g., Echo Request/Reply or BFD.

Flags - eight bits long field carries bit flags that define optional capability and thus processing of the SFC active OAM control packet, e.g., optional timestamping.

Length - two octets long field that is the length of the SFC active OAM control packet in octets.

5. Echo Request/Echo Reply for SFC in Networks

Echo Request/Reply is a well-known active OAM mechanism that is extensively used to detect inconsistencies between a state in control and the data planes, localize defects in the data plane. The format of the Echo request/Echo reply control packet is to support ping and traceroute functionality in SFC in networks Figure 3 resembles the format of MPLS LSP Ping [RFC8029] with some exceptions.

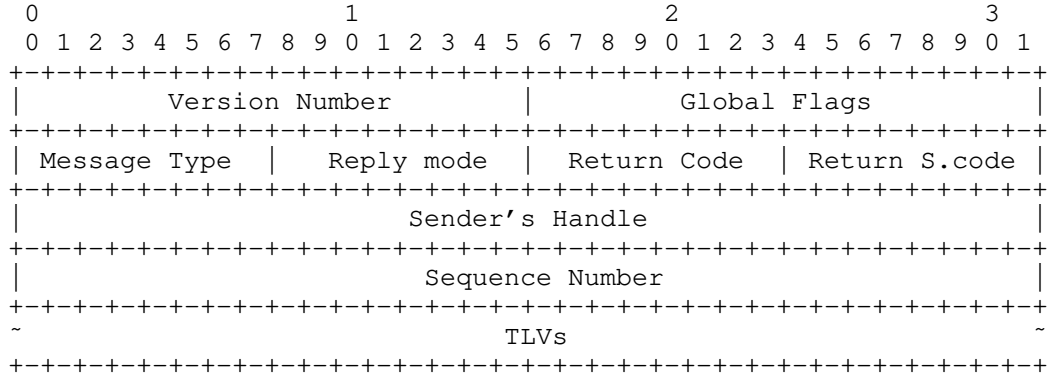


Figure 3: SFC Echo Request/Reply format

The interpretation of the fields is as follows:

The Version reflects the current version. The version number is to be incremented whenever a change is made that affects the ability of an implementation to parse or process control packet correctly.

The Global Flags is a bit vector field.

The Message Type filed reflects the type of the packet. Value TBA3 identifies echo request and TBA4 - echo reply

The Reply Mode defines the type of the return path requested by the sender of the echo request.

Return Codes and Subcodes can be used to inform the sender about the result of processing its request.

The Sender's Handle is filled in by the sender and returned unchanged by the receiver in the echo reply. The sender MAY use a pseudo-random number generator (PRNG) to set the value of the Sender's Handle field. The value of the Sender's Handle field SHOULD NOT be changed in the course of the test session.

The Sequence Number is assigned by the sender and can be (for example) used to detect missed replies. The value of the Sequence Number field SHOULD be monotonically increasing in the course of the test session.

TLVs (Type-Length-Value tuples) have the two octets long Type field, two octets long Length field that is the length of the Value field in octets.

5.1. SFC Echo Request Transmission

SFC echo request control packet MUST use the appropriate encapsulation of the monitored SFP. If Network Service Header (NSH) is used, echo request MUST set O bit, as defined in [RFC8300]. SFC NSH MUST be immediately followed by the SFC Active OAM Header defined in Section 4. Message Type field in the SFC Active OAM Header MUST be set to SFC Echo Request/Echo Reply value (TBA2) per Section 8.2.

Value of the Reply Mode field MAY be set to:

- Do Not Reply (TBA5) if one-way monitoring is desired. If the echo request is used to measure synthetic packet loss; the receiver may report loss measurement results to a remote node.

- Reply via an IPv4/IPv6 UDP Packet (TBA6) value likely will be the most used.

- Reply via Application Level Control Channel (TBA7) value if the SFP may have bi-directional paths.

- Reply via Specified Path (TBA7) value to enforce the use of the particular return path specified in the included TLV to verify bi-directional continuity and also increase the robustness of the monitoring by selecting a more stable path.

5.2. SFC Echo Request Reception

5.3. SFC Echo Reply Transmission

The Reply Mode field directs whether and how the echo reply message should be sent. The sender of the echo request MAY use TLVs to request that the corresponding echo reply is transmitted over the specified path. Value TBA3 is referred to as "Do not reply" mode and suppresses transmission of echo reply packet. The default value (TBA6) for the Reply mode field requests the responder to send the echo reply packet out-of-band as IPv4 or IPv6 UDP packet.

Responder to the SFC echo request sends the echo reply over IP network if the Reply mode is Reply via an IPv4/IPv6 UDP Packet. Because SFC NSH does not identify the ingress of the SFP the echo request, the source ID MUST be included in the message and used as the IP destination address for IP/UDP encapsulation of the SFC echo reply. The sender of the SFC echo request MUST include SFC Source TLV Figure 4.

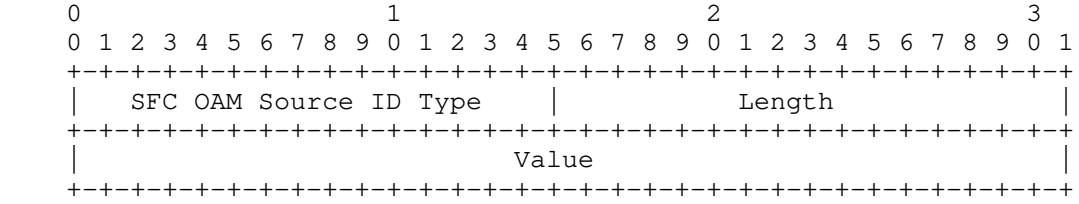


Figure 4: SFC Source TLV

where

SFC OAM Source Id Type is two octets in length and has the value of TBA9 Section 8.6.

Length is two octets long field, and the value equals the length of the Value field in octets.

Value field contains the IP address of the sender of the SFC OAM control message, IPv4 or IPv6.

The UDP destination port for SFC Echo Reply TBA10 will be allocated by IANA Section 8.7.

5.4. Overlay Echo Reply Reception

6. Security Considerations

Overlay Echo Request/Reply operates within the domain of the overlay network and thus inherits any security considerations that apply to the use of that overlay technology and, consequently, underlay data plane. Also, the security needs for SFC echo request/reply are similar to those of ICMP ping [RFC0792], [RFC4443] and MPLS LSP ping [RFC8029].

There are at least three approaches of attacking a node in the overlay network using the mechanisms defined in the document. One is a Denial-of-Service attack, by sending SFC ping to overload an element of the SFC. The second may use spoofing, hijacking,

replying, or otherwise tampering with SFC echo requests and/or replies to misrepresent, alter operator's view of the state of the SFC. The third is an unauthorized source using an SFC echo request/ reply to obtain information about the SFC and/or its elements, e.g. SFF or SF.

It is RECOMMENDED that implementations throttle the SFC ping traffic going to the control plane to mitigate potential Denial-of-Service attacks.

Reply and spoofing attacks involving faking or replying SFC echo reply messages would have to match the Sender's Handle and Sequence Number of an outstanding SFC echo request message which is highly unlikely. Thus the non-matching reply would be discarded.

To protect against unauthorized sources trying to obtain information about the overlay and/or underlay an implementation MAY check that the source of the echo request is indeed part of the SFP.

7. Acknowledgments

Authors greatly appreciate thorough review and the most helpful comments from Dan Wing.

8. IANA Considerations

8.1. SFC Active OAM Protocol

IANA is requested to assign a new type from the SFC Next Protocol registry as follows:

Value	Description	Reference
TBA1	SFC Active OAM	This document

Table 1: SFC Active OAM Protocol

8.2. SFC Active OAM Message Type

IANA is requested to create a new registry called "SFC Active OAM Message Type". All code points in the range 1 through 32767 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Remaining code points to be allocated according to the table Table 2:

Value	Description	Reference
0	Reserved	
1 - 32767	Reserved	IETF Consensus
32768 - 65530	Reserved	First Come First Served
65531 - 65534	Reserved	Private Use
65535	Reserved	

Table 2: SFC Active OAM Message Type

IANA is requested to assign new type from the SFC Active OAM Message Type registry as follows:

Value	Description	Reference
TBA2	SFC Echo Request/Echo Reply	This document

Table 3: SFC Echo Request/Echo Reply Type

8.3. SFC Echo Request/Echo Reply Parameters

IANA is requested to create new SFC Echo Request/Echo Reply Parameters registry.

8.4. SFC Echo Request/Echo Reply Message Types

IANA is requested to create in the SFC Echo Request/Echo Reply Parameters registry the new sub-registry Message Types. All code points in the range 1 through 191 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126] and assign values as follows:

Value	Description	Reference
0	Reserved	
TBA3	SFC Echo Request	This document
TBA4	SFC Echo Reply	This document
TBA4+1-191	Unassigned	IETF Review
192-251	Unassigned	First Come First Served
252-254	Unassigned	Private Use
255	Reserved	

Table 4: SFC Echo Request/Echo Reply Message Types

8.5. SFC Echo Reply Modes

IANA is requested to create in the SFC Echo Request/Echo Reply Parameters registry the new sub-registry Reply Modes All code points in the range 1 through 191 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126] and assign values as follows:

Value	Description	Reference
0	Reserved	
TBA5	Do Not Reply	This document
TBA6	Reply via an IPv4/IPv6 UDP Packet	This document
TBA7	Reply via Application Level Control Channel	This document
TBA8	Reply via Specified Path	This document
TBA8+1-191	Unassigned	IETF Review
192-251	Unassigned	First Come First Served
252-254	Unassigned	Private Use
255	Reserved	

Table 5: SFC Echo Reply Modes

8.6. SFC TLV Type

IANA is requested to create SFC OAM TLV Type registry. All code points in the range 1 through 32759 in this registry shall be allocated according to the "IETF Review" procedure as specified in [RFC8126]. Code points in the range 32760 through 65279 in this registry shall be allocated according to the "First Come First

Served" procedure as specified in [RFC8126]. Remaining code points are allocated according to the Table 6:

Value	Description	Reference
0	Reserved	This document
1- 32759	Unassigned	IETF Review
32760 - 65279	Unassigned	First Come First Served
65280 - 65519	Experimental	This document
65520 - 65534	Private Use	This document
65535	Reserved	This document

Table 6: SFC TLV Type Registry

This document defines the following new value in SFC OAM TLV Type registry:

Value	Description	Reference
TBA9	Source IP Address	This document

Table 7: SFC OAM Source IP Address Type

8.7. SFC OAM UDP Port

IANA is requested to allocate UDP port number according to

Service Name	Port Number	Transport Protocol	Description	Semantics Definition	Reference
SFC OAM	TBA10	UDP	SFC OAM	Section 5.3	This document

Table 8: SFC OAM Port

9. References

9.1. Normative References

[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>.

[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>.

[RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <https://www.rfc-editor.org/info/rfc8300>.

9.2. Informative References

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <https://www.rfc-editor.org/info/rfc7799>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

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

Authors' Addresses

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Wei Meng
ZTE Corporation
No.50 Software Avenue, Yuhuatai District
Nanjing
China

Email: meng.wei2@zte.com.cn,vally.meng@gmail.com

Bhumip Khasnabish
ZTE TX, Inc.
55 Madison Avenue, Suite 160
Morristown, New Jersey 07960
USA

Email: bhumip.khasnabish@ztetx.com

Cui Wang

Email: lindawangjoy@gmail.com

MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2017

X. Xu
S. Bryant
Huawei
H. Assarpour
Broadcom
H. Shah
Ciena
L. Contreras
Telefonica I+D
D. Bernier
Bell Canada
J. Tantsura
Individual
S. Ma
Juniper
M. Vigoureux
Nokia

June 29, 2017

Service Chaining using Unified Source Routing Instructions
draft-xu-mpls-service-chaining-03

Abstract

Source Packet Routing in Networking (SPRING) WG is developing an MPLS source routing mechanism. The MPLS source routing mechanism can be leveraged to realize a unified source routing instruction which works across both IPv4 and IPv6 underlays in addition to the MPLS underlay. This document describes how to leverage the unified source routing instruction to realize a transport-independent service function chaining by encoding the service function path information or service function chain information as an MPLS label stack.

Requirements Language

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 31, 2017.

Table of Contents

1. Introduction

When applying a particular Service Function Chain (SFC) [RFC7665] to the traffic selected by a service classifier, the traffic need to be steered through an ordered set of Service Functions (SF) in the network. This ordered set of SFs in the network indicates the Service Function Path (SFP) associated with the above SFC. In order

to steer the selected traffic through the required ordered list of SFs, the service classifier needs to attach information to the packet specifying exactly which Service Function Forwarders (SFFs) and which SFs are to be visited by traffic), the SFC, or the partially specified SFP which is in between the former two extremes.

The Source Packet Routing in Networking (SPRING) WG is developing an MPLS source routing mechanism which can be used to steer traffic through an ordered set of routers (i.e., an explicit path) and instruct nodes on that path to execute specific operations on the packet. By leveraging the MPLS source routing mechanism, [I-D.xu-mpls-unified-source-routing-instruction] describes a unified source routing instruction which works across both IPv4 and IPv6 underlays in addition to the MPLS underlay. This document describes how to leverage the unified source routing instruction to realize a transport-independent service function chaining by encoding the service function path information or service function chain information as an MPLS label stack.

2. Terminology

This memo makes use of the terms defined in [I-D.ietf-spring-segment-routing-mpls], [I-D.xu-mpls-unified-source-routing-instruction] and [RFC7665].

3. Solution Description

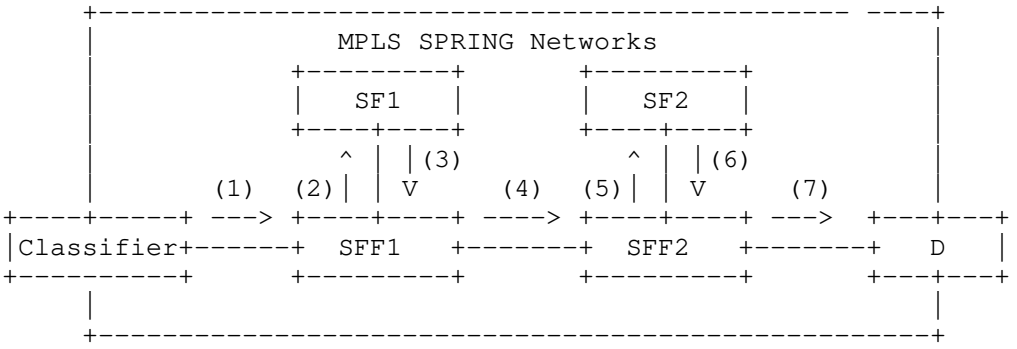


Figure 1: Service Function Chaining in MPLS-SPRING Networks

As shown in Figure 1, SFF1 and SFF2 are two MPLS-SPRING-capable nodes. They are also SFFs, each with one SF attached. In addition, they have allocated and advertised MPLS labels for their locally attached SFs. For example, SFF1 allocates and advertises a label (i.e., L(SF1)) for SF1 while SFF2 allocates and advertises a label (i.e., L(SF2)) for SF2. These labels, which are used to indicate SFs are referred to as SF labels. To encode the SFP information as an

MPLS label stack, local MPLS labels are allocated from SFFs' (e.g., SFF1 in Figure 1) label spaces to identify their locally attached SFs (e.g., SF1 in Figure 1), whilst the SFFs are identified by either nodal SIDs or adjacency SIDs depending on how strictly the network path needs to be specified. In addition, assume node SIDs for SFF1 and SFF2 are L(SFF1) and L(SFF2) respectively. In contrast, to encode the SFC information by an MPLS label stack, those SF labels MUST be domain-wide unique MPLS labels.

Now assume a given traffic flow destined for destination D is selected by the service classifier to go through a particular SFC (i.e., SF1-> SF2) before reaching its final destination D. Section 3.1 and 3.2 describe approaches of leveraging the MPLS- based source routing mechanisms to realize the service function chaining by encoding the SFP information within an MPLS label stack and by encoding the SFC information within an MPLS label stack respectively. Since the encoding of the partially specified SFP is just a simple combination of the encoding of the SFP and the encoding of the SFC, this document would not describe how to encode the partially specified SFP anymore.

3.1. Encoding SFP Information by an MPLS Label Stack

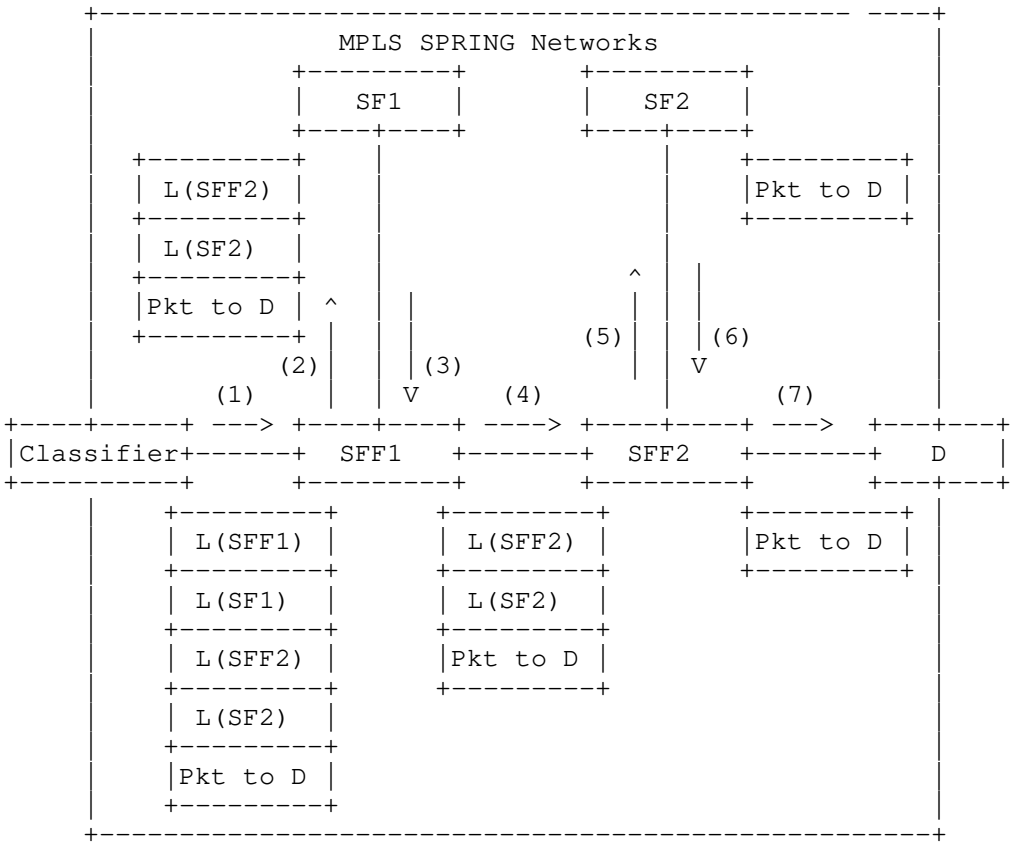


Figure 2: Packet Walk in MPLS underlay

As shown in Figure 2, since the selected packet needs to travel through an SFC (i.e., SF1->SF2), the service classifier would attach a segment list of (i.e., SID(SFF1)->SID(SF1)->SID(SFF2)-> SID(SF2)) which indicates the corresponding SFP to the packet. This segment list is represented by an MPLS label stack. To some extent, the MPLS label stack here could be looked as a specific implementation of the SFC encapsulation used for containing the SFP information [RFC7665]. When the encapsulated packet arrives at SFF1, SFF1 would know which SF should be performed according to the top label (i.e., SID (SF1)) of the received MPLS packet. We first consider the case where SF1 is an encapsulation aware SF, i.e., it understands how to process a packet with a pre-pended MPLS label stack. In this case the packet would be sent to SF1 by SFF1 with the label stack SID(SFF2)-> SID(SF2). SF1 would perform the required service function on the received MPLS packet where the payload is constrained to be an IP packet, and the SF needs to process both IPv4 and IPv6 packets (note that the SF would use the first nibble of the MPLS payload to

identify the payload type). After the MPLS packet is returned from SF1, SFF1 would send it to SFF2 according to the top label (i.e., SID (SFF2)).

If SF1 is a legacy SF, i.e. one that is unable to process the MPLS label stack, the remaining MPLS label stack (i.e., SID(SFF2)->SID(SF2)) MUST be saved and stripped from the packet before sending the packet to SF1. When the packet is returned from SF1, SFF1 would re-impose the MPLS label stack which had been previously stripped and then send the packet to SFF2 according to the current top label (i.e., SID (SFF2)). As for how to associate the corresponding MPLS label stack with the packets returned from legacy SFs, those mechanisms as described in [I-D.song-sfc-legacy-sf-mapping] could be considered.

When the encapsulated packet arrives at SFF2, SFF2 would perform the similar action to that described above.

As shown in Figure 3, if there is no MPLS LSP towards the next node segment (i.e., the next SFF identified by the current top label), the corresponding IP-based tunnel for MPLS (e.g., MPLS-in-IP/GRE tunnel [RFC4023], MPLS-in-UDP tunnel [RFC7510] or MPLS-in-L2TPv3 tunnel [RFC4817]) would be used instead, according to the unified source routing instruction as described in [I-D.xu-mpls-unified-source-routing-instruction].

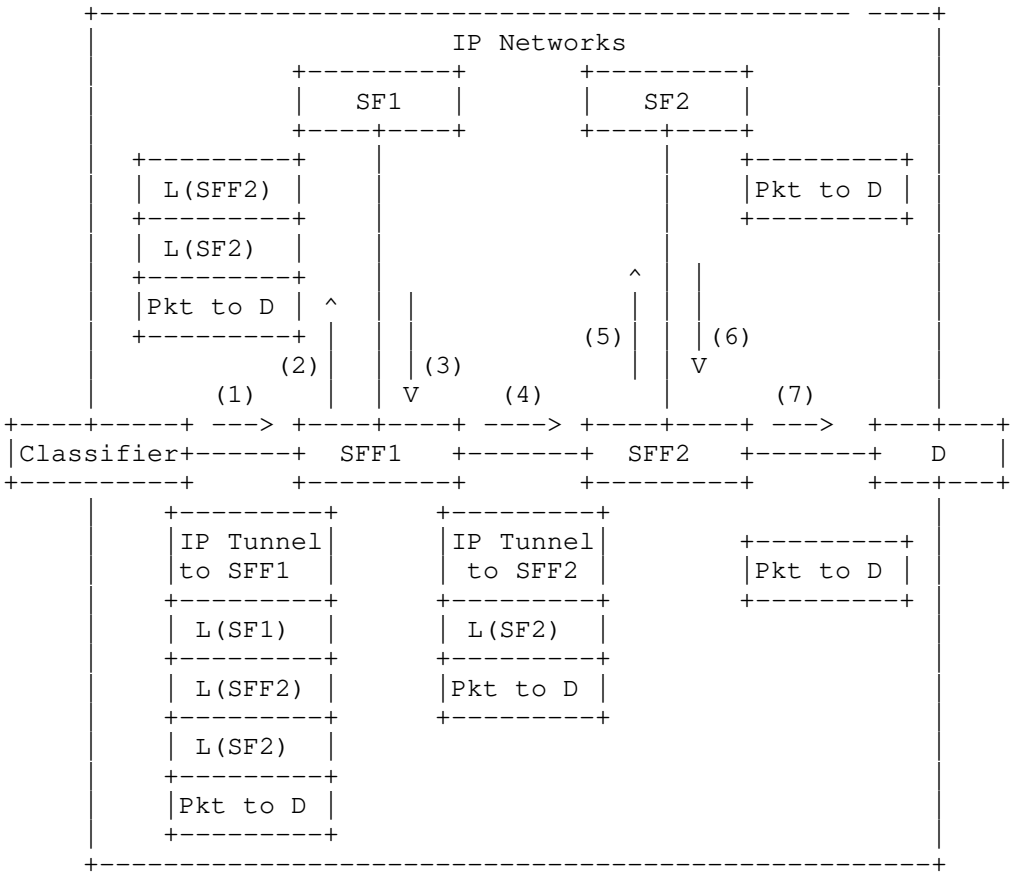


Figure 3: Packet Walk in IP underlay

Since the transport (i.e., the underlay) could be IPv4, IPv6 or even MPLS networks, the above approach of encoding the SFP information by an MPLS label stack is fully transport-independent which is one of the major requirements for the SFC encapsulation [RFC7665].

3.2. Encoding SFC Information by an MPLS Label Stack

Since the selected packet needs to travel through an SFC (i.e., SF1->SF2), the service classifier would attach an MPLS label stack (i.e., L(SF1)->L(SF2)) which indicates that SFC to the packet. Since it's known to the service classifier that SFF1 is attached with an instance of SF1, the service classifier would therefore send the MPLS encapsulated packet through either an MPLS LSP tunnel or an IP-based tunnel towards SFF1 (as shown in Figure 4 and 5 respectively). When the MPLS encapsulated packet arrives at SFF1, SFF1 would know which SF should be performed according to the current top label (i.e.,

L(SF1)). Similarly, SFF1 would send the packet returned from SF1 to SFF2 through either an MPLS LSP tunnel or an IP-based tunnel towards SFF2 since it's known to SFF1 that SFF2 is attached with an instance of SF2. When the encapsulated packet arrives at SFF2, SFF2 would do the similar action as what has been done by SFF1. Since the transport (i.e., the underlay) could be IPv4, IPv6 or even MPLS networks, the above approach of encoding the SFC information by an MPLS label stack is fully transport-independent which is one of the major requirements for the SFC encapsulation [RFC7665].

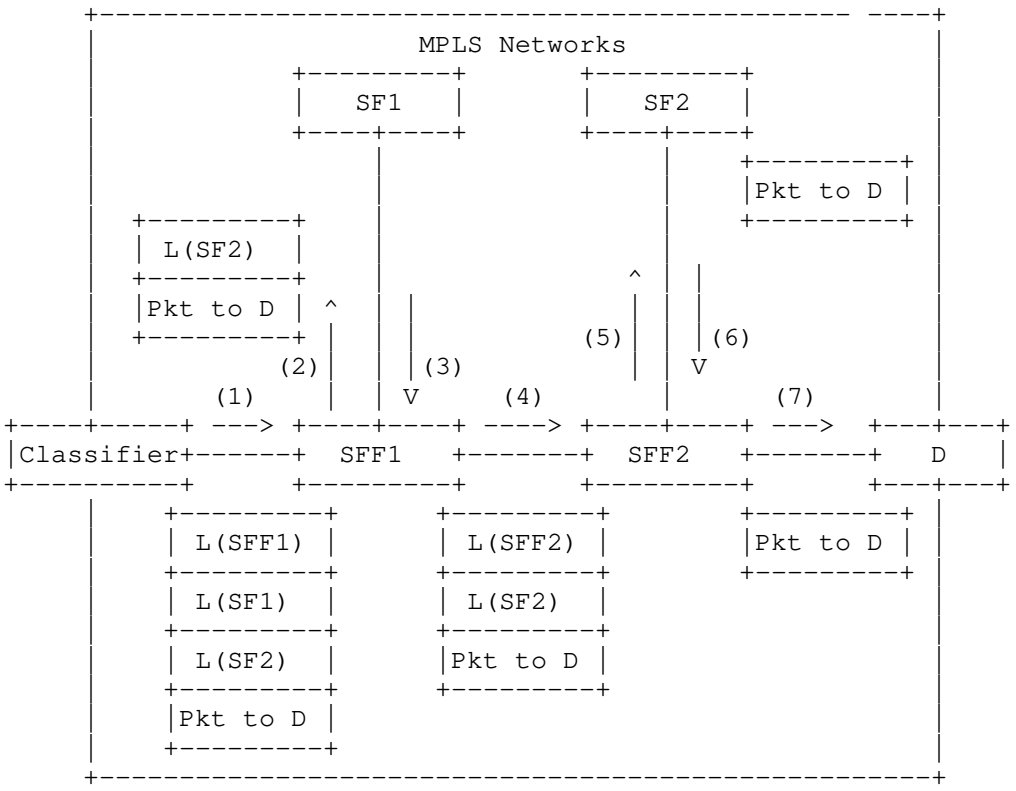


Figure 4: Packet Walk in MPLS underlay

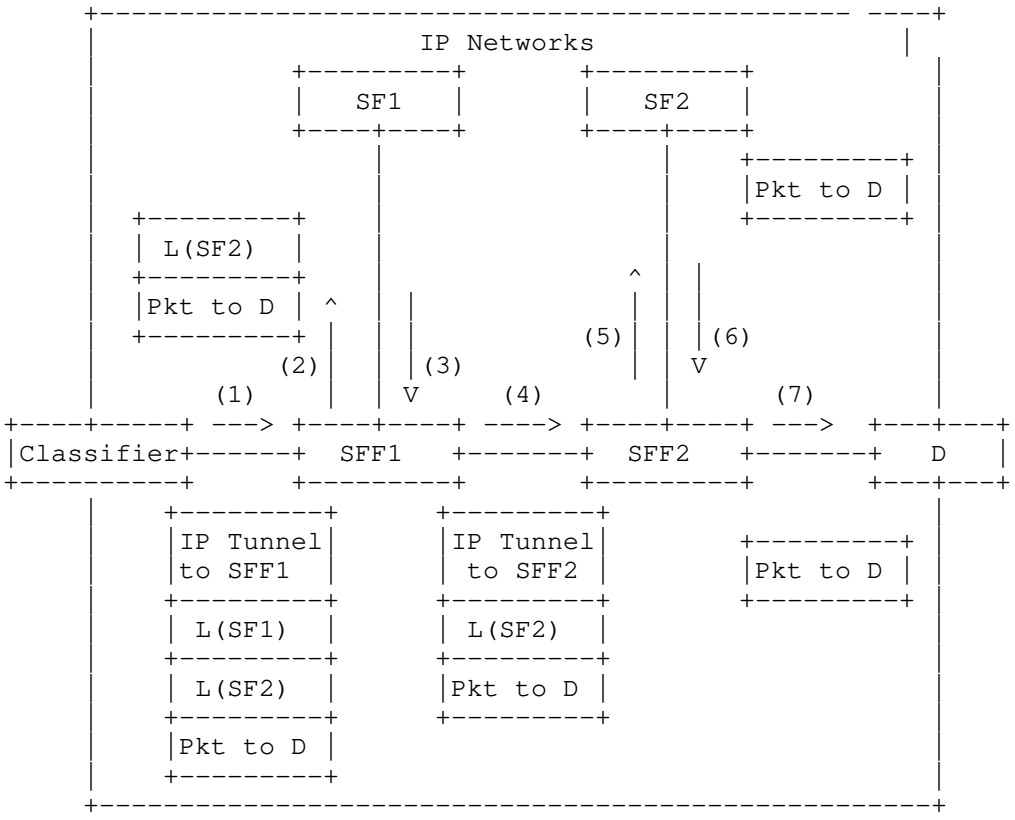


Figure 5: Packet Walk in IP underlay

3.3. How to Contain Metadata within an MPLS Packet

Since the MPLS encapsulation has no explicit protocol identifier field to indicate the protocol type of the MPLS payload, how to indicate the presence of metadata (i.e., the NSH which is only used as a metadata containner) in an MPLS packet is a potential issue to be addressed. One possible way to address the above issue is: SFFs allocate two different labels for a given SF, one indicates the presence of NSH while the other indicates the absence of NSH. This approach has no change to the current MPLS architecture but it would require more than one label binding for a given SF. Another possible way is to introduce a protocol identifier field within the MPLS packet as described in [I-D.xu-mpls-payload-protocol-identifier].

More details about how to contain metadata within an MPLS packet would be considered in the future version of this draft.

4. Acknowledgements

The authors would like to thank Loa Andersson, Andrew G. Malis, Adrian Farrel, Alexander Vainshtein and Joel M. Halpern for their valuable comments and suggestions on the document.

5. IANA Considerations

This document makes no request of IANA.

6. Security Considerations

It is fundamental to the SFC design that the classifier is a trusted resource which determines the processing that the packet will be subject to, including for example the firewall. It is also fundamental to the SPRING design that packets are routed through the network using the path specified by the node imposing the SIDs. Where an SF is not encapsulation aware the packet may exist as an IP packet, however this is an intrinsic part of the SFC design which needs to define how a packet is protected in that environment. Where a tunnel is used to link two non-MPLS domains, the tunnel design needs to specify how it is secured. Thus the secutity vulnerabilities are addressed in the underlying technologies used by this design, which itself does not introduce any new security vulnerabilities.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

7.2. Informative References

[I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-12 (work in progress), February 2017.

[I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing with MPLS data plane", draft-ietf-spring-segment-routing-mpls-10 (work in progress), June 2017.

[I-D.song-sfc-legacy-sf-mapping]
Song, H., You, J., Yong, L., Jiang, Y., Dunbar, L., Bouthors, N., and D. Dolson, "SFC Header Mapping for Legacy SF", draft-song-sfc-legacy-sf-mapping-08 (work in progress), September 2016.

[I-D.xu-mpls-payload-protocol-identifier]
Xu, X., "MPLS Payload Protocol Identifier", draft-xu-mpls-payload-protocol-identifier-02 (work in progress), December 2016.

[I-D.xu-mpls-unified-source-routing-instruction]
Xu, X., Bryant, S., Raszuk, R., Chunduri, U., Contreras, L., Jalil, L., Assarpour, H., Velde, G., Tantsura, J., and S. Ma, "Unified Source Routing Instruction using MPLS Label Stack", draft-xu-mpls-unified-source-routing-instruction-02 (work in progress), June 2017.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <http://www.rfc-editor.org/info/rfc4023>.

[RFC4817] Townsley, M., Pignataro, C., Wainner, S., Seely, T., and J. Young, "Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3", RFC 4817, DOI 10.17487/RFC4817, March 2007, <http://www.rfc-editor.org/info/rfc4817>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <http://www.rfc-editor.org/info/rfc7510>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <http://www.rfc-editor.org/info/rfc7665>.

Authors' Addresses

Xiaohu Xu
Huawei

Email: xuxiaohu@huawei.com

Stewart Bryant
Huawei

Email: stewart.bryant@gmail.com

Hamid Assarpour
Broadcom

Email: hamid.assarpour@broadcom.com

Himanshu Shah
Ciena

Email: hshah@ciena.com

Luis M. Contreras
Telefonica I+D
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid, 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://people.tid.es/LuisM.Contreras/

Daniel Bernier
Bell Canada

Email: daniel.bernier@bell.ca

Jeff Tantsura
Individual

Email: jefftant@gmail.com

Shaowen Ma
Juniper

Email: mashaowen@gmail.com

Martin Vigoureux
Nokia

Email: martin.vigoureux@nokia.com