Stream: Internet Engineering Task Force (IETF)
RFC: 9516
Category: Standards Track
Published: November 2023
ISSN: 2070-1721

Authors:
G. Mirsky *Ericsson*
W. Meng *ZTE Corporation*
T. Ao *China Mobile*
B. Khasnabish *Individual contributor*
K. Leung *Individual contributor*
G. Mishra *Verizon Inc.*

RFC 9516
Active Operations, Administration, and Maintenance (OAM) for Service Function Chaining (SFC)

Abstract

A set of requirements for active Operations, Administration, and Maintenance (OAM) for Service Function Chaining (SFC) in a network is presented in this document. Based on these requirements, an encapsulation of active OAM messages in SFC and a mechanism to detect and localize defects are described.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

[RFC7665] defines data plane elements necessary to implement Service Function Chaining (SFC). These include the following:

1. Classifiers that perform the classification of incoming packets. Such classification may result in associating a received packet to a service function chain.
2. Service Function Forwarders (SFFs) that are responsible for forwarding traffic to one or more connected Service Functions (SFs) according to the information carried in the SFC encapsulation and handling traffic coming back from the SFs and forwarding it to the next SFF.
3. SFs that are responsible for executing specific service treatment on received packets.

There are different views from different levels of SFC. One is the service function chain, an entirely abstract view, which defines an ordered set of SFs that must be applied to packets selected based on classification rules. But the service function chain doesn't specify the exact mapping between SFFs and SFs. Thus, another logical construct used in SFC is a Service Function Path (SFP). According to [RFC7665], an SFP is the instantiation of SFC in the network and provides a level of indirection between the entirely abstract SFCs and a fully specified, ordered list of SFF and SF identities that the packet will visit when it traverses SFC. The latter entity is referred to as Rendered Service Path (RSP). The main difference between an SFP and RSP is that the former is the logical construct, while the latter is the realization of the SFP via the sequence of specific SFC data plane elements.

This document defines how active Operations, Administration, and Maintenance (OAM), per the definition of active OAM in [RFC7799], is implemented when the Network Service Header (NSH) [RFC8300] is used as the SFC encapsulation. Following the analysis of SFC OAM in [RFC8924], this document applies and, when necessary, extends requirements listed in Section 4 of [RFC8924] for the use of active OAM in an SFP supporting fault management and performance monitoring. Active OAM tools that are conformant to this specification improve OAM's ability for Fault Management (FM) by, for example, using the query mechanism to troubleshoot and localize defects, which conforms to the stateless character of transactions in SFC NSH [RFC8300]. Note that Performance Monitoring OAM, as required by [RFC8924], is not satisfied by this document and is out of scope. For the purpose of FM OAM in SFC, the SFC Echo Request and Echo Reply are specified in Section 6. These mechanisms enable on-demand continuity check and connectivity verification, among other operations, over SFC in networks and address functionalities discussed in Sections 4.1, 4.2, and 4.3 of [RFC8924]. The SFC Echo Request and Echo Reply can be used with encapsulations other than the NSH, for example, using MPLS encapsulation, as described in [RFC8595]. The applicability of the SFC Echo Request/Reply mechanism in SFC encapsulations other than the NSH is outside the scope of this document.

The intended scope of SFC active OAM is for use within a single provider's operational domain. The SFC active OAM deployment scope is deliberately constrained, as explained in [RFC7665] and [RFC8300], and limited to a single network administrative domain.

2. Terminology and Conventions

The terminology defined in [RFC7665] is used extensively throughout this document, and the reader is expected to be familiar with it.

In this document, SFC OAM refers to an active OAM [RFC7799] in an SFC architecture. Additionally, "Echo Request/Reply" and "SFC Echo Request/Reply" are used interchangeably.

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. Acronyms

E2E:	End-to-End

FM: Fault Management

MAC: Message Authentication Code

NSH: Network Service Header

OAM: Operations, Administration, and Maintenance

RSP: Rendered Service Path

SF: Service Function

SFC: Service Function Chaining

SFF: Service Function Forwarder

SFI: Service Function Instance

SFP: Service Function Path

3. Requirements for Active OAM in SFC

As discussed in [RFC8924], SFC-specific means are needed to perform the FM OAM task in an SFC architecture, including failure detection, defect characterization, and localization. This document defines the set of requirements for active FM OAM mechanisms to be used in an SFC architecture.

```
                +-----+ +-----+ +-----+ +-----+ +-----+ +-----+
                |SFI11| |SFI12| |SFI21| |SFI22| |SFI31| |SFI32|
                +-----+ +-----+ +-----+ +-----+ +-----+ +-----+
                    \    /          \   /           \    /
     +----------+   +----+         +----+          +----+
     |Classifier|---|SFF1|---------|SFF2|----------|SFF3|
     +----------+   +----+         +----+          +----+
```

Figure 1: An Example of SFC Data Plane Architecture

The architecture example depicted in Figure 1 considers a service function chain that includes three distinct service functions. In this example, the SFP traverses SFF1, SFF2, and SFF3. Each SFF is connected to two Service Function Instances (SFIs) of the same SF. End-to-End (E2E) SFC OAM has the Classifier as the ingress and SFF3 as its egress. The scope of Segment SFC OAM is between two elements that are part of the same SFP. The following are the requirements for an FM SFC OAM, whether with the E2E or segment scope:

REQ1: Packets of SFC active OAM **SHOULD** be fate sharing with the monitored SFC data in the forward direction from ingress toward egress endpoint(s) of the OAM test.

The fate sharing, in the SFC environment, is achieved when a test packet traverses the same path and receives the same treatment in the underlay network layer as an SFC-encapsulated packet.

REQ2: SFC OAM **MUST** support monitoring of the continuity of the SFP between any of its elements.

An SFC failure might be declared when several consecutive test packets are not received within a predetermined time. For example, in the E2E FM SFC OAM case, i.e., the egress, SFF3 (Figure 1) could be the entity that detects the SFP's failure by monitoring a flow of periodic test packets. The ingress may be capable of recovering from the failure, e.g., using redundant SFC elements. Thus, it is beneficial for the egress to signal the new defect state to the ingress, which in this example, is the Classifier, hence, the following requirement:

REQ3: SFC OAM **MUST** support Remote Defect Indication notification by the egress to the ingress.

REQ4: SFC OAM **MUST** support connectivity verification of the SFP. The definitions of the misconnection defect, entry, and exit criteria are outside the scope of this document.

Once an SFF detects the defect, the objective of the SFC OAM changes from the detection of a defect to defect characterization and localization.

REQ5: SFC OAM **MUST** support fault localization of the loss of continuity check within an SFP.

REQ6: SFC OAM **MUST** support an SFP tracing to discover the RSP.

In the example presented in Figure 1, two distinct instances of the same SF share the same SFF. In this example, the SFP can be realized over several RSPs that use different instances of the SF of the same type, for instance, RSP1(SFI11--SFI21--SFI31) and RSP2(SFI12--SFI22--SFI32). Available RSPs can be discovered using the trace function discussed in Section 4.3 of [RFC8924] or the procedure defined in Section 6.5.4.

REQ7: SFC OAM **MUST** have the ability to discover and exercise all available RSPs in the network.

The SFC OAM layer model described in [RFC8924] offers an approach for defect localization within a service function chain. As the first step, the SFP's continuity for SFFs that are part of the same SFP could be verified. After the reachability of SFFs has already been verified, SFFs that serve an SF may be used as a test packet source. In such a case, an SFF can act as a proxy for another element within the service function chain.

REQ8: SFC OAM **MUST** be able to trigger on-demand FM remotely with responses being directed toward the initiator of the remote request.

The conformance of the SFC Echo Request/Reply mechanism to these requirements is reflected below:

REQ1: Fate sharing via the SFC Echo Request/Reply defined in Section 6.

REQ2: Continuity monitoring via the SFP tracing defined in Section 6.5.4.

REQ3: Remote defect detection via the SFC Echo Request/Reply defined in Section 6.

REQ4: Connectivity verification via the SFP tracing defined in Section 6.5.4.

REQ5: Fault localization via verification of the SFP consistency defined in Section 6.6.

REQ6: SFP tracing as described in Section 6.5.4 and verification of SFP consistency as defined in Section 6.6.

REQ7: Discover and exercise available RSPs via trace defined in Section 6.5.4.

REQ8: Can be addressed by adding the proxying capability to the SFC Echo Request/Reply described in this document. [RFC7555] describes an example of a proxy function for an Echo Request. Specification of a proxy function for SFC Echo Request is outside the scope of this document.

4. Active OAM Identification in the NSH

SFC active OAM combines OAM commands and/or data included in a message that immediately follows the NSH. To identify the SFC active OAM message, the Next Protocol field **MUST** be set to SFC Active OAM (0x07) (Section 9.1). The O bit in the NSH **MUST** be set, according to [RFC9451]. A case when the O bit is clear and the Next Protocol field value is set to SFC Active OAM (0x07) is considered an erroneous combination. An implementation **MUST** report it. Although the notification mechanism is outside the scope of this specification, note that it **MUST** include rate-limiting control. The packet **SHOULD** be dropped. An implementation **MAY** have control to enable the processing of the OAM payload.

5. SFC Active OAM Header

SFC OAM is required to perform multiple tasks. Several active OAM protocols could be used to address all the requirements. When IP/UDP encapsulation of an SFC OAM control message is used, protocols can be demultiplexed using the destination UDP port number. But an extra IP/UDP header, especially in an IPv6 network, adds overhead compared to the length of an Active OAM Control Packet (e.g., BFD Control packet [RFC5880]). In some environments, for example, when measuring performance metrics, it is beneficial to transmit OAM packets in a broad range of lengths to emulate application traffic closer. This document defines an Active OAM Header (Figure 2) to demultiplex active OAM protocols on SFC.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   V   | Msg Type  | Reserved  |           Length              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~               SFC Active OAM Control Packet                   ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: SFC Active OAM Header

V - a four-bit field that indicates the current version of the SFC Active OAM Header. The current value is 0. The version number is to be incremented whenever a change is made that affects the ability of an implementation to parse or process the SFC Active OAM Header correctly, for example, if syntactic or semantic changes are made to any of the fixed fields.

Msg Type - a six-bit field that identifies the OAM protocol, e.g., the Echo Request/Reply.

Reserved - a six-bit field. It **MUST** be zeroed on transmission and ignored on receipt.

Length - a two-octet field that is the length of the SFC Active OAM Control Packet in octets.

6. Echo Request/Reply for SFC

The Echo Request/Reply is a well-known active OAM mechanism extensively used to verify a path's continuity, detect inconsistencies between a state in control and the data planes, and localize defects in the data plane. ICMP ([RFC0792] for IPv4 and [RFC4443] for IPv6 networks) and MPLS [RFC8029] are examples of broadly used active OAM protocols based on the Echo Request/Reply principle. The SFC Echo Request/Reply control message (format is presented in Figure 3) is an instance of the SFC Active OAM Control Packet that is a part of the SFC Active OAM Header (Figure 2).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Echo Request Flags       |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Echo Type   |   Reply Mode  |  Return Code  |Return Subcode |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sender's Handle                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Sequence Number                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                              TLVs                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: SFC Echo Request/Reply Format

The interpretation of the fields is as follows:

Echo Request Flags - a two-octet bit vector field. Section 9.2.2 requests IANA to create a new registry for flags. This specification defines all flags for future use. Flags **MUST** be zeroed on transmission and ignored on receipt.

Reserved - a two-octet field. It **MUST** be zeroed on transmission and ignored on receipt.

Echo Type - a one-octet field that reflects the packet type. SFC Echo Request/Reply Echo Types, defined in this document, are listed in Section 9.2.3.

Reply Mode - a one-octet field. It defines the type of the return path requested by the sender of the Echo Request.

Return Code and Return Subcode - one-octet fields each. These can be used to inform the sender about the result of processing its request. For all Return Code values defined in this document (Section 9.2.5), the value of the Return Subcode field **MUST** be set to zero.

Sender's Handle - a four-octet field. It **MUST** be filled in by the sender of the Echo Request and returned unchanged by the Echo Reply sender (if a reply is being sent). The sender of the Echo Request **SHOULD** use a pseudorandom number generator [RFC4086] to set the value of the Sender's Handle field. In some use cases, an implementation **MAY** use the Sender's Handle for proprietary signaling as long as the system that receives the SFC Echo Request doesn't alter the value of the Sender's Handle field but copies it into the SFC Echo Reply.

Sequence Number - a four-octet field. It is assigned by the sender and can be, for example, used to detect missed replies. The initial Sequence Number contains an unsigned integer that wraps around. It **MUST** be pseudorandomly generated [RFC4086] and then **SHOULD** be monotonically increasing in the course of the test session. If a reply is sent, the sender of the SFC Echo Reply message **MUST** copy the value from the received SFC Echo Request.

TLV is a variable-length construct whose length is multiple four-octet words. Multiple TLVs **MAY** be placed in an SFC Echo Request/Reply packet. None, one, or more sub-TLVs may be enclosed in the value part of a TLV, subject to the semantics of the (outer) TLV. If no TLVs are included in an SFC Echo Request/Reply, the value of the Length field in the SFC Active OAM Header **MUST** be 16 octets. Figure 4 presents the format of an SFC Echo Request/Reply TLV, where the fields are defined as follows:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |   Reserved    |            Length             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~                             Value                             ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: SFC Echo Request/Reply TLV Format

Type - a one-octet field that characterizes the interpretation of the Value field. Type values are allocated according to Section 9.2.6.

Reserved - a one-octet field. The field **MUST** be zeroed on transmission and ignored on receipt.

Length - a two-octet field equal to the Value field's length in octets as an unsigned integer.

Value - a variable-length field. The value of the Type field determines its interpretation and encoding.

6.1. Return Codes

The value of the Return Code field **MUST** be set to zero by the sender of an Echo Request. The receiver of said Echo Request **MUST** set it to one of the values in IANA's "SFC Echo Return Codes" registry (Section 9.2.5) in the corresponding Echo Reply that it generates.

6.2. Authentication in Echo Request/Reply

Authentication can be used to protect the integrity of the information in the SFC Echo Request and/or Echo Reply. In [RFC9145], a variable-length Context Header has been defined to protect the integrity of the NSH and the payload. The header can also be used for the optional encryption of sensitive metadata. The MAC#1 Context Header is more suitable for the integrity protection of SFC active OAM, particularly of the SFC Echo Request and Echo Reply, as defined in this document. On the other hand, using the MAC#2 Context Header allows the detection of mishandling of the O bit by a transient SFC element.

6.3. SFC Echo Request Transmission

The SFC Echo Request control packet **MUST** use the appropriate underlay network encapsulation of the monitored SFP. The Echo Request **MUST** set the O bit in the NSH, as defined in [RFC9451]. The NSH **MUST** be immediately followed by the SFC Active OAM Header defined in Section 4. The Echo Type field's value in the SFC Active OAM Header **MUST** be set to the SFC Echo Request/Reply value (1), per Section 9.2.1.

The value of the Reply Mode field **MUST** be set to one of the following:

Do Not Reply (1) - This is the value 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. Ways of learning the identity of that node are outside the scope of this specification.

Reply via an IPv4/IPv6 UDP Packet (2) - If an SFC Echo Request is not encapsulated in IP/UDP, then this value requests the use of the Source ID TLV Section 6.3.1).

Reply via Specified Path (4) - This value requests the use of the particular return path specified in the included TLV to verify bidirectional continuity and may also increase the robustness of the monitoring by selecting a more stable path. Section 6.5.1 provides an example of communicating an explicit path for the Echo Reply.

Reply via an IPv4/IPv6 UDP Packet with the data integrity protection (5) - This value requests the use of the MAC Context Header [RFC9145].

Reply via Specified Path with the data integrity protection (7) - This value requests the use of the MAC Context Header [RFC9145].

6.3.1. Source ID TLV

The responder to the SFC Echo Request encapsulates the SFC Echo Reply message in the IP/UDP packet if the Reply Mode is "Reply via an IPv4/IPv6 UDP Packet" or "Reply via an IPv4/IPv6 UDP Packet with the data integrity protection". Because the NSH does not identify the ingress node that generated the Echo Request, information that sufficiently identifies the source **MUST** be included in the message so that the IP destination address and destination UDP port number for IP/UDP encapsulation of the SFC Echo Reply could be derived. The sender of the SFC Echo Request **MUST** include the Source ID TLV (Figure 5).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Source ID  |   Reserved1   |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Port Number          |           Reserved2           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                          IP Address                           ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: SFC Source ID TLV

The fields are defined as follows:

Source ID - the value **MUST** be set to 1 (Section 9.2.6).

Reserved1 - a one-octet field. The field **MUST** be zeroed on transmission and ignored on receipt.

Length - the value equals the length of the data following the Length field counted in octets. The value of the Length field can be 8 or 20. If the value of the field is neither, the Source ID TLV is considered to be malformed.

Port Number - a two-octet field. It contains the UDP port number of the sender of the SFC OAM control message. The value of the field **MUST** be used as the destination UDP port number in the IP/UDP encapsulation of the SFC Echo Reply message.

Reserved2 - a two-octet field. The field **MUST** be zeroed on transmit and ignored on receipt.

IP Address - a field that contains the IP address of the sender of the SFC OAM control message, i.e., IPv4 or IPv6. The value of the field **MUST** be used as the destination IP address in the IP/UDP encapsulation of the SFC Echo Reply message.

A single Source ID TLV for each address family, i.e., IPv4 and IPv6, **MAY** be present in an SFC Echo Request message. If the Source ID TLVs for both address families are present in an SFC Echo Request message, the SFF **MUST NOT** replicate an SFC Echo Reply but choose the destination IP address for the one SFC Echo Reply it sends based on the local policy. The source IP address used in the IP/UDP encapsulation of the SFC Echo Reply is one of the IP addresses associated with the responder. The value of the Port Number field **MUST** be used as the destination UDP port number in the IP/UDP encapsulation of the SFC Echo Reply message. The responder selects the source UDP port number from the dynamic range of port numbers. If more than one Source ID TLV per the address family is present, the receiver **MUST** use the first TLV and ignore the rest. The Echo Reply message, including relevant TLVs, follows the IP/UDP headers immediately.

6.4. Processing a Received SFC Echo Request

Punting a received SFC Echo Request to the control plane for validation and processing is triggered by one of the following packet processing exceptions: NSH TTL expiration, NSH Service Index expiration, or the receiver is the terminal SFF for an SFP.

An SFF that received the SFC Echo Request **MUST** validate the packet as follows:

1. If the SFC Echo Request is integrity protected, the receiving SFF first **MUST** verify the authentication.

   1.1. Suppose the authentication validation has failed and the Source ID TLV is considered properly formatted. In that case, the SFF **MUST** send an SFC Echo Reply with the Return Code set to 3 ("Authentication failed") and the Subcode set to zero to the system identified in the Source ID TLV (see Section 6.5), according to a rate-limit control mechanism.

   1.2. If the authentication is validated successfully, the SFF that has received an SFC Echo Request verifies the rest of the packet's general consistency.

2. Validate the Source ID TLV, as defined in Section 6.3.1.

   2.1. If the Source ID TLV is determined to be malformed, the received SFC Echo Request processing is stopped, the message is dropped, and the event **SHOULD** be logged, according to a rate-limiting control for logging.

3. The Sender's Handle and Sequence Number fields are not examined but are copied in the SFC Echo Reply message.
4. If the packet is not well formed, i.e., not formed according to this specification, the receiving SFF **SHOULD** send an SFC Echo Reply with the Return Code set to 1 ("Malformed Echo Request received") and the Subcode set to zero under the control of the rate-limiting mechanism to the system identified in the Source ID TLV (see Section 6.5).
5. If there are any TLVs that the SFF does not understand, the SFF **MUST** send an SFC Echo Reply with the Return Code set to 2 ("One or more of the TLVs was not understood") and set the Subcode to zero. Also, the SFF **MAY** include an Errored TLVs TLV (Section 6.4.1) that, as sub-TLVs, contains only the misunderstood TLVs.
6. If the consistency check of the received Echo Request succeeded, i.e., the Echo Request is deemed properly formed, then the SFF at the end of the SFP **MUST** send an SFC Echo Reply with the Return Code set to 5 ("End of the SFP") and the Subcode set to zero.
7. If the SFF is not at the end of the SFP and the NSH TTL value is 1, the SFF **MUST** send an SFC Echo Reply with the Return Code set to 4 ("SFC TTL Exceeded") and the Subcode set to zero.
8. In all other cases, for the validated Echo Request message, a transit, i.e., not at the end of the SFP, SFF **MUST** send an SFC Echo Reply with the Return Code set to 0 ("No Error") and the Subcode set to zero.

6.4.1. Errored TLVs TLV

If the Return Code for the Echo Reply is determined as 2 ("One or more of the TLVs was not understood"), the Errored TLVs TLV might be included in an Echo Reply. The use of this TLV is meant to inform the sender of an Echo Request of TLVs either not supported by an implementation or parsed and found to be in error.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Errored TLVs |    Reserved   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Value                             |
.                                                               .
.                                                               .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Errored TLVs TLV

The fields are defined as follows:

Errored TLVs - the field **MUST** be set to 2 (Section 9.2.6).

Reserved - the field **MUST** be zeroed on transmission and ignored on receipt.

Length - the value equals to length of the Value field in octets.

Value - the field contains the TLVs, encoded as sub-TLVs (as shown in Figure 7), that were not understood or failed to be parsed correctly.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Sub-TLV Type |    Reserved   |        Sub-TLV Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                         Sub-TLV Value                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Not Understood or Failed TLV as a Sub-TLV

The fields are defined as follows:

Sub-TLV Type - a copy of the first octet of the TLV that is not understood or failed to be parsed.

Reserved - **MUST** be zeroed on transmission and ignored on receipt.

Sub-TLV Length - the value equals the value of the Length field of the errored TLV.

Sub-TLV Value - the field contains data that follows the Length field in the errored TLV.

6.5. SFC Echo Reply Transmission

The Reply Mode field directs whether and how the Echo Reply message should be sent. The Echo Request sender **MAY** use TLVs to request that the corresponding Echo Reply be transmitted over the specified path. For example, a TLV that specifies the return path of the Echo Reply if the Return Mode in the Echo Request is set to Reply via Specified Path (4) is described in Section 6.5.1. Value 1 is the "Do Not Reply" mode and suppresses the Echo Reply packet transmission. The value 2 of the Reply Mode field requests sending the Echo Reply packet out-of-band as an IPv4/IPv6 UDP packet.

6.5.1. Reply Service Function Path TLV

While the SFC Echo Request always traverses the SFP it is directed to by using the NSH, the corresponding Echo Reply usually is sent without the NSH. In some cases, an operator might choose to direct the responder to send and Echo Reply with the NSH over a particular SFP. This section defines a new TLV, i.e., Reply Service Function Path TLV, for Reply via Specified Path mode of the SFC Echo Reply.

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

The Reply Service Function Path TLV carries the information that sufficiently identifies the return SFP that the SFC Echo Reply message is expected to follow. The format of Reply Service Function Path TLV is shown in Figure 8.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Reply SFP   |    Reserved   |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Reply Service Function Path Identifier     | Service Index |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: SFC Reply TLV Format

The fields are defined as follows:

Reply SFP (3) - identifies the TLV that contains information about the SFC Reply path.

Reserved - **MUST** be zeroed on transmission and ignored on receipt.

Length - the value **MUST** be equal to 4.

Reply Service Function Path Identifier - a three-octet field that contains the SFP identifier for the path that the SFC Echo Reply message is requested to be sent over.

Service Index - a one-octet field. The value is set to the value of the Service Index field in the NSH of the SFC Echo Reply message.

6.5.2. Theory of Operation

[RFC7110] defines a mechanism to control the return path for the MPLS Label Switched Path (LSP) Echo Reply. In the SFC's case, the return path is an SFP along which the SFC Echo Reply message **MUST** be transmitted. Hence, the Reply Service Function 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 the Reply Mode field to "Reply via Specified Path", defined in Section 6.3, and if the specified path is an SFC path, the Request **MUST** include the Reply Service Function Path TLV. The Reply Service Function Path TLV consists of the identifier of the reverse SFP and an appropriate Service Index.

If the NSH of the received SFC Echo Request includes the MAC Context Header, the packet's authentication **MUST** be verified before using any data, as defined in Section 6.4.

The destination SFF of the SFP being tested and the SFF at which the NSH TTL expired (as per [RFC8300]) are referred to as responding SFFs. The processing described below equally applies to both cases.

If the Echo Request message with the Reply Service Function Path TLV received by the responding SFF has the Reply Mode value of "Reply via Specified Path" but no Reply Service Function Path TLV is present, then the responding SFF **MUST** send an Echo Reply with the Return Code set to 6 ("Reply Service Function Path TLV is missing"). If the responding SFF cannot find the requested SFP, it **MUST** send an Echo Reply with the Return Code set to 7 ("Reply SFP was not found") and include the Reply Service Function Path TLV from the Echo Request message.

Suppose the SFC Echo Request receiver cannot determine whether the specified return path SFP has the route to the initiator. In that case, it **SHOULD** set the value of the Return Code field to 8 ("Unverifiable Reply Service Function Path"). The receiver **MAY** drop the Echo Request when it cannot determine whether the SFP's return path has the route to the initiator. When sending the Echo Request, the sender **SHOULD** choose a proper source address according to the specified return path SFP to help the receiver find the viable return path.

6.5.2.1. Bidirectional SFC Case

The ability to specify the return path for an Echo Reply might be used in the case of bidirectional SFC. The egress SFF of the forward SFP might not be co-located with a classifier of the reverse SFP, and thus, the egress SFF has no information about the reverse path of SFC. Because of that, even for bidirectional SFC, a reverse SFP needs to be indicated in a Reply Service Function Path TLV in the Echo Request message.

6.5.3. SFC Echo Reply Reception

An SFF **SHOULD NOT** accept the SFC Echo Reply unless the received message passes the following checks:

- the received SFC Echo Reply is well formed;
- the matching SFC Echo Request is found, that is, the value of the Sender's Handle in the Echo Request sent is equal to the value of Sender's Handle in the Echo Reply received;
- the Sequence Number in the Echo Reply received matches the Sequence Number of one of the outstanding transmitted Echo Requests; and
- all other checks passed.

6.5.4. Tracing an SFP

The SFC Echo Request/Reply can be used to isolate a defect detected in the SFP and trace an RSP. As with the ICMP Echo Request/Reply [RFC0792] and the MPLS Echo Request/Reply [RFC8029], this mode is referred to as "traceroute". In the traceroute mode, the sender transmits a sequence of SFC Echo Request messages starting with the NSH TTL value set to 1 and is incremented by 1 in each next Echo Request packet. The sender stops transmitting SFC Echo Request packets when the Return Code in the received Echo Reply equals 5 ("End of the SFP").

Suppose a specialized information element (e.g., IPv6 Flow Label [RFC6437] or Flow ID [RFC9263]) is used for distributing the load across Equal Cost Multipath or Link Aggregation Group paths. In that case, such an element **SHOULD** also be used for the SFC OAM traffic. Doing so is meant to induce the SFC Echo Request to follow the same RSP as the monitored flow.

6.6. The Use of the Consistency Verification Request Message

The consistency of an SFP can be verified by comparing the view of the SFP from the control or management plane with information collected from traversing by an SFC Echo Request/Reply message (Figure 3). The sender of an SFP Consistency Verification Request (CVReq) message **MUST** set the value of the SFC Echo Request/Reply Echo Type field to 3 ("SFP Consistency Verification Request"). The sender of an SFP Consistency Verification Reply (CVRep) message **MUST** set the value of the SFC Echo Request/Reply Echo Type field to 4 ("SFP Consistency Verification Reply"). All processing steps of SFC Echo Request and Echo Reply messages described in Sections 6.3 through 6.5 apply to the processing of CVReq and CVRep, respectively.

Every SFF that receives a CVReq message **MUST** perform the following actions:

- Collect information about the SFs traversed by the CVReq packet and send it to the ingress SFF as a CVRep packet over an IP network.
- Forward the CVReq to the next downstream SFF if the one exists.

As a result, the ingress SFF collects information about all traversed SFFs and SFs, i.e., information on the actual path the CVReq packet has traveled. That information can be used to verify the SFC's path consistency. The mechanism for the SFP consistency verification is outside the scope of this document.

6.6.1. SFF Information Record TLV

For the received CVReq, an SFF that supports this specification **MUST** include in the CVRep message the information about SFs that are available from that SFF instance for the specified SFP. The SFF **MUST** include the SFF Information Record TLV (Figure 9) in the CVRep message. Every SFF sends back a single CVRep message, including information on all the SFs attached to that SFF on the SFP, as requested in the received CVReq message using the SF Information Sub-TLV (Section 6.6.2).

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |SFF Record TLV |    Reserved   |            Length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |       Service Path Identifier (SPI)           |   Reserved    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    |                   SF Information Sub-TLV                      |
    ~                                                               ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: SFF Information Record TLV

The SFF Information Record TLV is a variable-length TLV that includes the information of all SFs available from the particular SFF instance for the specified SFP. Figure 9 presents the format of an SFF Information Record TLV, where the fields are defined as follows:

SFF Record TLV - the value is (4) (Section 9.2.6).

Reserved - **MUST** be zeroed on transmission and ignored on receipt.

Length - the value equals the sum of lengths of the Service Path Identifier, reserved, and SF Information Sub-TLV fields in octets.

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

SF Information Sub-TLV - the sub-TLV is as defined in Section 6.6.2.

If the NSH of the received SFC Echo Reply includes the MAC Context Header [RFC9145], the authentication of the packet **MUST** be verified before using any data. If the verification fails, the receiver **MUST** stop processing the SFF Information Record TLV and notify an operator. The notification mechanism **SHOULD** include control of rate-limited messages. Specification of the notification mechanism is outside the scope of this document.

6.6.2. SF Information Sub-TLV

Every SFF receiving a CVReq packet **MUST** include the SF characteristic data into the CVRep packet. The format of an SF Information Sub-TLV, included in a CVRep packet, is shown in Figure 10.

After the CVReq message traverses the SFP, all the information about the SFs on the SFP is available from the TLVs included in CVRep messages.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  SF Sub-TLV   |    Reserved   |           Length              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Service Index  |          SF Type              |   SF ID Type  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          SF Identifier                        |
   ~                                                               ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Service Function Information Sub-TLV

SF Sub-TLV - one-octet field. The value is (5) (Section 9.2.6).

Reserved - one-octet field. The field **MUST** be zeroed on transmission and ignored on receipt.

Length - two-octet field. The value of this 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-octet field. It is defined in [RFC9015] and indicates the type of SF, e.g., firewall, Deep Packet Inspection, WAN optimization controller, etc.

SF ID Type - one-octet field with values defined as in Section 9.2.7.

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.

6.6.3. SF Information Sub-TLV Construction

Each SFF in the SFP **MUST** send one and only one CVRep corresponding to the CVReq. If only one SF is attached to the SFF in the SFP, only one SF Information Sub-TLV is included in the CVRep. If several SFs are attached to the SFF in the SFP, the SF Information Sub-TLV **MUST** be constructed as described below in either Section 6.6.3.1 or 6.6.3.2.

6.6.3.1. Multiple SFs as Hops of an SFP

Multiple SFs attached to the same SFF can be the hops of the SFP. The service indexes of these SFs on that SFP will be different. Service Function Types of these SFs could be different or be the same. Information about all SFs **MAY** be included in the CVRep message. Information about each SF **MUST** be listed as separate SF Information Sub-TLVs in the CVRep message. The same SF can even appear more than once in an SFP with a different service index.

An example of the SFP consistency verification procedure for this case is shown in Figure 11. The Service Function Path (SPI=x) is SF1->SF2->SF4->SF3. SF1, SF2, and SF3 are attached to SFF1, and SF4 is attached to SFF2. The CVReq 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 10 contains information for each SF (SF1, SF2, SF3, and SF4).

```
                        SF1         SF2           SF4                SF3
                        +------+------+            |                  |
           CVReq  ......>  SFF1       ......>  SFF2       ......> SFF1
           (SPI=x)             .                   .                  .
                   <............         <..........       <...........
                     CVRep1(SF1,SF2)    CVRep2(SF4)    CVRep3(SF3)
```

Figure 11: Example 1 for CVRep with Multiple SFs

6.6.3.2. Multiple SFs for Load Balance

Multiple SFs may be attached to the same SFF to spread the load; in other words, that means that the particular traffic flow will traverse only one of these SFs. These SFs have the same Service Function Type and Service Index. For this case, the SF ID Type, which must be the same for all of these SFs, appears once, but all the respective SF Identifiers will be listed sequentially in the SF Identifier field of the Service Function Information Sub-TLV (see Figure 10). The number of these SFs can be calculated from the SF ID Type and the value of the Length field of the sub-TLV.

An example of the SFP consistency verification procedure for this case is shown in Figure 12. The Service Function Path (SPI=x) is SF1a/SF1b->SF2a/SF2b. The Service Functions SF1a and SF1b are attached to SFF1, which balances the load among them. The Service Functions SF2a and SF2b are attached to SFF2, which in turn, balances its load between them. The CVReq 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 10 contains information for all SFs at that hop.

```
                            /SF1a                   /SF2a
                            \SF1b                   \SF2b
                              |                       |
                             SFF1                    SFF2
         CVReq   .........>   .           .........>   .
         (SPI=x)                .                       .
                    <............        <...............
             CVRep1(SF1a,SF1b)       CVRep2(SF2a,SF2b)
```

Figure 12: Example 2 for CVRep with Multiple SFs

7. Security Considerations

As an element of SFC OAM and, specifically, based on the NSH, the Echo Request/Reply mechanism described in this document inherits security considerations discussed in [RFC7665] and [RFC8300].

When the integrity protection for SFC active OAM, particularly the SFC Echo Request/Reply, is required, using one of the Context Headers defined in [RFC9145] is **RECOMMENDED**. The MAC#1 Context Header could be more suitable for SFC active OAM because it does not require recalculation of the MAC when the value of the NSH Base Header's TTL field is changed. Integrity protection for SFC active OAM can also be achieved using mechanisms in the underlay data plane. For example, if the underlay is an IPv6 network, i.e., an IP Authentication Header [RFC4302] or IP Encapsulating Security Payload Header [RFC4303], it can be used to provide integrity protection. Confidentiality for the SFC Echo Request/Reply exchanges can be achieved using the IP Encapsulating Security Payload Header [RFC4303]. Also, the security needs for the SFC Echo Request/Reply are similar to those of ICMP ping [RFC0792] [RFC4443] and MPLS LSP ping [RFC8029].

There are at least three approaches to attacking a node in the overlay network using the mechanisms defined in the document. One is a Denial-of-Service attack, i.e., sending SFC Echo Requests to overload an element of SFC. The second may use spoofing, hijacking, replying, or otherwise tampering with SFC Echo Requests and/or Replies to misrepresent and alter the 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., SFFs and/or SFs.

It is **RECOMMENDED** that implementations throttle the number of SFC Echo Request/Reply messages going to the control plane to mitigate potential Denial-of-Service attacks.

Reply and spoofing attacks involving faking or replying to 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 for off-path attackers. A non-matching reply would be discarded.

To protect against unauthorized sources trying to obtain information about the overlay and/or underlay, an implementation **MUST** have means to check that the source of the Echo Request is part of the SFP.

Also, since the SF Information Sub-TLV discloses information about the SFP, the spoofed CVReq packet may be used to obtain network information. Thus, implementations **MUST** provide a means of checking the source addresses of CVReq messages, as specified in Section 6.3.1 ("Source ID TLV"), against an access list before accepting the message.

8. Operational Considerations

This section provides information about operational aspects of the SFC NSH Echo Request/Reply according to recommendations in [RFC5706].

The SFC NSH Echo Request/Reply provides essential OAM functions for network operators. The SFC NSH Echo Request/Reply is intended to detect and localize defects in SFC. For example, by comparing results of the trace function in operational and failed states, an operator can locate the defect, e.g., the connection between SFF1 and SFF2 (Figure 1). After narrowing down a failure to an overlay link, a more specific failure location can be determined using OAM tools in the underlay network. The mechanism defined in this document can be used on demand or for periodic validation of an SFP or RSP. Because the protocol makes use of the control plane, which may have limited capacity, an operator must be able to rate limit Echo Request and Echo Reply messages. A reasonably selected default interval between Echo Request control packets can provide additional benefit for an operator. If the protocol is incrementally deployed in the NSH domain, SFC elements, e.g., Classifier or SFF, that don't support SFC active OAM will discard the protocol's packets. If SFC uses a reclassification along the SFP or when the principle of load balancing is unknown, the fate sharing between data and active OAM packets cannot be guaranteed. As a result, the OAM outcome might not reflect the state of the entire SFC properly but only its segment. In general, it is an operational task to consider the cases where active OAM may not share fate with the monitored SFP. The SFC NSH Echo Request/Reply also can be used in combination with the existing mechanisms discussed in [RFC8924], filling the gaps and extending their functionalities.

Management of the SFC NSH Echo Request/Reply protocol can be provided by a proprietary tool, e.g., command line interface, or based on a data model that is structured or standardized.

9. IANA Considerations

The terms used in the IANA considerations below are intended to be consistent with [RFC8126].

9.1. SFC Active OAM Protocol

IANA has assigned the following new type in the "NSH Next Protocol" registry within the "Network Service Header (NSH) Parameters" group of registries:

Next Protocol	Description	Reference
0x07	SFC Active OAM	RFC 9516

Table 1: SFC Active OAM Protocol

9.2. SFC Active OAM

IANA has created the "Service Function Chaining (SFC) Active Operations, Administration, and Maintenance (OAM)" group of registries, which contains the registries described in the following subsections.

9.2.1. SFC Active OAM Message Types

IANA has created the "SFC Active OAM Message Types" registry as follows:

Registry Name: SFC Active OAM Message Types

Assignment Policy:
0 - 31 IETF Review
32 - 62 First Come First Served

Reference: RFC 9516

Value	Description	Reference
0	Reserved	RFC 9516
1	SFC Echo Request/Reply	RFC 9516
2 - 62	Unassigned	
63	Reserved	RFC 9516

Table 2: SFC Active OAM Message Types

9.2.2. SFC Echo Request Flags

IANA has created the "SFC Echo Request Flags" registry to track the assignment of the 16 flags in the SFC Echo Request Flags field of the SFC Echo Request message. The flags are numbered from 0 (the most significant bit is transmitted first) to 15.

IANA has created the "SFC Echo Request Flags" registry as follows:

Registry Name: SFC Echo Request Flags

Assignment Policy:
0 - 15 Standards Action

Reference:
RFC 9516

Bit Number	Description	Reference
0 - 15	Unassigned	

Table 3: SFC Echo Request Flags

9.2.3. SFC Echo Types

IANA has created the "SFC Echo Types" registry as follows:

Registry Name: SFC Echo Types

Assignment Policy:

0 - 175	IETF Review
176 - 239	First Come First Served
240 - 251	Experimental Use
252 - 254	Private Use

Reference: RFC 9516

Value	Description	Reference
0	Reserved	RFC 9516
1	SFC Echo Request	RFC 9516
2	SFC Echo Reply	RFC 9516
3	SFP Consistency Verification Request	RFC 9516
4	SFP Consistency Verification Reply	RFC 9516
5 - 239	Unassigned	
240 - 251	Reserved for Experimental Use	RFC 9516
252 - 254	Reserved for Private Use	RFC 9516
255	Reserved	RFC 9516

Table 4: SFC Echo Types

9.2.4. SFC Echo Reply Modes

IANA has created the "SFC Echo Reply Modes" registry as follows:

Registry Name: SFC Echo Reply Modes

Assignment Policy:

0 - 175	IETF Review
176 - 239	First Come First Served
240 - 251	Experimental Use
252 - 254	Private Use

Reference: RFC 9516

Value	Description	Reference
0	Reserved	RFC 9516
1	Do Not Reply	RFC 9516
2	Reply via an IPv4/IPv6 UDP Packet	RFC 9516
3	Unassigned	
4	Reply via Specified Path	RFC 9516
5	Reply via an IPv4/IPv6 UDP Packet with the data integrity protection	RFC 9516
6	Unassigned	
7	Reply via Specified Path with the data integrity protection	RFC 9516
8 - 239	Unassigned	
240 - 251	Reserved for Experimental Use	RFC 9516
252 - 254	Reserved for Private Use	RFC 9516
255	Reserved	RFC 9516

Table 5: SFC Echo Reply Modes

9.2.5. SFC Echo Return Codes

IANA has created the "SFC Echo Return Codes" registry as follows:

Registry Name: SFC Echo Return Codes

Assignment Policy:

0 - 191	IETF Review
192 - 251	First Come First Served

252 - 254 Private Use

Reference: RFC 9516

Value	Description	Reference
0	No Error	RFC 9516
1	Malformed Echo Request received	RFC 9516
2	One or more of the TLVs was not understood	RFC 9516
3	Authentication failed	RFC 9516
4	SFC TTL Exceeded	RFC 9516
5	End of the SFP	RFC 9516
6	Reply Service Function Path TLV is missing	RFC 9516
7	Reply SFP was not found	RFC 9516
8	Unverifiable Reply Service Function Path	RFC 9516
9 - 251	Unassigned	
252 - 254	Reserved for Private Use	RFC 9516
255	Reserved	RFC 9516

Table 6: SFC Echo Return Codes

9.2.6. SFC Active OAM TLV Types

IANA has created the "SFC Active OAM TLV Types" registry as follows:

Registry Name: SFC Active OAM TLV Types

Assignment Policy:
0 - 175 IETF Review
176 - 239 First Come First Served
240 - 251 Experimental Use
252 - 254 Private Use

Reference: RFC 9516

Value	Description	Reference
0	Reserved	RFC 9516
1	Source ID TLV	RFC 9516
2	Errored TLVs	RFC 9516
3	Reply Service Function Path Type	RFC 9516
4	SFF Information Record Type	RFC 9516
5	SF Information	RFC 9516
6 - 239	Unassigned	
240 - 251	Reserved for Experimental Use	RFC 9516
252 - 254	Reserved for Private Use	RFC 9516
255	Reserved	RFC 9516

Table 7: SFC Active OAM TLV Types

9.2.7. SF Identifier Types

IANA has created the "SF Identifier Types" as follows:

Registry Name: SF Identifier Types

Assignment Policy:

0 - 191	IETF Review
192 - 251	First Come First Served
252 - 254	Private Use

Reference: RFC 9516

Value	Description	Reference
0	Reserved	RFC 9516
1	IPv4	RFC 9516
2	IPv6	RFC 9516
3	MAC	RFC 9516
4 - 251	Unassigned	

Value	Description	Reference
252 - 254	Reserved for Private Use	RFC 9516
255	Reserved	RFC 9516

Table 8: SF Identifier Types

10. References

10.1. 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>.

[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>.

[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>.

[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/info/rfc9015>.

[RFC9145] Boucadair, M., Reddy.K, T., and D. Wing, "Integrity Protection for the Network Service Header (NSH) and Encryption of Sensitive Context Headers", RFC 9145, DOI 10.17487/RFC9145, December 2021, <https://www.rfc-editor.org/info/rfc9145>.

[RFC9451] Boucadair, M., "Operations, Administration, and Maintenance (OAM) Packet and Behavior in the Network Service Header (NSH)", RFC 9451, DOI 10.17487/RFC9451, August 2023, <https://www.rfc-editor.org/info/rfc9451>.

10.2. 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>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[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>.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, DOI 10.17487/RFC5706, November 2009, <https://www.rfc-editor.org/info/rfc5706>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[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>.

[RFC7555] Swallow, G., Lim, V., and S. Aldrin, "Proxy MPLS Echo Request", RFC 7555, DOI 10.17487/RFC7555, June 2015, <https://www.rfc-editor.org/info/rfc7555>.

[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>.

[RFC8595] Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", RFC 8595, DOI 10.17487/RFC8595, June 2019, <https://www.rfc-editor.org/info/rfc8595>.

[RFC8924] Aldrin, S., Pignataro, C., Ed., Kumar, N., Ed., Krishnan, R., and A. Ghanwani, "Service Function Chaining (SFC) Operations, Administration, and Maintenance (OAM) Framework", RFC 8924, DOI 10.17487/RFC8924, October 2020, <https://www.rfc-editor.org/info/rfc8924>.

[RFC9263] Wei, Y., Ed., Elzur, U., Majee, S., Pignataro, C., and D. Eastlake 3rd, "Network Service Header (NSH) Metadata Type 2 Variable-Length Context Headers", RFC 9263, DOI 10.17487/RFC9263, August 2022, <https://www.rfc-editor.org/info/rfc9263>.

Acknowledgments

The authors greatly appreciate the thorough review and the most helpful comments from Dan Wing, Dirk von Hugo, Mohamed Boucadair, Donald Eastlake 3rd, Carlos Pignataro, and Frank Brockners. The authors are thankful to John Drake for his review and the reference to the work on BGP control plane for NSH SFC. The authors express their appreciation to Joel M. Halpern for his suggestion about the load-balancing scenario. The authors greatly appreciate the thoroughness of comments and thoughtful suggestions by Darren Dukes that significantly improved the document.

Contributors

Cui Wang
Individual contributor
Email: lindawangjoy@gmail.com

Zhonghua Chen
China Telecom
No.1835, South PuDong Road
Shanghai
201203
China
Phone: +86 18918588897
Email: chenzhongh@chinatelecom.cn

Authors' Addresses

Greg Mirsky
Ericsson
Email: gregimirsky@gmail.com

Wei Meng
ZTE Corporation
Yuhuatai District
No.50 Software Avenue
Nanjing,
China
Email: meng.wei2@zte.com.cn

Ting Ao
China Mobile
No.889, BiBo Road
Shanghai
201203
China
Phone: +86 17721209283
Email: 18555817@qq.com

Bhumip Khasnabish
Individual contributor
Email: vumip1@gmail.com

Kent Leung
Individual contributor
530 Showers Drive Ste 7
Mountain View, CA 94040
United States of America
Email: mail4kentl@gmail.com

Gyan Mishra
Verizon Inc.
Email: gyan.s.mishra@verizon.com