Network Working Group
Internet-Draft
Intended Status: Proposed Standard

H. Zhang
L. Fourie
Huawei
R. Parker
Affirmed Networks
M. Zarny
Goldman Sachs

Expires: May 24, 2015

December 23, 2014

Service Chain Header
draft-zhang-sfc-sch-03

Abstract This document describes a service chaining header format and encapsulation mechanism that is used to facilitate the forwarding of data packets along the service function chain path. This header also allows for the transport of metadata to support various service chain related functionality.

Table of Contents

1. Introduction

Service chaining is a traffic steering technology for applying an ordered set of network service functions to traffic flows between two endpoints. Network service functions may include NAT, firewall, server load balancing, WAN optimization, and others, many of which can operate up to OSI Layer 7, and run on hardware appliances or virtual machines.

Service function chaining is an enabling technology for providing more agile service delivery required by cloud computing and network functions virtualization. (See [SFC-PS] and [SFC-ARCH] for more detailed discussions on service chaining.)

In order to support service function chaining, a mechanism is needed to carry service function chain (SFC) path and optional metadata information across various SFC entities. Ideally, the mechanism imposes minimal network overhead, works over various transport technologies at different OSI layers, and can accommodate future services. Given the likelihood of innovation in existing and future service functions and the impossibility of predicting what form every type of metadata will take, the mechanism should allow for the flexibility of carrying different types and lengths of metadata for different usage scenarios, and accommodate the addition of new types of service metadata. (See [SFC-META] for metadata considerations for service function chains.)

It is acknowledged that an either out-of-band or in-band mechanism or a combination of both may be utilized to transfer metadata in service function chains. This document describes only an in-band mechanism.

This document proposes the use of the Service Chain Header (SCH), which comprises a fixed-length mandatory portion used for steering purposes and an optional variable-length TLV (Type-Length-Value) approach to carry in-band metadata between service function chaining entities.

The SCH may be used to carry: (1) both SFC path steering information and metadata; (2) only SFC path steering information, in which case the Metadata Length field shall be set to zero; or (3) only metadata, in which case the Path Identifier and SF Index fields shall be set to zero for transmit and ignored upon receipt.

2. Conventions used in this document

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].

3. Terminology

Most of the terminologies used in this document are from [SFC-ARCH], [SFC-FWK] and [SFC-META], and are summarized here for convenience:

Service Function (SF): A network function that provides a value-added service to packet flows. A service function can act at any OSI layer. Service functions include: firewall, DPI (Deep Packet Inspection), NAT, HTTP Header Enrichment function, TCP optimizer, load-balancer, etc.

SF Chain: An ordered list of Service Function instances. See SF Map.

SF Chain Identifier: Identifies an SF Chain (SF Map Index).

SFC-enabled domain: Denotes a network (or a region thereof) that implements SFC.

SF Identifier: A unique identifier that unambiguously identifies an SF within an SFC-enabled domain. SF Identifiers are assigned, configured and managed by the administrative entity that operates the SFC-enabled domain. SF identifiers can be structured as strings, or in other formats. SF Identifiers are not required to be globally unique nor be exposed to or used by another SF-enabled domain.

Service Function Forwarder (SFF): Provides service layer forwarding. An SFF receives frames/packets from an SFC Network Forwarder and forwards the traffic to the associated SF(s) using information contained in the SCH.

SFC Network Forwarder (SNF): Forwards traffic flows along the SFPs they belong to based on the information contained in the SFC encapsulation.

SF Map: Refers to an ordered list of SF identifiers. Each SF Map is identified with a unique identifier called SF Map Index. This is referred to as a service function chain in this document.

SF Locator: An SF Node identifier used to reach the said SF node. A locator is typically an IP address or a FQDN.

Legacy Node: Refers to any node that is not an SF Node nor an SFC Boundary Node. This node can be located within an SFC-enabled domain or outside an SFC-enabled domain.

SF Proxy Node: A Network Element along the data path, to enforce SFC functions on behalf of legacy SF nodes.

SFC Boundary Node (or Boundary Node): Denotes a node that connects one SFC-enabled domain to a node either located in another SFC-

enabled domain or in a domain that is SFC-unaware.

SFC Egress Node (or Egress Node): Denotes an SFC Boundary Node that handles traffic which leaves the SFC-enabled domain the Egress Node belongs to.

SFC Ingress Node (or Ingress Node): Denotes an SFC Boundary Node that handles the traffic entering the SFC-enabled domain the Ingress Node belongs to.

SF Node: Denotes any node within an SFC-enabled domain that embeds one or multiple SFs.

Service Function Instance (SFI): Denotes an instantiation of a service function on a service node, such as a FW instance. A service function can have multiple service instances running on the same SF node with each service instance having its own service profile.

SFC Classifier (or Classifier): An entity that classifies frames or packets for service chaining according to classification rules defined in an SFC Policy Table. Frames/packets are then marked with the corresponding SF Chain Identifier. SFC Classifier can be embedded in an SFC Boundary (Ingress) Node or SF proxy node in which case the Classifier will do the encapsulation after getting the L2/L3 frame/packet from a Legacy SN. The SFC Classifier can also run on an independent (physical or virtual) platform.

Metadata: Provides contextual information about the data packets which traverse a service chain. Metadata can be used to convey contextual information not available at one location in the network to another location in the network where that information is not readily available. While primarily intended for consumption by SF(s), metadata MAY also be interpreted by other SFC entities.

4. Service Chain Header 4.1 Header Format

The Service Chain Header (SCH) consists of a mandatory fixed length part followed by a number of optional variable length metadata as shown in Figure 1. The mandatory fields carry "SFC path" information, which is used to steer the frames or packets through an ordered set of service function instances along the service function chain. The optional variable length fields carry application/service/content related metadata information which can be used by any SFC entities. The optional fields are formatted as Type-Length-Value structures. If any field in the header is not in use, the value of that field MUST be set to zero.

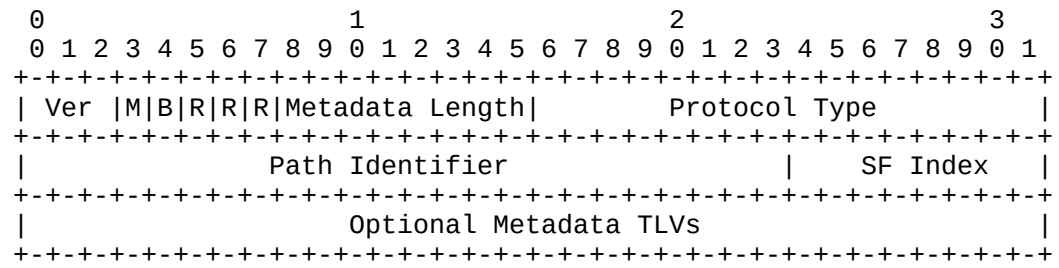


Figure 1: Service Chain Header

4.2 Mandatory Fields

Ver: Represents the Service Chain Header version. This field is 3 bits long, and the current version is 0.

M bit: Indicates that this frame/packet is an operations and management (OAM) packet. SF nodes MUST examine the payload and take appropriate action (i.e. return status information). The M-bit may be used in conjunction with OAM-specific TLVs (See Appendix A for an example). OAM message specifics and handling details are for future study and are outside the scope of this document.

B bit: The B (Bypass) bit, when set to 1, it is used by a Service Function to signal to its Service Function Forwarder that no further packets are to be sent to it for the flow specified in encapsulated packet.

R bits: Reserved bits for future extensions. These bits MUST be set to 0 on transmit, and ignored on receive.

Metadata length: The total length of all of the optional Metadata TLVs, in 4 octet units.

Protocol type: The 2-octet IEEE EtherType of the packet or frame immediately following the entire SCH header [ETYPES].

Path Identifier: Identifies a service function path. It represents a sequence of network locators, one for each service function that is to be invoked. Participating SFC entities MUST use this identifier when selecting the next hop for the packet or frame. The path identifier can also be used for diagnostics and trouble-shooting associated with a service chain. For cases where the SCH is used solely to convey metadata, the Path Identifier is set to 0

on transmit and ignored on receive.

Service Function Index: An index to each service function instance associated with the service path. The service path index can be used to handle loop detection, flow reentry into a previous SF node of the SFC requiring another different service instance treatment, etc. The first service function instance in the path is identified with service index 1. For cases where the SCH is used solely to convey metadata, the Service Function Index is set to 0 on transmit and ignored on receive.

4.3 Optional Metadata Fields

Optional metadata are added after the fixed part of the SCH. Each option is of variable length and has a minimum length of four octets. An optional 3-octet Organizational Unique Identifier (OUI) may be provided to differentiate multiple private number spaces for the Type field. If the OUI is not provided, the Type is assumed to be a registered globally unique type. Any SFC entities MAY add, modify, or remove metadata TLVs.

The Appendix provides some examples of potential metadata TLV types and usage for reference.

Figure 2 shows the format of the TLV:

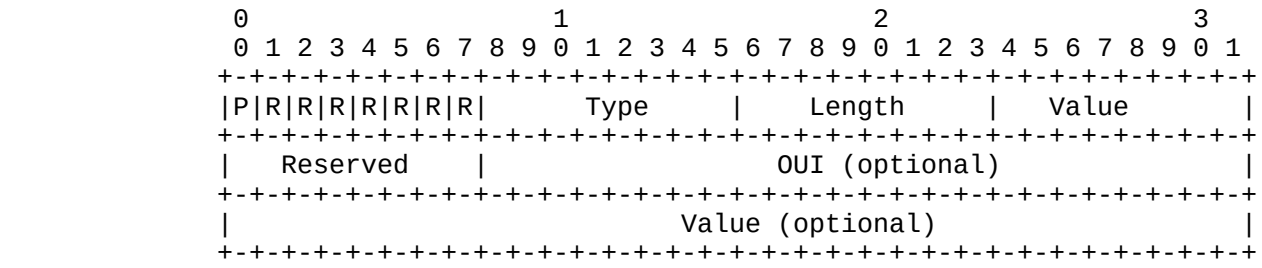


Figure 2: TLV format

P bit: The P (private) bit, when set to P=1, indicates that the optional OUI word (8-bit Reserved and 24-bit OUI) is present in this TLV. P=0 indicates that the optional OUI word is not present.

R bits: Reserved bits for future extensions. These bits MUST be set to 0 on transmit and ignored on receive.

Type: The type of the TLV, interpreted globally or per the OUI if present. A maximum of 256 types may be expressed within any

particular number space.

Length: The total length of the TLV in 4-octet units. A maximum length of 1024 octets may be expressed.

OUI: The 24-bit Organizational Unique Identifier [IEEE-OUI], present only if the P bit is set.

Value: Usage of the 8-bit value in the first word is purely optional and may be useful for certain types of metadata, allowing registered types to fit into one 32-bit word and OUI-based types to fit into two 32-bit words. When the 8-bit value is insufficient, additional words for the value are added to the end of the TLV. The existence of optional Value words is inferred from the content of the Length field and the value of the P bit.

4.3.1 Generic Context Block

The Generic Context Block creates a 16-octet metadata scratchpad to be used in a deployment-specific manner.

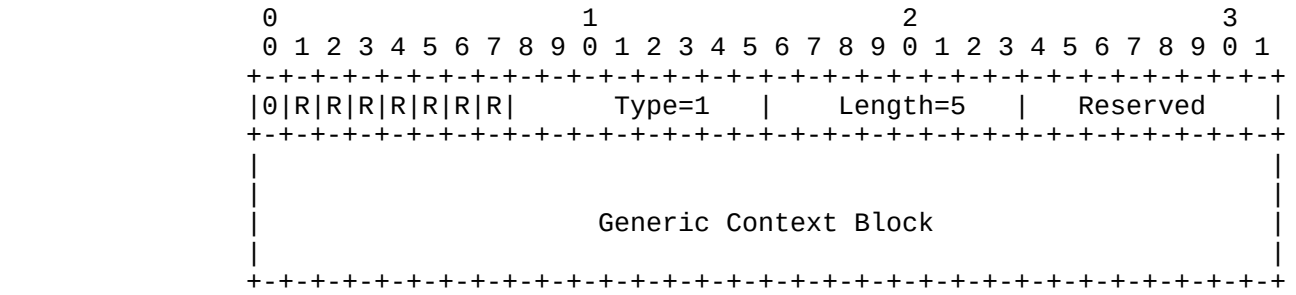


Figure 3: Generic Context Block

4.3.2 Rendered Service Path Identifier

The Rendered Service Path Identifier TLV is used to identify a specific Rendered Service Path (RSP). The Rendered Service Path is the exact sequence of SFFs and SF instances followed for a given flow per [SFC-ARCH].

An RSP may be generated in the control plane with the exact sequence of next-hops and communicated to all involved nodes via control plane or management channels. The exact sequence of SFFs and SF instances is known a priori. This may be viewed as an early binding of SFFs and SF instances to an RSP. This can be satisfied by the Path Identifier alone.

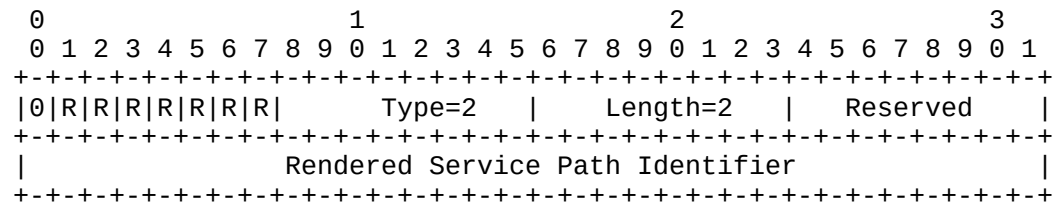


Figure 4: Rendered Service Path Identifier

There is also a need to provide a late binding approach to support load balancing across a group of SF instances, for example, to maintain statefulness. This allows the RSP to be constructed dynamically on a hop-by-hop basis by the the SFFs. The RSP identifies the exact sequence of SF instances to ensure that all traffic for a flow traverses the same SF instances.

When this option is used, a new RSP Identifier is generated for the given SFP ID and is added as an optional metadata TLV to the SCH. The binding of each SFF and SF instance to a RSP is done through the hop-by-hop selection of the SFFs and SF instances as the first packet for a flow traverses the chain.

The packet is sent to each SFF with the SFP ID and the RSP ID. The first SFF will select an instance of the first required SF and bind it to the {SFP ID, RSP ID}. The SFF will then choose a next SFF that hosts at least one instance of the next SF. (The next SFF could be the same SFF). The SF instance selection and binding to the {SFP ID, RSP ID} may be performed as such at each hop.

Once the binding of the SFFs and SF instances is complete, the RSP ID is used to steer all subsequent traffic.

A sufficient number of RSP IDs must be allocated for load balancing purposes.

4.4 Header Associated Operations

The Service Chain Header is processed by SFC-aware entities. These entities include but are not limited to the SFC classifiers, SF nodes, SF forwarding nodes, SFC boundary nodes, and SF proxy nodes. The SCH can be used by any SFC entities. Its use between the SFF and SF is optional and in many cases will be implementation dependent. The typical SFF behavior is simplified due to the presence of the metadata length field in the fixed portion of the header.

These entities can perform the following operations related to the

SCH:

1. Insertion of the Service Chain Header. This occurs at the start of a service chain. An SFC classifier determines which frames/packets are associated with which service chain. The classifier performs this task by matching the incoming frames/packets against certain flow descriptors and assigning them to specific chains. The service classifier MUST then insert appropriate SCH in the frames/packets of a flow that has been assigned to a service chain.

2. Removal of the Service Chain Header. This occurs at the end of the service path. The last Service Network Forwarder (SNF) in the service path MUST remove the SCH from the frames/packets and then forward them to their final destination using its existing transport mechanism. This MUST also be performed by an SF Proxy Node if it is the last forwarding node.

3. Service Path Selection. The Service Network Forwarder (SNF) uses the Service Chain Identification information in the SCH to steer the traffic flow along the SFC path.

4. Service Function Instance Selection. The Service Function Forwarder (SFF) uses the Service Chain Identification information in the SCH to locate the service instance and forward the traffic flow to the service instance. The mapping of the Service Chain Identification to a service instance can be established through the management/control plane, which is out of scope of this document.

5. Service Treatment. The Service Node could use the Service Chain Identification information as well as service metadata in the SCH to locate the service instance associated with the service chain and apply appropriate service treatment.

6. Service Bypass. There are cases, e.g., long-lived flows, where a Service Function does not need to process any further packets for flow and that it should be bypassed. The Service Function signals this by setting the Bypass bit in the SCH of packets that it sends back to the SFF. The flow in question is identified from the encapsulated packet header. When the SFF receives such a packet it marks that flow as an exception flow and does not send any further packets for that flow to the SF but instead forwards the packets to the next SFF or next SF in the chain. The SFF resets the Bypass bit to zero before sending the packet to the next downstream SFF or SF.

5. SCH Encapsulation

The SFC classifier adds the SCH to the original frame or packet (of types defined in [ETYPES]), and then adds the transport header used to forward the frame/packet through the service chain.

The SCH is independent of the underlying transport used to provide reachability amongst the SFC entities. It can be encapsulated inside any transport mechanism. The following examples illustrate how the SCH can be encapsulated in typical transport mechanisms.

5.1 SCH in Overlay GRE Encapsulation

The SCH can be encapsulated in a GRE transport as follows:

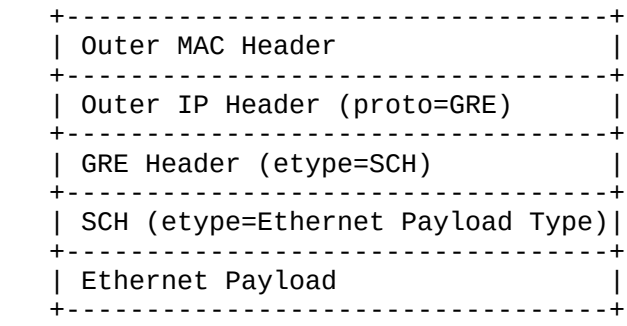


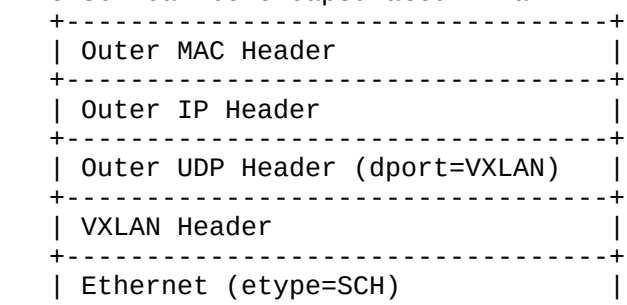
Figure 5: SCH Encapsulation in GRE Transport

Set GRE Header Protocol = SCH Ethertype

SCH inserted between the GRE header and MAC payload

5.2 SCH in Overlay VXLAN Encapsulation

The SCH can be encapsulated in a VXLAN transport as follows:



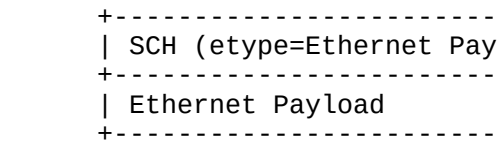


Figure 6: SCH Encapsulation in VXLAN Transport

Set VXLAN's inner Ethernet Ethertype= SCH Ethertype

SCH inserted between the inner Ethernet header and MAC payload

5.3 SCH in IP/UDP Encapsulation

The SCH can be encapsulated in a IP/UDP transport as follows.:

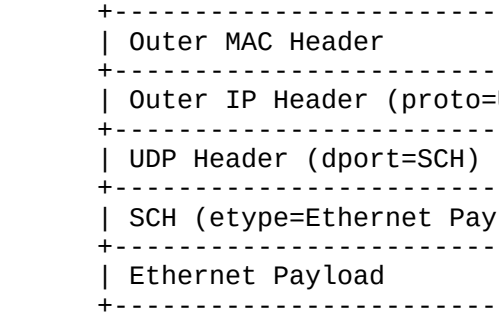


Figure 7: SCH Encapsulation in IP/UDP Transport

Set UDP Header dport = SCH Ethertype, a special UDP port value should be assigned to SCH by IANA

SCH inserted between the IP/UDP header and MAC payload.

5.4 SCH in MAC Encapsulation

The SCH can be encapsulated directly in a MAC transport (VLAN optional) as follows:

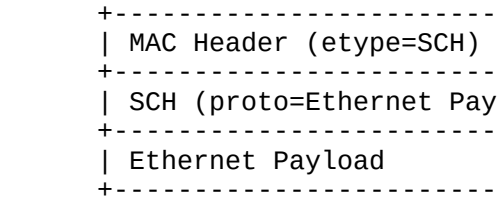


Figure 8: SCH Encapsulation in MAC Transport

Set MAC Header Ethertype = SCH Ethertype

SCH inserted between the MAC header and MAC Payload

6. Security Considerations

Although the SCH can be modified or spoofed by an unauthorized party, various options are available to avoid this.

Existing security protocols RFC 6071 [RFC6071] may be used to encrypt the content of a packet that includes the SCH. It should be noted that encryption and decryption of the SCH will impose a performance penalty. Existing security protocols that provide authenticity and authorization (e.g., RFC 2119 [RFC6071]) can be used.

If possible, the SCH should be used in a controlled network with trusted devices, for example, a data center or a Gi-LAN network, thus reducing the risk of unauthorized header manipulation.

7. IANA and IEEE Considerations

Non-OUI TLV types shall be assigned by IANA.

An EtherType assignment for the SCH will be requested from IEEE.

8. References

8.1 Normative References

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

[RFC0781] Su, Z., "Specification of the Internet Protocol (IP) timestamp option", RFC 781, May 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, February 2011.

8.2 Informative References

[SFC-ARCH] Halpern, J. et al, Service Function Chaining (SFC) Architecture <https://datatracker.ietf.org/doc/draft-ietf-sfc-architecture/>.

[SFC-FWK] Boucadair, M. et al, Service Function Chaining: Framework & Architecture <http://datatracker.ietf.org/doc/draft-boucadair-sfc-framework/>.

[SFC-META] Rijsman, B. and J. Moisand, Metadata Considerations <http://http://datatracker.ietf.org/doc/draft-rijsman-sfc-metadata-considerations/>.

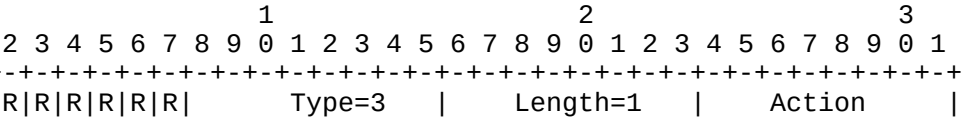
[SFC-PS] Quinn, P., Ed. and T. Nadeau, Ed., "Service Function Chaining Problem Statement", 2014, <http://datatracker.ietf.org/doc/draft-ietf-sfc-problem-statement/>.

9. Appendix A

Appendix A gives some examples of potential metadata TLV types and usage. It is out of scope of the document to define the list of types and usage. For exemplary purposes only, the formats assume globally registered types (i.e., the OUI word is not present).

9.1 OAM Operation

The OAM Operation TLV, used in conjunction with the M-bit in the fixed portion of the SCH, allows a service function to specify an action to be taken by a downstream service function as a result of its service processing.

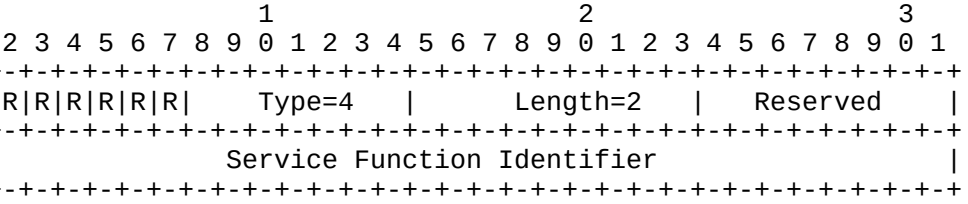


Service actions could include the following:

1 = drop packet
2 = redirect flow
3 = mirror flow
4 = terminate connection

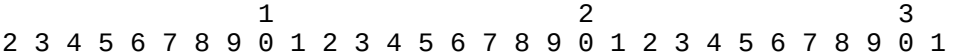
9.2 Service Function Selector

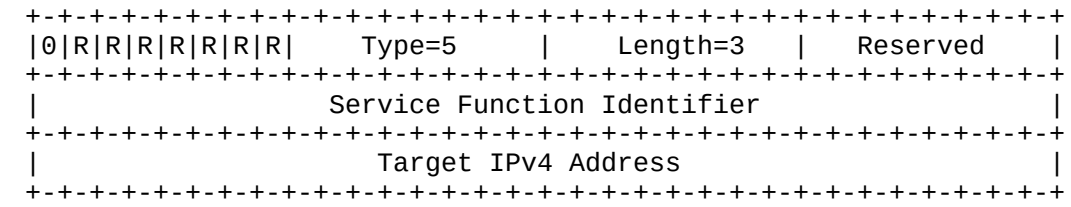
The Service Function Identifier TLV allows a service function to select a specific service function to be used on a downstream service node.



9.3 Target Address

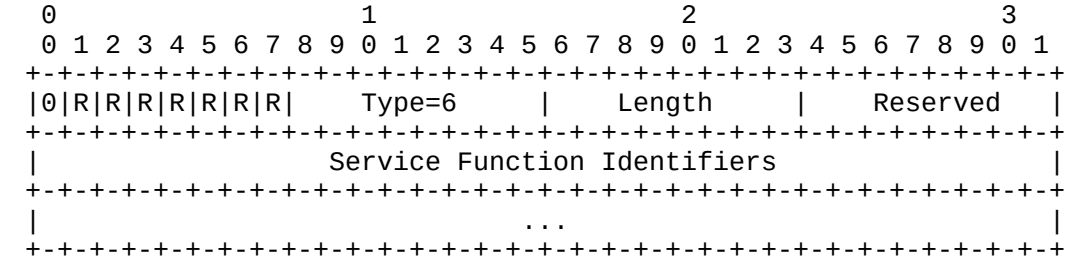
The Target Address TLV allows an original destination IP address to be transported across the service chain to the last service function which restores that IP address to the destination IP address of the original packet.





9.4 OAM Service Function Trace

The OAM Service Function List TLV supports OAM functionality and is used to obtain a list of service functions that have been traversed by the packet. Each service function adds its service function identifier to this list if the OAM Operation TLV indicates an SF Identifier.



10. Acknowledgments

The authors would like to thank Nicolas Bouthors, Linda Dunbar, Lucy Yong and Kevin Glavin for their review and comments.

Authors' Addresses

Hong (Cathy) Zhang
Huawei US R&D

EMail: cathy.h.zhang@huawei.com

Louis Fourie
Huawei US R&D

EMail: louis.fourie@huawei.com

Ron Parker
Affirmed Networks

EMail: ron_parker@affirmednetworks.com

Myo Zarny
Goldman Sachs

EMail: myo.zarny@gs.com