INTERNET-DRAFT Sami Boutros, Ed.
Intended Status: Standard Track VMware
Dharma Rajan
Philip Kippen
Pierluigi Rolando
VMware
Jim Guichard
Huawei
Sam Aldrin
Google

Expires: March 19, 2020 September 16, 2019

Geneve applicability for service function chaining
draft-boutros-nvo3-geneve-applicability-for-sfc-04

Abstract

This document describes the applicability of using Generic Network Virtualization Encapsulation (Geneve), to carry the service function path (SFP) information, and the network service header (NSH) encapsulation. The SFP information will be carried in Geneve option TLV(s).

Status of this Memo

Copyright and License Notice

Table of Contents

1. Introduction

The Service Function Chaining (SFC) Architecture [rfc7665] defines a service function chain (SFC) as (1) the instantiation of an ordered set of service functions and (2) the subsequent "steering" of traffic through them.

SFC defines a Service Function Path (SFP) as the exact set of service function forwarders (SFF)/service functions (SF)s the packet will visit when it actually traverses the network.

An optimized SFP helps to build an efficient Service function chain (SFC) that can be used to steer traffic based on classification rules, and metadata information to provide services for Network Function Virtualization (NFV). Metadata are typically passed between service functions and Service function forwarders SFF(s) along a service function path.

In a Network Virtualization Overlays (NVO3) domain, Network Virtualization Edges (NVE)s can be implemented on hypervisors hosting virtual network functions VNF(s) or cloud native functions CNF(s) implementing service functions, or on CNFs on bare metal servers or on physical routers connected to service function appliances. NVO3 domain uses tunneling and encapsulation protocols such as Geneve to provide connectivity for tenants workloads and service function running in its domain. NVEs in an NVO3 domain are typically controlled by a centralized network virtualization authority NVA.

[RFC8300] defines a new encapsulation protocol, network service header (NSH) to encode the SFP and the metadata.

1.1 Requirement for SFC in NVO3 domain

The requirement is to provide service function chaining in an NVO3 domain without the need to implement yet another control plane for service topology.

1.2 Proposed solution for SFC in NVO3 domain

This document specifies the applicability of using Generic Network Virtualization Encapsulation (Geneve), to carry the service function path (SFP) information, and the network service header (NSH) encapsulation.

The SFP will be implemented using a new Geneve Service Function List (SFL) option for use strictly between Network Virtualization Edges (NVEs) performing the service forwarding function (SFF) in the same Network Virtualization Overlay over Layer 3 NVO3 domain. The next

protocol in the Geneve Header will be the NSH EtherType, 0x894F. The NSH encapsulation will include the Service Path Identifier (SPI) and the Service Index (SI). The NSH SI will serve as an index to the VNF/CNF hop to visit in the SFL.

In the absence of the SFL we would need a service topology control plane. The Geneve overlay will encap the NSH encapsulation and the next protocol on Geneve will be the NSH Ethertype.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Abbreviations

NVO3 Network Virtualization Overlays over Layer 3

OAM Operations, Administration, and Maintenance

TLV Type, Length, and Value

VNI Virtual Network Identifier

NVE Network Virtualization Edge

NVA Network Virtualization Authority

NIC Network interface card

VTEP Virtual Tunnel End Point

Transit device Underlay network devices between NVE(s).

Service Function (SF): Defined in [RFC7665].

Service Function Chain (SFC): Defined in [RFC7665].

Service Function Forwarder (SFF): Defined in [RFC7665].

Service Function Path (SFP): Defined in [RFC7665].

Metadata: Defined in [[draft-ietf-sfc-nsh]

NFV: Network function virtualization.

VNF: Virtual network function

CNF: Cloud native function

4. Geneve Option TLV(s)

4.1 Geneve Service Function List (SFL) Option TLV

Geneve Header:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver|  Opt Len  |O|C|    Rsvd.  |Protocol Type = NSH Ethertype  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Virtual Network Identifier (VNI)       |    Reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Geneve Option Header:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          SFL Option Class     |      Type     |R|R|R| Length  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Variable Option Data                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Followed by the NSH encapsulation which is composed of a 4-byte Base Header, a 4-byte Service Path Header, and optional Context Headers.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Base Header                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Service Path Header                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                Context Header(s)                              ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SFL Option Class = To be assigned by IANA

Type = To be assigned by IANA

'C' bit set, indicating endpoints must drop if they do not recognize this option)

Length = variable.

Variable option data:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|Flags          |Reserved                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~            SF List[0] (32 or 128 bits IPv4/6 address)         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
                              ...
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~            SF List [n] (32 or 128 bits IPv4/6 address)        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                                                             //
//         Optional sub-Type Length Value objects (variable)   //
//                                                             //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Service Function List (SFL) Option TLV.

Reserved: 12 bits. SHOULD be unset on transmission and MUST be ignored on receipt.

Flags:

```
          0 1 2 3 4 5 6 7
         +-+-+-+-+-+-+-+-+
         |H| Unused      |
         +-+-+-+-+-+-+-+-+
```

Figure 2: SFL flags

H-flag: HMAC flag. If set, the HMAC sub-TLV is present and is encoded as the last sub-TLV.

SF List[n]: 32 or 128 bits IPv4/6 addresses representing the nth service function ip address in the List.

The SF List is encoded starting from the last hop of the path. I.e., the first element of the list (SF List [0]) contains the last service function of the path while the last element of the SF List (SF List[n]) contains the first service function in the path.

HMAC sub-TLV is optional and contains the HMAC information. The

HMAC sub-TLV has the following format:

```
 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     |     Length    |          Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      HMAC Key ID (4 octets)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                              //
|                      HMAC (32 octets)                        //
|                                                              //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: SFL HMAC sub-TLV.

Type: to be assigned by IANA (suggested value 1).
Length: 38.
Reserved: 2 octets. SHOULD be unset on transmission and MUST be ignored on receipt.
HMAC Key ID: 4 octets.
HMAC: 32 octets.
HMAC and HMAC Key ID usage is described in Operation section.

The Following applies to the HMAC TLV:

When present, the HMAC sub-TLV MUST be encoded as the last sub-TLV

If the HMAC sub-TLV is present, the H-Flag (Figure 2) MUST be set.

When the H-flag is set, the NVE inspecting the Geneve Service Function List Option TLV MUST find the HMAC sub-TLV in the last 38 octets of the option TLV.

5.. Operation

The mechanisms described in this section should work with both ipv4 and ipv6 for both customer inner payload and Geneve tunnel packets.

5.1 Operation at Ingress

A Source NVE acting as a service function classifier and a service function forwarder can be any node in an NVO3 domain, originating based on a classification policy for some customer inner payload an IP Geneve tunnel packet with the service function list (SFL) option TLV. The service functions in the SFL represent the IP addresses of the service functions that the inner customer packets needs to be inspected by. A controller can program the ingress NVE node to classify traffic and identify a service function paths i.e the set of

service functions in the path. The mechanism through which an SFL is derived by a controller or any other mechanisms is outside of the scope of this document.

The ingress NVE node fills in the list of service functions in the path, to the Geneve Service Function List option TLV, putting the first service function ip address as the last element in the list and the last service function ip address as the first element, setting of the NSH service index to the first element. The ingress NVE node, then, resolves the service first function ip address, to the NVE virtual tunnel endpoint node hosting or directly connected to the service function.

The Geneve tunnel destination is then set to the NVE tunnel endpoint hosting the first service function, and the service index is decremented to n-1 (where n is the number of elements in the SFL), and set on the SFL option TLV. An NSH metadata can also be set on the packet by the NVE ingress node.

The Geneve packet is sent out towards the first NVE.

HMAC optional sub-TLV may be set too.

5.2 Operation at each NVE along the service function path

The NVE node along the service function path corresponding to the Geneve tunnel destination of the packet, receives the packet, perform the service function forwarder function and identifies the SFL option, and locates the service function in the list based on the service index.

The Geneve tunnel header and option TLV(s) will be stripped and the packet will be delivered to the service function or virtual network function VNF or CNF. The NVE maintains state related to the association of the SFL option TLV and the NSH service path identifier. The packet passed to the service function encaped with the NSH header and NSH context, if the SF is NSH aware, other encapsulations like vlan or q- in-q encap may be used to pass the metadata and NSH SPI to the SF too.

When the packet comes back from the service function along with the service path identifier (SPI) context, based on SPI on the packet the NVE acting as the SFF will be able to locate the SFL option TLV.

If the metadata context indicate (1) that some service functions need to be bypassed the NVE should bypass in the SFL the service functions to be skipped and update the NSH service index accordingly. (2) A new

classification need to be performed on the packet, in that case the NVE can re-classify the packet or sent it to an NVE node capable of classification.

The NVE node, then, resolves the next service function ip address, to the NVE virtual tunnel endpoint node hosting or directly connected to the service function.

The NVE then sets the Geneve tunnel destination to the next NVE tunnel endpoint, and the NSH service index is decremented by 1 and set on the NSH Header, along with other NSH metadata option TLV.

The Geneve ip packet is sent out towards the next NVE.

5.3 Operation at Egress

At the last NVE node along the service function path, the NVE locates the service function in the SFL option TLV based on the NSH service index. The service index received at the last NVE node will be set to 1.

The Geneve tunnel header and option TLV(s) will be stripped and the packet will be delivered to the service function. The NVE maintains state related to the association of the SFL option TLV and the NSH service path identifier. The packet passed to the service function encaped with the NSH header and NSH context, if the SF is NSH aware, other encapsulations like vlan or q-in-q encap may be used to pass the metadata and NSH SPI to the SF too.

When the packet comes back from the service function, based on NSH SPI on the packet or based the NVE will be able to locate the SFL option TLV.

Given that the service index will be set to 1, the last NVE will now deliver the packet to the NVE hosting or directly connected to the inner packet destination.

A packet received with a service function index of 0 MUST be dropped.

6. Security Considerations

Only NVE(s) that are the destinations of the Geneve tunnel packet will be inspecting the List of Service Function next hops Option. A Source routing option has some well-known security issues as described in [RFC4942] and [RFC5095].

The main use case for the use of the Geneve List of Service Function next hops Option will be within a single NVO3 administrative domain

where only trusted NVE nodes are enabled and configured participate, this is the same model as in [RFC6554].

NVE nodes MUST ignore the Geneve List of Service Function next hops Option created by outsiders based on NVA or trusted control plane information.

There is a need to prevent non-participating NVE node from using the Geneve Service Function List option TLV, as described in [draft-ietf-6man-segment-routing-header], we will use a security sub-TLV in the Service Function List option TLV, the security sub-TLV will be based on a key-hashed message authentication code (HMAC).

HMAC sub-TLV will contain:

HMAC Key-id, 32 bits wide;

HMAC, 256 bits wide (optional, exists only if HMAC Key-id is not 0).

The HMAC field is the output of the HMAC computation (per RFC 2104 [RFC2104]) using a pre-shared key identified by HMAC Key-id and of the text which consists of the concatenation of:

The source IPv4/IPv6 Geneve tunnel address

Version and Flags

HMAC Key-id.

All addresses in the List.

The purpose of the HMAC optional sub-TLV is to verify the validity, the integrity and the authorization of the Geneve Service Function List option TLV itself.

The HMAC optional sub-TLV is located at the end of the Geneve Service Function List option TLV.

The HMAC Key-id field serves as an index to the right combination of pre-shared key and hash algorithm and except that a value of 0 means that there is no HMAC field.

The HMAC Selection of a hash algorithm and Pre-shared key management will follow the procedures described in [draft-ietf-6man-segment-routing-header] section 6.2.

7. Management Considerations

The Source NVE can receive its information through any form of north bound Orchestrator. These could be from any open networking automation platform (ONAP) or others. The ingress to egress tunnel is built and managed by the service function classifier and service function forwarder by each node in an NVO3 domain. Error handling, is handled by the classifier reporting to north bound management systems.

8. Acknowledgements

The authors would like to acknowledge Jim Guichard for his feedback and valuable comments to this document.

9. IANA Considerations

This document makes the following registrations in the "Geneve Option Class" registry maintained by IANA:

Suggested Value	Description	Reference
XX	Geneve List of Service Function next hops	This document

In addition, this document request IANA to create and maintain a new Registry: "Geneve List of Service Function next hops Type-Value Objects".

The following code-point are requested from the registry:

Registry: Geneve List of Service Function next hops Type-Value Objects

Suggested Value	Description	Reference
1	HMAC TLV	This document

10. References

10.1 Normative References

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

10.2 Informative References

[Geneve] "Generic Network Virtualization Encapsulation", [I-D.ietfnvo3-geneve]

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

[RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <http://www.rfc-editor.org/info/rfc4942>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <http://www.rfc-editor.org/info/rfc6554>.

[draft-ietf-6man-segment-routing-header] Previdi, S., et all, "IPv6 Segment Routing Header (SRH)", July 20, 2017, draft-ietf-6man-segment-routing-header-07

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <http://www.rfc-editor.org/info/rfc5095>.

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

Authors' Addresses

Sami Boutros
VMware
Email: boutross@vmware.com

Dharma Rajan
VMware
Email: drajan@vmware.com

Philip Kippen
VMware
Email: pkippen@vmware.com

Pierluigi Rolando
VMware
Email: prolando@vmware.com

Jim Guichard
Huawei
Email: james.n.guichard@huawei.com

Sam Aldrin
Google
Email:aldrin.ietf@gmail.com