Workgroup: SPRING
Internet-Draft: draft-huang-spring-sr-hsfc-00
Published: 13 March 2023
Intended Status: Standards Track
Expires: 14 September 2023
Authors: H. Huang X. Chen Z. Li
Huawei Huawei Huawei

Hierarchical Service Function Chaining with Segment Routing

Abstract

SFC offers ordered list of service functions applied to packets, include traditional network service functions and application-specific functions. hSFC defines a hierarchical architecture to deploy SFC in large networks, typically spanning multiple data centers or domains.

This document complements RFC 8459 to enable SR-based hSFC. This document specifies and categorizes the interactions between upper-level domains and lower-level sub-domains and appends SR-specific support in constructing hSFC.

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 14 September 2023.

respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Service Function Chaining (SFC) defines an ordered set of service functions and subsequent "steering" of traffic through them. The architecture of SFC is defined in [RFC7665].

To ease the problem of implementing SFC across a large, geographically dispersed network, hSFC[RFC8459] is proposed to decomposing a large domain to multiple SFC-enabled sub-domains. The topmost level domain encompasses the entire network domain to be managed, and each sub-domain may support a subset of the Service Functions (SFs). Such hierarchical approach can reduce SFC's management complexity and improve the scalability.

A simple example to deploy hSFC is where data centers with scattered locations host different types of service functions on a range of hardware and services are provided by SFCs spanning multiple data centers with each as an independent SFC sub-domain [I-D.draft-ietf-sfc-dc-use-cases].

SFC can be implemented based on Network Service Header (NSH) [RFC8300], which requires a mapping between both Service Path

Identifier (SPI) and Service Index (SI) to next-hop forwarding. SFC can also be instantiated based on SR[I-D.draft-ietf-spring-sr-service-programming], where service SIDs can be combined in a SID-list explicitly indicated by the source SR node to represent an SFC.

[RFC8459] proposes the hSFC data plane solution based on the assumption of NSH. As a supplement, this document is also based on hierarchical SFC but utilizes SR-based service programming [I-D.draft-ietf-spring-sr-service-programming] instead.

This document follows the same concept of Internal Boundary Node (IBN) defined in [RFC8459] to act as a gateway between upper-level domain and lower-level sub-domain. This document mainly describes possible interactions of the upper and lower domains at the IBN using taxonomy and appends SR-specific support in constructing hSFC. This document assumes IBN is SR-aware and follows similar endpoint behavior in [I-D.draft-ietf-spring-sr-service-programming].

[I-D.draft-ietf-spring-nsh-sr] provides another SR-based method (i.e., SR-based SFC with integrated NSH service plane) to implement SFC in single domain. The methods specified in this document still apply to that scenario with some modifications and this may be considered in future work.

1.1. Terminology

The terms in this document are defined in [RFC7665], [RFC8459] and [I-D.draft-ietf-spring-sr-service-programming].

The following lists widely used terms in this document.

CF: Classifier

IBN: Internal Boundary Node

NSH: Network Service Header

SF: Service Function

SFC: Service Function Chaining

hSFC: Hierarchical Service Function Chaining

SR: Segment Routing

SC: Service Chain

TTL: Time To Live

NAT: Network Address Translator

1.2. Requirements Language

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

2. Internal Boundary Node (IBN)

This document follows the concept of Internal Boundary Node (IBN) defined in [RFC8459], which acts as a gateway to connect the upper and lower domains.

Figure 1 depicts an example architecture of hSFC. In SR-based hSFC, an IBN acts as an SR Proxy in the upper-level domain and as a classifier in the lower-level domain . The CF within IBN performs lower-level path selection (reclassification in sub-domain) and SR Proxy within IBN is responsible to handle upper-level SR paths, including caching and restoring them whenever the packets exit the sub-domain via IBN. IBN also realizes some cross-domain information transfer (e.g., SFC metadata).

In the example, packets are classified to traverse SC#1 in the upper-level classifier. When they arrives at the IBN, they're further re-classified by lower-level classifier to enter either SC#2 or SC#3. After the packets are processed by last SF (i.e., SF#2.2 or SF#3.2) of each lower-level SC, they're returned to IBN and then resume SC#1 traversal.

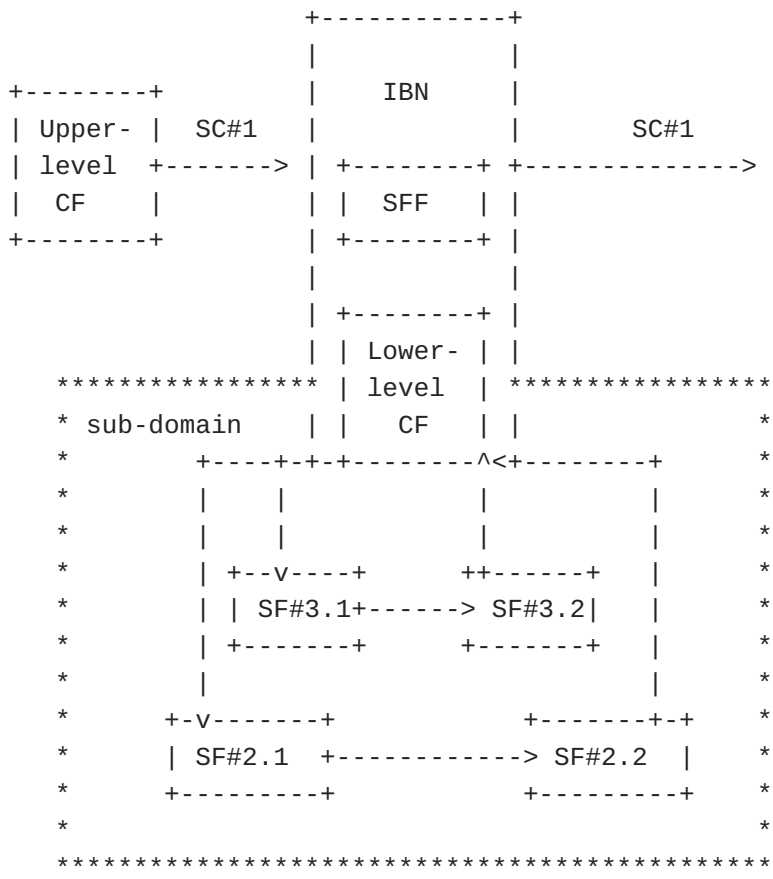

Figure 1: Illustration of hSFC Architecture

2.1. Path Selection/(Re-)Classification

The classification in upper-level domain (SR-based) follows the definition in [I-D.draft-ietf-spring-sr-service-programming]. The packets steered into an SR policy carry an ordered list of segments associated with that SR policy including corresponding IBNs[RFC9256].

The re-classification in lower-level domain varies according to the SFC deployment method applied in the sub-domain, [I-D.draft-ietf-spring-sr-service-programming] for SR-based SFC and [RFC8300] for NSH-based SFC.

2.2. Path Restoration

The information of upper-level SC is either not required or not available in the sub-domain. As per [RFC8459], it's recommended to store such information somewhere, within the packets themselves or in the IBN. At the termination of an SC in the sub-domain, the packets **SHOULD** be restored to the original upper-level SC to resume remaining SFC.

The following methods are categorized into stateful and stateless according to whether the state is stored in IBN.

2.2.1. Stateful IBN

The stateful method needs to store the state in the IBN, including the upper-level SC path (e.g., the whole IPv6 header, or partial fields such as SRH that are necessary to recover the path) and other auxiliary information (e.g., SFC metadata), and restore the state in order to recover the upper-level path when packets are returned to the IBN in the sub-domain.

The state saved in the IBN needs to be organized by selecting an appropriate index method so that the upper-level path can be accurately restored when exiting the sub-domain and consequently packets resume the remaining SFC processing.

The downside of stateful method is that it requires IBN to maintain scalability to handle the state explosion in large-scale networks with massive flows. The design of scalable IBNs is out-of-scope in this document.

This document further categorizes stateful methods with different indexing system.

2.2.1.1. Flow-indexed State

IBN can identify traffic in the granularity of flow such as five-tuple (source address, destination address, source port, destination port) , and use such information as an index to cache the state. Assume that the flow-specific information should not be modified in the lower-level domain, the path can be later restored by adopting the same identification.

However, this information may be modified in the lower-level domain (SFs such as NAT may modify the fields in the five-tuple). In this case, it is necessary to assign flow IDs with flow granularity in the IBN as described in Section 4.1.5 of [RFC8459] and encapsulate it in the packet (e.g., carried by metadata as shown in Figure 2), and use flow ID to index the state when exiting the sub-domain, with the assumption that this flow ID **MUST NOT** be modified along the path.

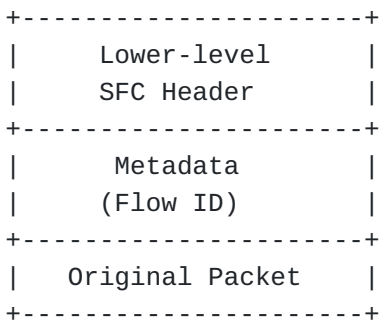

Figure 2: Encode Flow ID in Metadata

2.2.1.2. Path-indexed State

This section describes how to index state in IBN by path-specific identifier.

2.2.1.2.1. Upper-level Path ID

As per [RFC8459], SPI is used to distinguish different upper-level SC paths which can be used as an index method. As for SR-based SFC where service information is carried by data plane (e.g., SID list), the data plane information can be utilized to recognize different upper-level SC paths.

A passive method is to reuse the unique identifier of the upper-level path provided by native data plane mechanism such as path segment [I-D.draft-ietf-spring-srv6-path-segment] to index the states in IBN and encapsulate it in the metadata of packets throughout the sub-domain whereas the upper-level data plane information is thereby stripped and cached in IBN. Such path-specific identifiers can be used to index the state and perform path recovery at the termination of sub-domain in the IBN.

A proactive method is to compute and assign the path ID based of hash value of data plane information (e.g., source address and segment list) when the packets arrived at IBN. The packets would carry this path ID in metadata within the sub-domain. This identifier is also used to index the state and realize path restoration when the packets terminates sub-domain SC at the exit of IBN.

The assumption of the above two methods is that the sub-domain **MUST NOT** modify the path ID carried by metadata in the packets.

2.2.1.2.2. Lower-level Path ID

When the IBN reclassifies the packets, it can uniquely assign unique path ID to the selected lower-level path. This path ID is utilized to index the state and also be carried by the packet in order to

restore the upper-level state, following the aforementioned methods as upper-level path ID.

This method can only be effective assuming the traffic from different upper-level SCs **MUST NOT** be re-classified to the same lower-level SC. Otherwise, the same path ID will be ambiguously mapped into multiple upper-level path states.

Comparison: Typically, a class of flows may be classified into the same service function path. Therefore, flow-indexed method needs to assign more flow-specific identifiers (as index) and store more corresponding states (as value) to realize path restoration compared with path-indexed one.

2.2.1.3. SID-bound State

[I-D.draft-ietf-spring-sr-service-programming] proposes SR proxies to support SR-unaware SFs so that the SFs are agnostic about the SR network. This concept can also apply in hSFC since the lower-level SFC sub-domain is agnostic about the upper-level SR network. Therefore, SR-based hSFC can reuse the architecture of SR proxies (either static or dynamic) defined in [I-D.draft-ietf-spring-sr-service-programming] and thus, realize incremental deployment from single-layer SR SFC to hierarchical one. In such case, an IBN is composed of an SR Proxy in place of SFF in Figure 1 and an additional sub-domain classifier. The SR Proxy and the classifier could be deployed in either co-located or separate devices.

As shown in Figure 3, IBN in SR-based hSFC could be bound with SID in the granularity of upper-level path. As per [I-D.draft-ietf-spring-sr-service-programming], the SID is bound to a pair of directed interfaces (i.e., IFACE-IN and IFACE-OUT) on the proxy, where the matched traffic is sent via IFACE-OUT towards the associated lower-level CF and receiving the traffic returning from the CF via IFACE-IN. When the destination address of a packet matches the bound SID entry in the FIB of the SR-aware IBN, IBN decapsulates the upper-level SR Header of the packet and stores the state indexed by the receiving interface(IFACE-IN). After classified by the CF, lower-level SR header (if sub-domain supports SR-based SFC) will be encapsulated to the original packet in order for sub-domain SFC traversal.

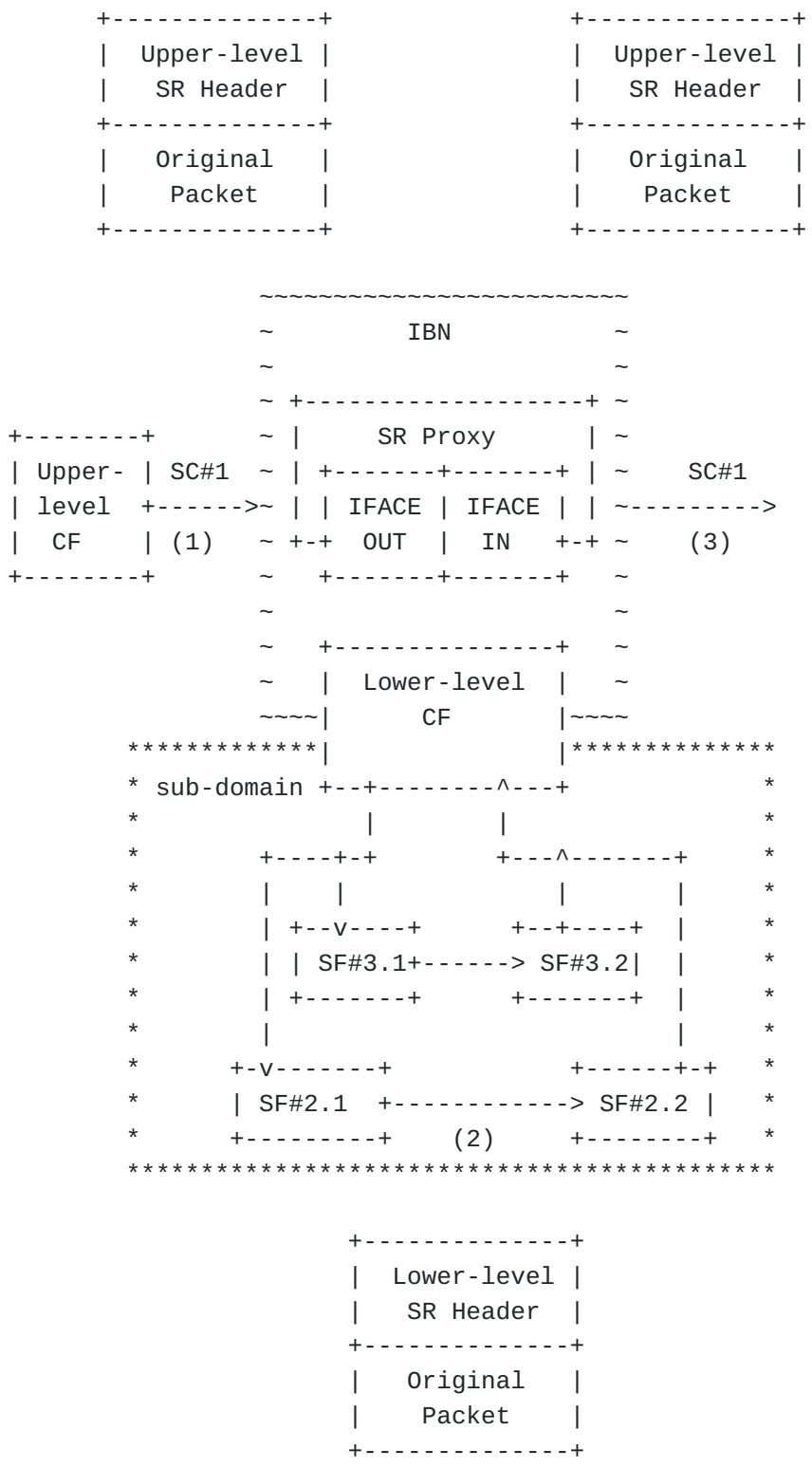

Figure 3: Example of IBN Evolving from SR Proxy

2.2.2. Stateless IBN

Compared with the stateful method, when entering the sub-domain, the stateless method does not require to store the state in the IBN locally. All upper-level path states are carried along with the packets themselves by either metadata or header encapsulation described below.

The disadvantage is that it will enlarge the original packet header, which may encounter MTU problems and reduce transmission efficiency in the sub-domains.

As the state is no longer stored in the IBN, when the sub-domain completes lower-level SFP, it is no longer required to return to the centralized IBN for the recovery of the upper-level path. Especially when some nodes (e.g., the last SF/SFF in lower-level path) are capable of restoring the upper-level SFP from the packet as IBN does, these nodes can extract the path information themselves without packets' going back to IBN. Furthermore, the packet can be directly forward to next-hop of upper-level path if the upper-level routing policy is visible to the sub-domain. This optimization would reduce the workload of the centralized IBN, promoting the scalability and reliability.

There are generally two ways to carry upper-level path state within the packet: metadata and header encapsulation.

2.2.2.1. Metadata

In Section 7 of [I-D.draft-ietf-spring-sr-service-programming] and Section 2 of [RFC8300], several methods of SFC metadata are defined, most of which are avaiable to take along upper-level path state as shown in Figure 4.

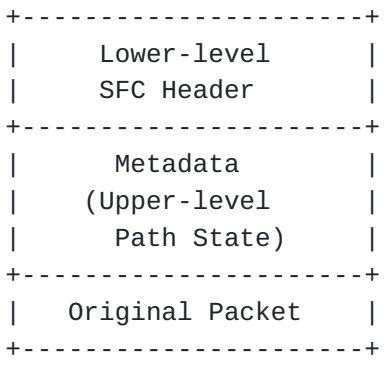

Figure 4: Encode Upper-level Path State in Metadata

In such case, SFs in the sub-domain are all required to support the recognition of the metadata it uses. Therefore, the inner payload can be processed by transparently skipping the metadata.

2.2.2.2. Header Encapsulation

The header encapsulation method is to nest the upper-level SR header between the header of the lower-level SFP and the inner payload as illustrated in Figure 5 and Figure 6. This requires all SFs in the sub-domain to be able to identify the upper-level IPv6 and extension header nested behind lower-level SFC header(e.g., SR or NSH). Therefore, the inner payload can be processed by transparently skipping the upper-SR header.

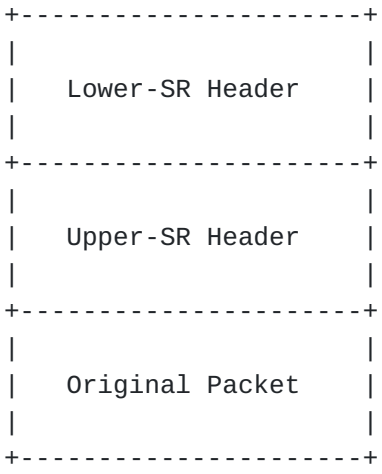

Figure 5: Encapsulation for SR-based Sub-domain

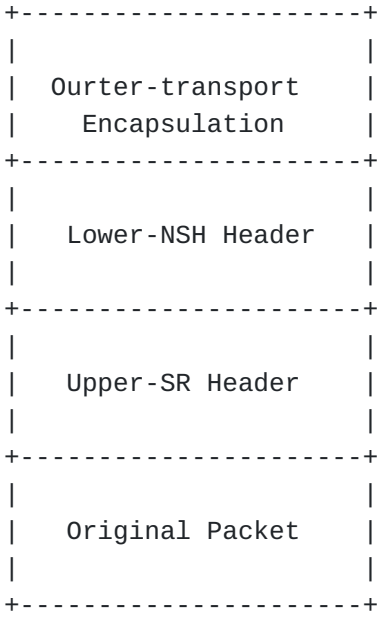

Figure 6: Encapsulation for NSH-based Sub-domain

Comparison: The encapsulation method requires to retain the whole upper-level header. In comparison, metadata can only carry partial information that is sufficient to reconstruct the upper-level path.

3. Control Plane

In addition to the hierarchical data plane, the control plane in hSFC is also hierarchical. In other words, the control planes across different domains can be invisible to each other. For example, a lower-level domain is agnostic about network service provided in neither upper-level domain nor other sub-domains, and vice versa.

In the case that the lower-level is willing to expose more information, SFCs can be orchestrated in more globally optimized manner, which, nevertheless, gradually walks away from the "hierarchical principle" and complicates the management and deployment.

[RFC9015] and [I-D.draft-li-spring-sr-sfc-control-plane-framework] provides some solutions for control plane of NSH and SR, respectively. But the control plane of hSFC still needs to be completed.

4. Experimental Considerations

4.1. OAM

SR-based SFC uses SRH Flags to mark O-bit identifying OAM packets while NSH-based SFC uses the unused bit of NSH Header as O-bit to identify OAM packets. For the end-to-end OAM method, IBN needs to hold consistent marker on the packets traversing different domains, that is, the O-bit needs to be copied to the corresponding position at IBN.

Deploying more OAM methods in hSFC, such as IOAM, iFIT [I-D.draft-song-opsawg-ifit-framework], etc., needs to be considered by future work.

5. Security Considerations

TBD.

6. IANA Considerations

TBD.

7. References

7.1. Normative References

[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/rfc/rfc8300>.

[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/rfc/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/rfc/rfc8174>.

7.2. Informative 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/rfc/rfc7665>.

[RFC8459] Dolson, D., Homma, S., Lopez, D., and M. Boucadair, "Hierarchical Service Function Chaining (hSFC)", RFC 8459, DOI 10.17487/RFC8459, September 2018, <https://www.rfc-editor.org/rfc/rfc8459>.

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

[I-D.draft-ietf-spring-sr-service-programming] Clad, F., Xu, X., Filsfils, C., Bernier, D., Li, C., Decraene, B., Ma, S., Yadlapalli, C., Henderickx, W., and S. Salsano, "Service Programming with Segment Routing", Work in Progress, Internet-Draft, draft-ietf-spring-sr-service-programming-07, 15 February 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-07>.

[I-D.draft-ietf-spring-nsh-sr] Guichard, J. and J. Tantsura, "Integration of Network Service Header (NSH) and Segment Routing for Service Function Chaining (SFC)", Work in Progress, Internet-Draft, draft-ietf-spring-nsh-sr-11, 20 April 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-nsh-sr-11>.

[I-D.draft-ietf-spring-srv6-path-segment] Li, C., Cheng, W., Chen, M., Dhody, D., and Y. Zhu, "Path Segment for SRv6 (Segment Routing in IPv6)", Work in Progress, Internet-Draft, draft-ietf-spring-srv6-path-segment-05, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-spring-srv6-path-segment-05>.

[I-D.draft-song-opsawg-ifit-framework] Song, H., Qin, F., Chen, H., Jin, J., and J. Shin, "A Framework for In-situ Flow Information Telemetry", Work in Progress, Internet-Draft, draft-song-opsawg-ifit-framework-19, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-song-opsawg-ifit-framework-19>.

[RFC9015] Farrel, A., Drake, J., Rosen, E., Uttaro, J., and L. Jalil, "BGP Control Plane for the Network Service Header in Service Function Chaining", RFC 9015, DOI 10.17487/RFC9015, June 2021, <https://www.rfc-editor.org/rfc/rfc9015>.

[I-D.draft-li-spring-sr-sfc-control-plane-framework] Li, C., El Sawaf, A., Huang, H., and Z. Li, "A Framework for Constructing Service Function Chaining Systems Based on Segment Routing", Work in Progress, Internet-Draft, draft-li-spring-sr-sfc-control-plane-framework-08, 11 January 2023, <https://datatracker.ietf.org/doc/html/draft-li-spring-sr-sfc-control-plane-framework-08>.

[I-D.draft-ietf-sfc-dc-use-cases] Kumar, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", Work in Progress, Internet-Draft, draft-ietf-sfc-dc-use-cases-06, 22 February 2017, <https://datatracker.ietf.org/doc/html/draft-ietf-sfc-dc-use-cases-06>.

Acknowledgements

Contributors

Authors' Addresses

Hongyi Huang
Huawei

Email: hongyi.huang@huawei.com

Xia Chen
Huawei

Email: jescia.chenxia@huawei.com

Zhenbin Li
Huawei

Email: lizhenbin@huawei.com