INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros, Ed.
VMware
Dharma Rajan
Philip Kippen
Pierluigi Rolando
VMware

Expires: March 18, 2019

September 14, 2018

Geneve applicability for service function chaining
draft-boutros-nvo3-geneve-applicability-for-sfc-02

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).

Table of Contents

1. Introduction 3
1.1 Requirement for SFC in NVO3 domain 3
1.2 Proposed solution for SFC in NVO3 domain 3
2. Terminology 4
3. Abbreviations 4
4. Geneve Option TLV(s) 5
4.1 Geneve Service Function List (SFL) Option TLV 5
5.. Operation 7
5.1 Operation at Ingress 7
5.2 Operation at each NVE along the service function path . . . 8
5.3 Operation at Egress 9
6. Security Considerations 9
7. Management Considerations 10
8. Acknowledgements 11
9. IANA Considerations 11
10. References 11
10.1 Normative References 11
10.2 Informative References 11
Authors' Addresses 12

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 implementing service functions, 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 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]

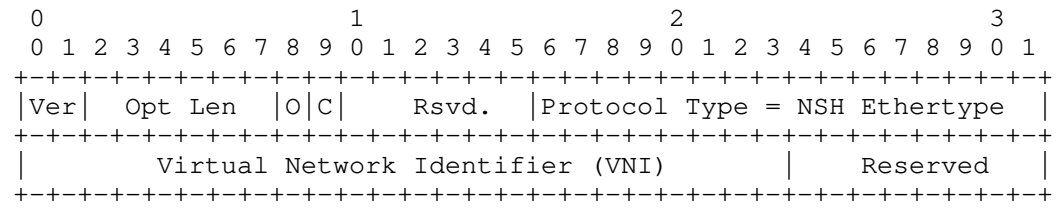
NFV: Network function virtualization.

VNF: Virtual network function

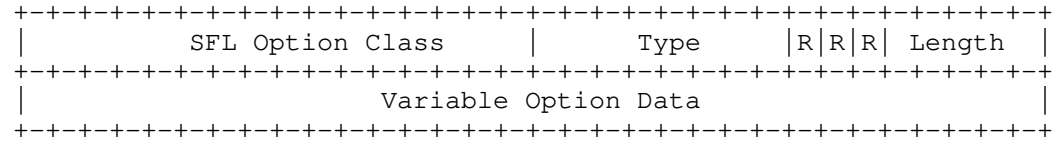
4. Geneve Option TLV(s)

4.1 Geneve Service Function List (SFL) Option TLV

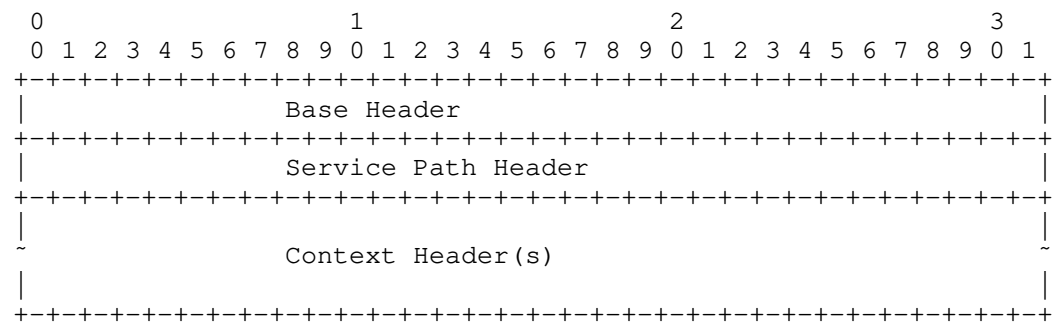
Geneve Header:



Geneve Option Header:



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



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:

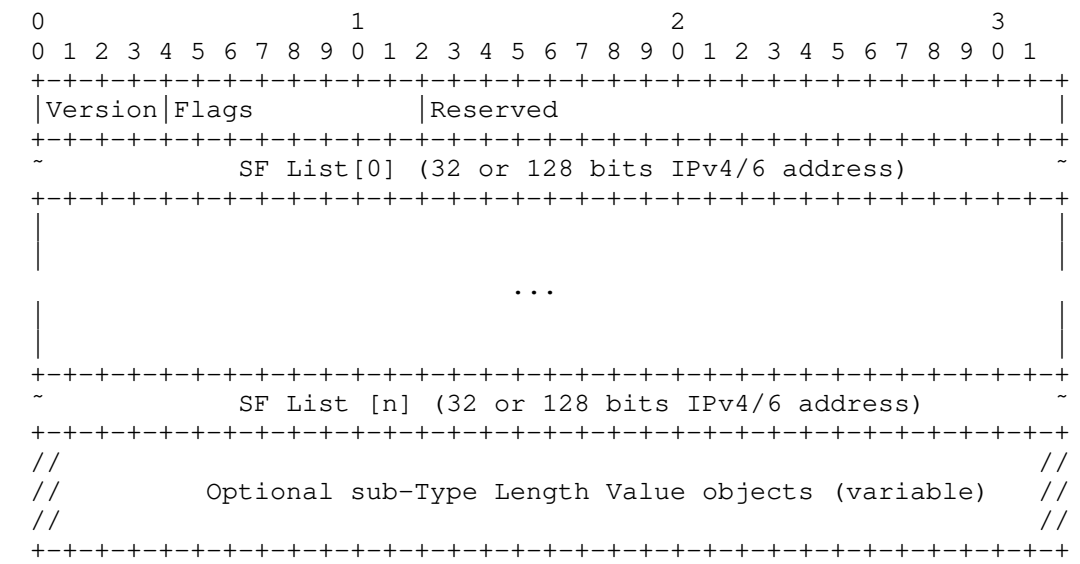


Figure 1: Service Function List (SFL) Option TLV.

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

Flags:

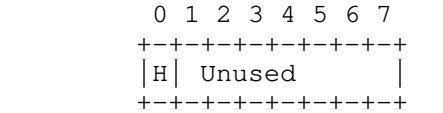


Figure 2: SFL flags

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:

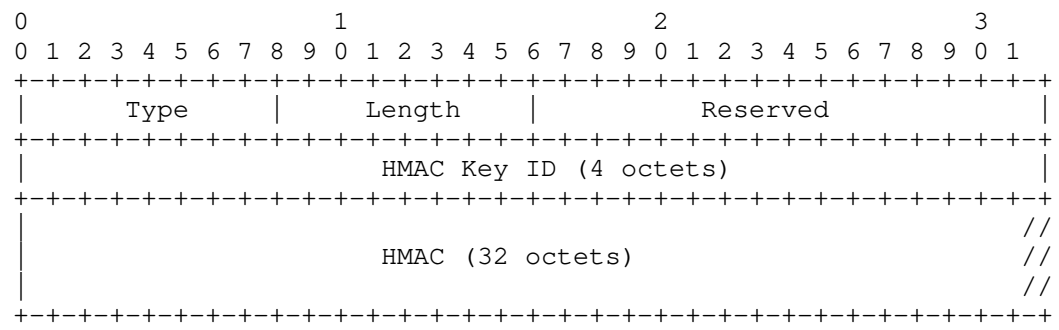


Figure 3: SFL HMAC sub-TLV.

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). 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.ietf-

nvo3-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: sboutros@vmware.com

Dharma Rajan
VMware
Email: drajan@vmware.com

Philip Kippen
VMware
Email: pkippen@vmware.com

Pierluigi Rolando
VMware
Email: prolando@vmware.com

INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros
Jerome Catrouillet
Ankur Sharma
VMware

Expires: April 30, 2018

October 27, 2017

MAC move/flush over Geneve encapsulation
draft-boutros-nvo3-mac-move-over-geneve-00

Abstract

This document specifies a mechanism to signal Media Access Control (MAC) addresses move or flush over a Network Virtualization Overlays over Layer 3 (NVO3) virtual tunnel. Such notification is useful in redundancy scenarios when a Layer 2 service that was active on a Network Virtualization Edge (NVE) fails over to a standby NVE. This notification can be used only when data plane mac learning is enabled over the NVO3 tunnels.

Status of this Memo

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

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

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

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

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

Copyright and License Notice

Table of Contents

1. Introduction 3
2. Terminology 3
3. Abbreviations 3
4.0 MAC Move/Flush Frame Format 4
5.0 Operation 5
5.1 Operation of Sender 5
5.2 Operation of Receiver 6
6. Acknowledgements 7
7. Security Considerations 7
8. IANA Considerations 7
9. References 7
9.1 Normative References 7
9.2 Informative References 7
Authors' Addresses 7

1. Introduction

In multi-homing scenarios a Layer 2 service can be multi homed to more than one Network virtualization Edge (NVE). Only one NVE can be active for a given Layer 2 service, and a standby NVE can be chosen to take over the Layer 2 service when the active NVE goes down. The mechanisms to elect which NVE will be active or standby to provide single active redundancy for a given Layer 2 service is outside the scope of this document.

When a standby NVE gets activated, Standby NVE needs to send a MAC Move/Flush message to all remote NVE(s) that spans this L2 service over the Geneve tunnels to Flush/Move all MAC learned in data plane via the old active NVE.

The MAC Move/Flush message will contain the NVE Identifier(s) of the old Active NVE and the new active NVE.

MAC Move/Flush can be used to optimize network convergence and reduce blackholes, when an active NVE hosting a logical L2 service fails over to a standby NVE.

The protocol defined in this document addresses possible loss of the MAC Move/Flush messages due to network congestion, but does not guarantee delivery.

In the event that MAC Move/Flush messages does not reach the intended target, the fallback to MAC re-learning or as a last resort aging out of MAC addresses in the absence of frames from the sources, will resume the traffic via new active NVE.

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

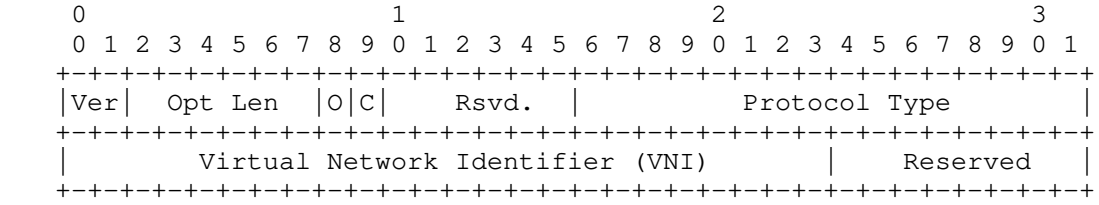
NIC Network interface card

VTEP Virtual Tunnel End Point

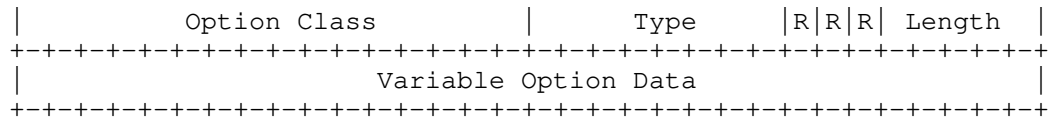
Transit device Underlay network devices between NVE(s).

4.0 MAC Move/Flush Frame Format

Geneve Header:



Geneve Option Header:



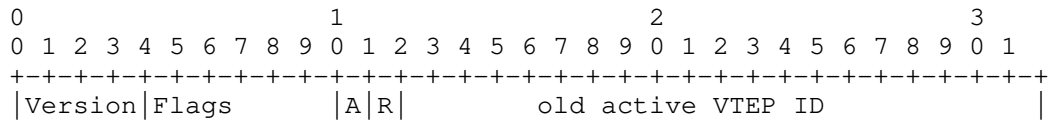
Option Class = To be assigned by IANA (TBA).

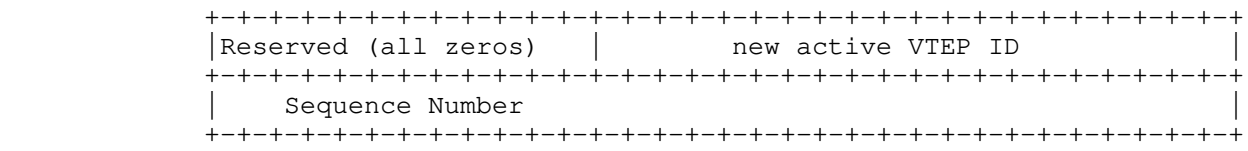
Type = TBA.

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

Length = 2 (8 bytes)

Variable option data:





Version (4 bits): Initially the Version will be 0.

A (1 bit): is set by a receiver to acknowledge receipt and processing of a MAC Flush message.

R (1 bit): is set to indicate if the sender is requesting reset of the sequence numbers. The sender sets this bit when it has no local record of previous send and expected receive sequence numbers.

Flags(6): Reserved and should be set to 0.

VTEP ID (20 bits): Identifies an NVE, for old and new active NVE(s), the new active NVE identifier will be set in case of a MAC move, and will be 0 for a MAC flush.

Sequence Number (32) bits: For overflow detection a sequence number that exceeds 2,147,483,647 (0x7FFFFFFF) is considered an overflow and reset to 1.

5.0 Operation

This section describes how the initial MAC Flush/Move Messages are sent and retransmitted, as well as how the messages are processed and retransmitted messages are identified. The mechanisms described are very similar to the one defined in [RFC 7769].

5.1 Operation of Sender

At the NVE , each L2 logical switch identified by a VNI is associated with a counter to keep track of the sequence number of the transmitted MAC Move/Flush messages. Whenever a node sends a MAC Move/Flush message, it increments the transmitted sequence-number counter and includes the new sequence number in the message.

The transmit sequence number is initialized to 1 at the onset, after the wrap and after the sequence number reset request receipt. Hence the transmit sequence number is set to 2 in the first MAC Flush/Move message sent after the sequence number is initialized to 1.

The sender expects an ACK from the receiver within a retransmit time interval, which can be either a default (1 second) or a configured value. If the ACK is not received within the Retransmit time, the sender retransmits the message with the same sequence number as the original message. The retransmission MUST cease when an ACK is received. In order to avoid continuous re-transmissions in the absence of acknowledgements, the sender MUST cease retransmission after a small number of transmissions, two retries is RECOMMENDED. Alternatively, an increasing backoff delay with a larger number of retries MAY be implemented to improve scaling issues.

During the period of retransmission, if a need to send a new MAC Move/Flush message with updated sequence number arises, then retransmission of the older unacknowledged Move/Flush message MUST be suspended and retransmit time for the new sequence number MUST be initiated. In essence, a sender engages in retransmission logic only for the most recently sent Move/Flush message for a given L2 Logical Switch identified by a VNI.

In the event that the L2 logical switch is deleted and re-added or the VTEP node is restarted with new configuration, the NVE may lose information about the previously sent sequence number. This becomes problematic for the remote peer as it will continue to ignore the received MAC Move/Flush messages with lower sequence numbers. In such cases, it is desirable to reset the sequence numbers, the reset R-bit is set in the first MAC Flush to notify the remote peer to reset the send and receive sequence numbers. The R-bit must be cleared in subsequent MAC Move/Flush messages after the acknowledgement is received.

5.2 Operation of Receiver

Each L2 logical switch identified by a VNI is associated with a receive sequence number per remote NVE to keep track of the expected sequence number of the MAC Move/Flush message.

Whenever a MAC Move/Flush message is received, and if the sequence number on the message is greater than the value in the receive sequence number of this remote NVE, the MAC addresses learned from the NVE associated with the NVE identifier in the message are flushed or moved to be associated with the new active NVE identifier, and the receive sequence number of the remote NVE is updated with the received sequence number. The receiver sends an ACK with the same sequence number in the received message.

If the sequence number in the received message is smaller than or equal to the value in the receive sequence number per remote NVE, the

MAC Move/Flush is not processed. However, an ACK with the received sequence number MUST be sent as a response to stop the sender retransmission.

A MAC Move/Flush message with the R-bit set MUST be processed by resetting the receive sequence number of the remote NVE, and Moving/flushing the MACs as described above. The acknowledgement is sent with the R-bit cleared.

6. Acknowledgements

7. Security Considerations

This document does not introduce any additional security constraints.

8. IANA Considerations

IANA is requested to assign a new option class from the "Geneve Option Class" registry for the Geneve MAC Move/Flush option.

Option Class	Description
XXXX	Geneve MAC Move/Flush

9. References

9.1 Normative References

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

9.2 Informative References

[Geneve] "Generic Network Virtualization Encapsulation", [I-D.ietf-nvo3-geneve] [RFC 7769] "MAC Address Withdrawal over Static PW", [RFC 7769]

Authors' Addresses

Sami Boutros
VMware
Email: sboutros@vmware.com

Jerome Catrouillet
VMware, Inc.

Email: jcatrouillet@vmware.com

Ankur Sharma
VMware, Inc.
Email: ankursharma@vmware.com

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

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

October 30, 2017

Geneve encapsulation for In-situ OAM Data
draft-brockners-nvo3-ioam-geneve-00

Abstract

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

Status of This Memo

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

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

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

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

Table of Contents

1. Introduction 2
2. Conventions 3
2.1. Requirement Language 3
2.2. Abbreviations 3
3. IOAM Data Field Encapsulation in Geneve 3
3.1. IOAM Trace Data in Geneve 3
3.2. IOAM POT Data in Geneve 7
3.3. IOAM Edge-to-Edge Data in Geneve 8
4. Discussion of the encapsulation approach 9
5. IANA Considerations 10
6. Security Considerations 10
7. Acknowledgements 11
8. References 11
8.1. Normative References 11
8.2. Informative References 12
Authors' Addresses 12

1. Introduction

In-situ OAM (IOAM) records OAM information within the packet while the packet traverses a particular network domain. The term "in-situ" refers to the fact that the IOAM data fields are added to the data packets rather than is being sent within packets specifically dedicated to OAM. This document defines how IOAM data fields are transported as part of the Geneve [I-D.ietf-nvo3-geneve] encapsulation. The IOAM data fields are defined in [I-D.ietf-ippm-ioam-data].

2. Conventions

2.1. Requirement Language

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

2.2. Abbreviations

Abbreviations used in this document:

IOAM: In-situ Operations, Administration, and Maintenance

MTU: Maximum Transmit Unit

OAM: Operations, Administration, and Maintenance

POT: Proof of Transit

Geneve: Generic Network Virtualization Encapsulation

3. IOAM Data Field Encapsulation in Geneve

For encapsulating IOAM data fields into Geneve [I-D.ietf-nvo3-geneve] the different IOAM data fields are included in the Geneve header using tunnel options. IOAM data fields use a tunnel option class which includes the different types of IOAM data, including trace data, proof-of-transit data, and edge-to-edge data. In an administrative domain where IOAM is used, insertion of the IOAM tunnel option(s) in Geneve is enabled at the Geneve tunnel endpoints which also serve as IOAM encapsulating/decapsulating nodes by means of configuration. The Geneve header is defined in [I-D.ietf-nvo3-geneve]. IOAM specific fields for Geneve are defined in this document.

3.1. IOAM Trace Data in Geneve

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

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

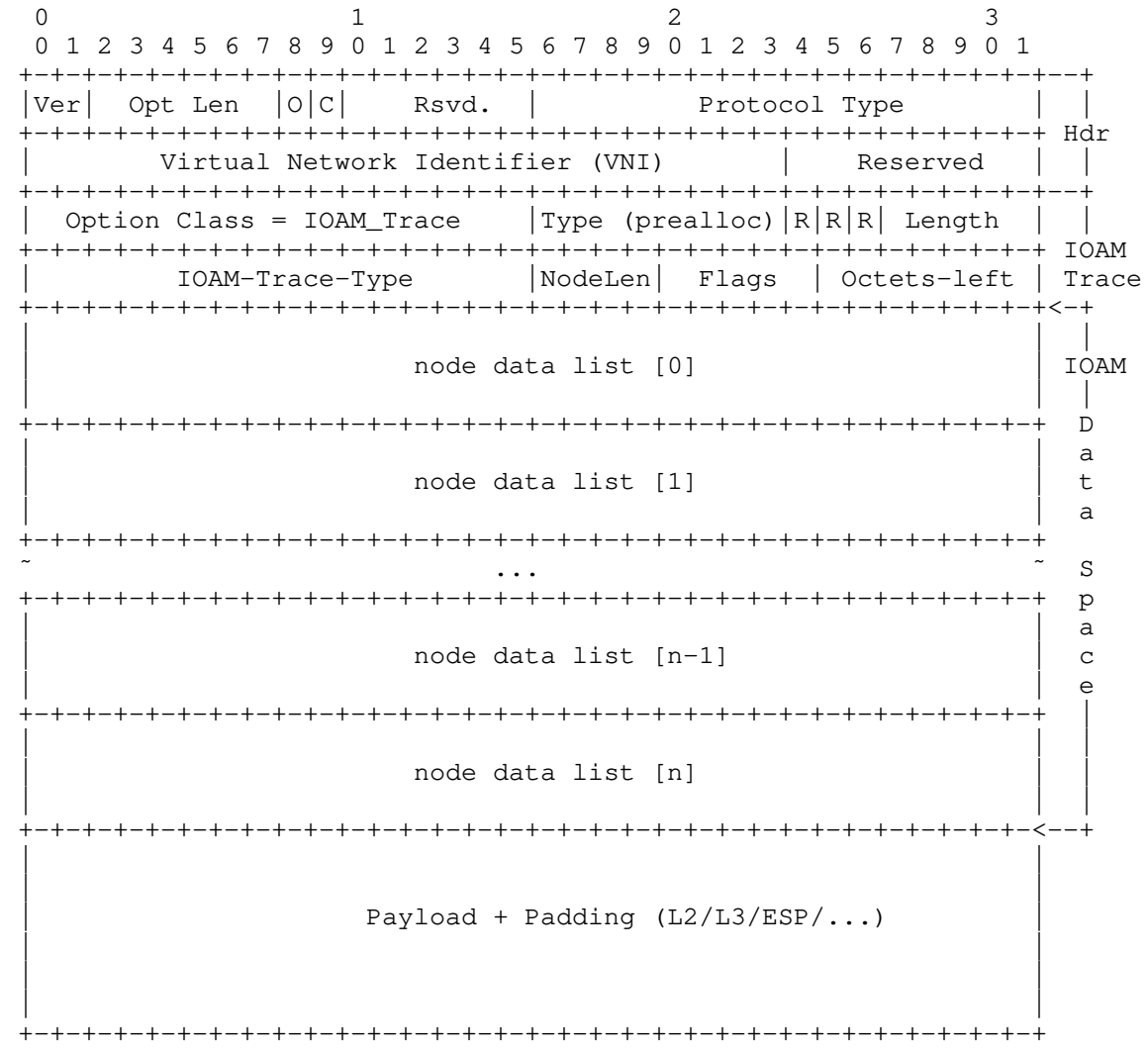
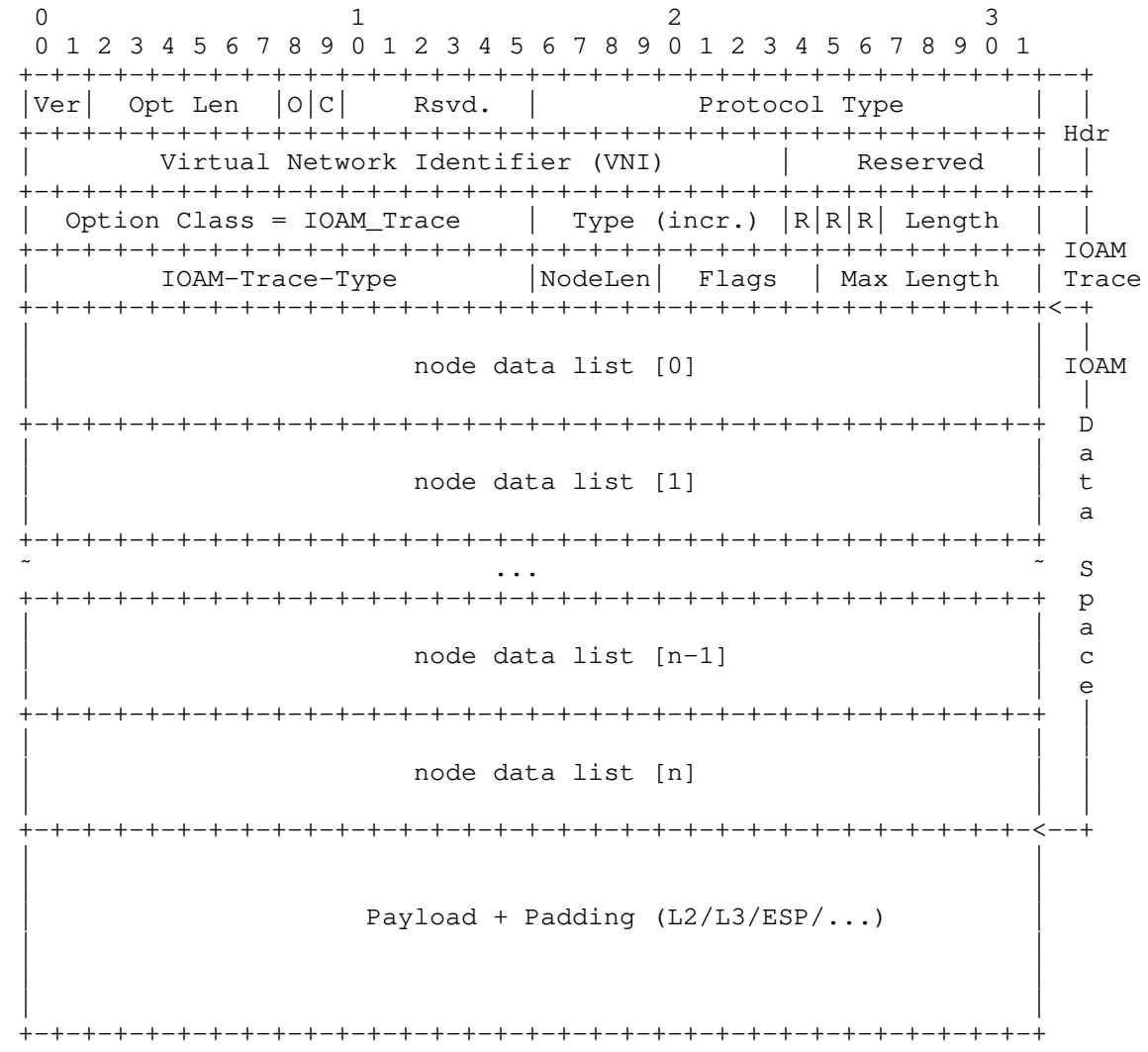


Figure 1: IOAM Pre-allocated Trace Option Format as a Geneve Tunnel Option



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

Figure 2: IOAM Incremental Trace Option Format as a Geneve Tunnel Option

The IOAM Trace header consists of 8 octets, as illustrated in Figure 1 and Figure 2. The first 4 octets are the Geneve Tunnel Option header [I-D.ietf-nvo3-geneve]. The next 4 octets are the trace option header; its format is defined in [I-D.ietf-ippm-ioam-data], and is described here for the sake of clarity.

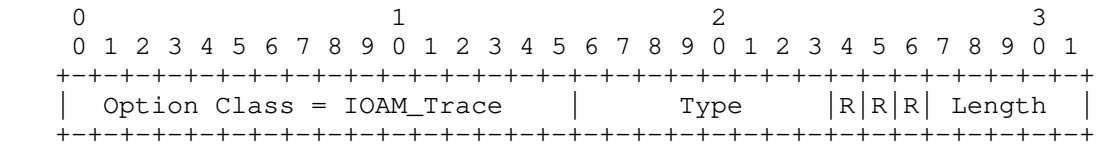


Figure 3: Geneve Tunnel Option for IOAM

The fields of the Geneve tunnel option are as follows:

Option Class: 16-bit unsigned integer that determines the IOAM option class. The value is from the IANA registry setup for Geneve option classes as defined in [I-D.ietf-nvo3-geneve].

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

R (3 bits): Option control flags reserved for future use. MUST be zero on transmission and ignored on receipt.

Length: 5-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

The fields of the trace option header [I-D.ietf-ippm-ioam-data] are as follows:

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

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

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

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

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

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

3.2. IOAM POT Data in Geneve

IOAM proof of transit (POT, see also [I-D.brockners-proof-of-transit]) offers a means to verify that a packet has traversed a defined set of nodes. IOAM POT data fields are encapsulated in Geneve as follows:

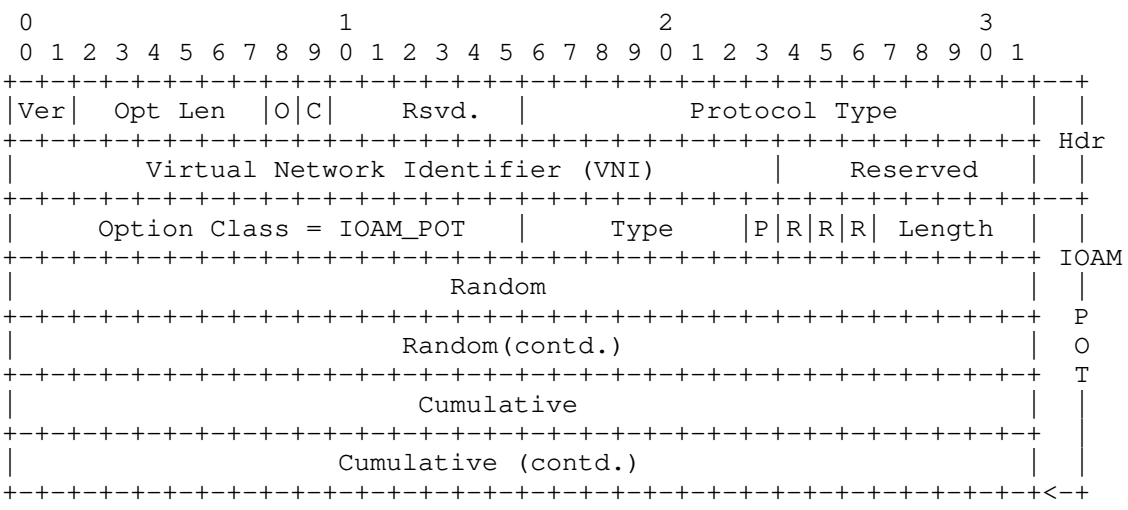


Figure 4: IOAM POT Header Following using a Geneve Tunnel Option

The first 4 octets of the IOAM POT are the Geneve tunnel option header (Figure 5), which includes the following fields:

Option Class: 16-bit unsigned integer that determines the IOAM_POT option class.The value is from the IANA registry setup for Geneve option classes as defined in [I-D.ietf-nvo3-geneve].

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

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

R (3 bits): Option control flags reserved for future use. MUST be zero on transmission and ignored on receipt.

Length: 5-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

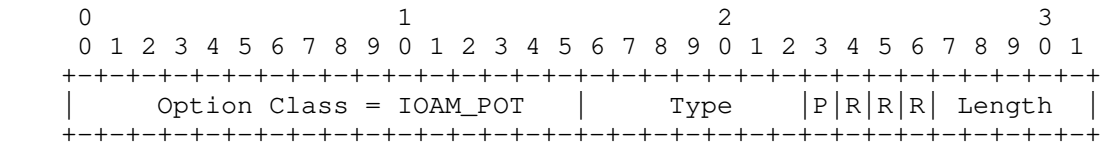


Figure 5: Geneve Tunnel Option for IOAM POT

The rest of the fields in the POT option [I-D.ietf-ippm-ioam-data] are as follows:

Random: 64-bit Per-packet random number.

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

3.3. IOAM Edge-to-Edge Data in Geneve

The IOAM edge-to-edge option is to carry data that is added by the IOAM encapsulating node and interpreted by the IOAM decapsulating node. IOAM specific fields to encapsulate IOAM Edge-to-Edge data fields are defined as follows:

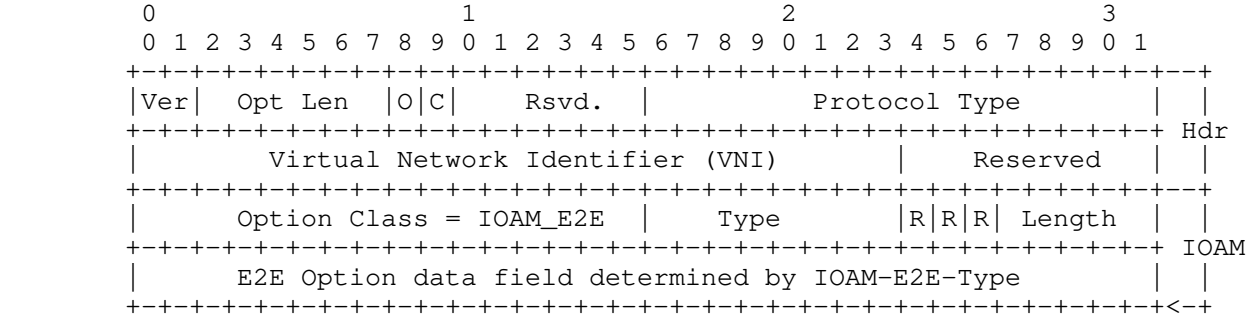


Figure 6: IOAM Edge-to-Edge using a Geneve Tunnel Option

The first 4 octets of the IOAM E2E option are the Geneve tunnel option header (Figure 5), which includes the following fields:

Option Class 16-bit unsigned integer that determines the IOAM_E2E option class.The value is from the IANA registry setup for Geneve option classes as defined in [I-D.ietf-nvo3-geneve].

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

R (3 bits): Option control flags reserved for future use. MUST be zero on transmission and ignored on receipt.

Length: 5-bit unsigned integer. Length of the IOAM HDR in 4-octet units.

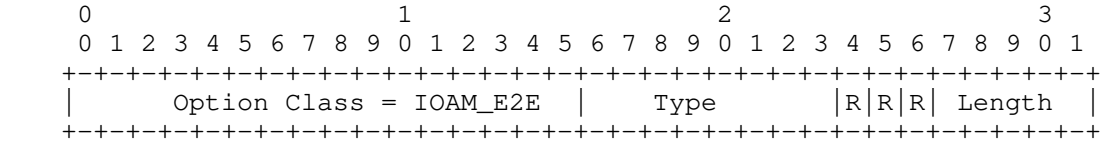


Figure 7: Geneve Tunnel Option for IOAM E2E

The rest of the E2E option [I-D.ietf-ippm-ioam-data] consists of:

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

4. Discussion of the encapsulation approach

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

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

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

Total length of the IOAM data fields: The total length of IOAM data can grow quite large in case multiple different IOAM data fields are used and large path-lengths need to be considered. If for example an operator would consider using the IOAM trace option

and capture node-id, app_data, egress/ingress interface-id, timestamp seconds, timestamps nanoseconds at every hop, then a total of 20 octets would be added to the packet at every hop. In case this particular deployment would have a maximum path length of 15 hops in the IOAM domain, then a maximum of 300 octets of IOAM data were to be encapsulated in the packet.

Concerns with the current encapsulation approach:

Hardware support: Using Geneve tunnel options to encapsulate IOAM data fields leads to a nested TLV structure. Each IOAM data field option (trace, proof-of-transit, and edge-to-edge) represents a type, with the different IOAM data fields being TLVs within this the particular option type. Nested TLVs require iterative look-ups, a fact that creates potential challenges for implementations in hardware. It would be desirable to offer a way to encapsulate IOAM in a way that keeps TLV nesting to a minimum.

Length: Geneve tunnel option length is a 5-bit field in the current specification [I-D.ietf-nvo3-geneve] resulting in a maximum option length of 128 (2^5 x 4) octets which constrains the use of IOAM to either small domains or a few IOAM data fields only. Support for large domains with a variety of IOAM data fields would be desirable.

5. IANA Considerations

IANA is requested to allocate a Geneve "option class" numbers for the following IOAM types:

Option Class	Description	Reference
x	IOAM_Trace	This document
y	IOAM_POT	This document
z	IOAM_E2E	This document

6. Security Considerations

The security considerations of Geneve are discussed in [I-D.ietf-nvo3-geneve], and the security considerations of IOAM in general are discussed in [I-D.ietf-ippm-ioam-data].

IOAM is considered a "per domain" feature, where one or several operators decide on leveraging and configuring IOAM according to their needs. Still, operators need to properly secure the IOAM

domain to avoid malicious configuration and use, which could include injecting malicious IOAM packets into a domain.

7. Acknowledgements

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

8. References

8.1. Normative References

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

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

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

[I-D.ietf-nvo3-geneve] Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-05 (work in progress), September 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

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

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

8.2. Informative References

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

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

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

Authors' Addresses

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

Email: fbrockne@cisco.com

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

Email: shwethab@cisco.com

Vengada Prasad Govindan
Cisco Systems, Inc.

Email: venggovi@cisco.com

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

Email: cpignata@cisco.com

Hannes Gredler
RtBrick Inc.

Email: hannes@rtbrick.com

John Leddy
Comcast

Email: John_Leddy@cable.comcast.com

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

Email: stephen.youell@jpmorgan.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam 20692
Israel

Email: talmi@marvell.com

David Mozes
Mellanox Technologies Ltd.

Email: davidm@mellanox.com

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

Email: petr@fb.com

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

NVO3 Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2019

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

October 23, 2018

Performance Measurement (PM) with Alternate Marking in Network Virtualization Overlays (NVO3)
draft-fmm-nvo3-pm-alt-mark-03

Abstract

This document describes how the alternate marking method can be used for performance measurement method in a Network Virtualization Overlays (NVO3) Domain. The description aims to be general for NVO3 encapsulations, but is focused to Geneve, recommended by the NVO3 design team [I-D.ietf-nvo3-encap].

Status of This Memo

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

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

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

This Internet-Draft will expire on April 26, 2019.

Table of Contents

1. Introduction 2
2. Conventions used in this document 2
2.1. Terminology 3
2.2. Requirements Language 3
3. OAM Performance Measurement in a NVO3 Domain 3
4. The Mark Field in the NVO3 Header 5
5. Theory of Operation 6
5.1. Single Mark Enabled Measurement 6
5.2. Double Mark Enabled Measurement 7
5.3. Multiplexed Mark Enabled Measurement 8
6. Multipoint Measurement Considerations 8
7. The Mark Field in Geneve 8
8. IANA Considerations 9
8.1. Mark Field in Geneve Header 9
9. Security Considerations 9
10. Acknowledgement 9
11. References 9
11.1. Normative References 9
11.2. Informative References 10
Authors' Addresses 11

1. Introduction

[RFC7365] provides a framework for Data Center (DC) Network Virtualization over Layer 3 (NVO3) tunnels. It is intended to aid in standardizing protocols and mechanisms to support large-scale network virtualization for data centers.

[RFC8321] describes a performance measurement method, which can be used to measure packet loss, latency, and jitter on live traffic. Since this method is based on marking consecutive batches of packets the method often referred to as the Alternate Marking Method (AMM).

This document defines how the alternate marking method can be used to measure packet loss and delay metrics of an NVO3 Domain.

2. Conventions used in this document

2.1. Terminology

AMM: Alternate Marking Method

OAM: Operations, Administration and Maintenance

NVO3: Network Virtualization Overlays

NVE: Network Virtualization Edge

VNI: Virtual Network Instance

DC: Data Center

NVA: Network Virtualization Authority

Geneve: Generic Network Virtualization Encapsulation

VXLAN: Virtual Extensible LAN

GUE: Generic UDP Encapsulation

2.2. Requirements Language

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

3. OAM Performance Measurement in a NVO3 Domain

Figure 1 shows the generic reference model for a DC network virtualization over an L3 infrastructure while Figure 2 shows the generic reference model for the Network Virtualization Edge (NVE). Both Figures are taken from [RFC7365] and [RFC8014].

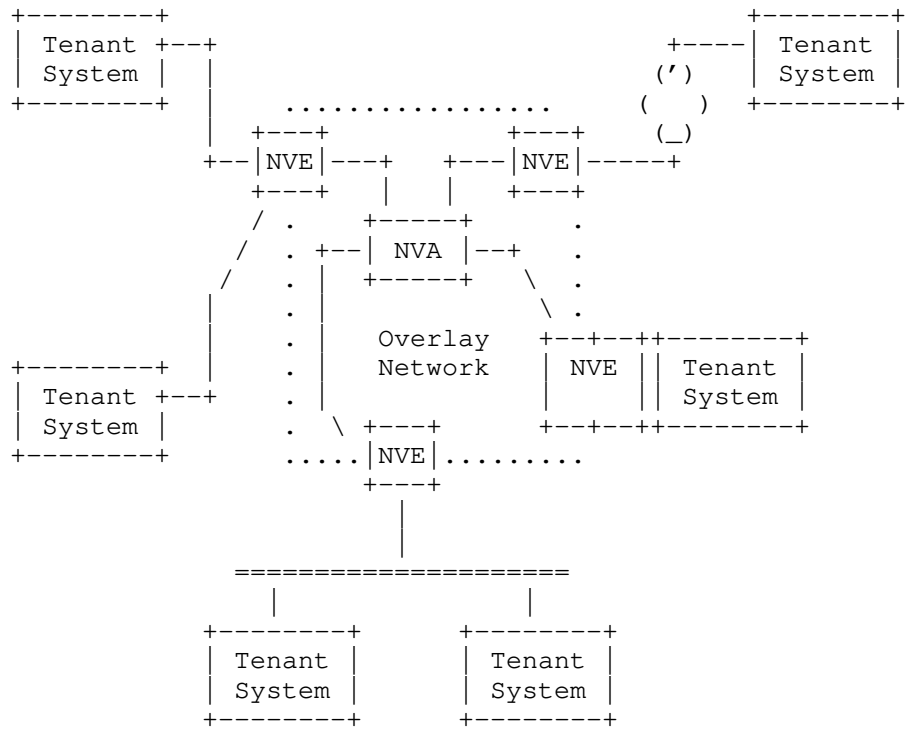


Figure 1: Generic Reference Model for DC Network Virtualization Overlays (RFC7365)

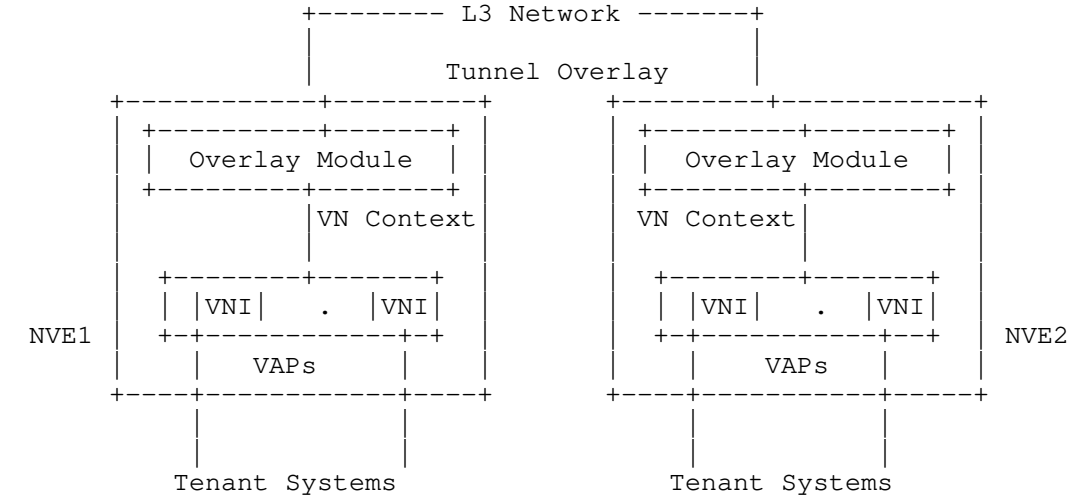


Figure 2: Generic NVE Reference Model (RFC7365)

L3 networks provide transport for an emulated Layer 2 created by NVE devices. The connectivity between the NVE devices is achieved with unicast and multicast tunneling methods. Then, the NVE devices present an emulated Layer 2 network to the Tenant End Systems at a Virtual Network Instance (VNI) through Virtual Access Points (VAPs). The NVE devices map Layer 2 unicast to Layer 3 unicast point-to-point tunnels and may either map Layer 2 multicast to Layer 3 multicast tunnels or may replicate packets onto multiple Layer 3 unicast tunnels.

The emulated Layer 2 network is provided by the NVE devices to which the Tenant End Systems are connected. This network of NVE can be operated by a single service provider or can span across multiple administrative domains. Likewise, the L3 Overlay Network can be operated by a single service provider or span across multiple administrative domains.

Each of the layers is responsible for its own OAM. Complex OAM relationships exist as a result of the hierarchical layering, but this is out of scope here.

When we refer to an OAM domain considered in this document we refer to a set of NVEs and the tunnels which interconnect them.

It is commonly agreed that NVO3 OAM Performance Management supports measurements (packet loss, latency, and jitter) per VNI between two NVE devices that support the same VNI within a given NVO3 domain.

4. The Mark Field in the NVO3 Header

This document defines a two-bit long field, referred to as Mark field (M), as part of the NVO3 Header and designated for the alternate marking performance measurement method [RFC8321]. The Mark field MUST NOT be used in defining forwarding and/or quality of service treatment of an NVO3 packet. The Mark field MUST be used only for the performance measurement of data traffic in the NVO3 layer. Since the field does not affect forwarding and/or quality of service treatment of packets, the alternate marking method in the NVO3 layer can be viewed as nearly-passive performance measurement method.

Figure 3 displays the format of the Mark field.

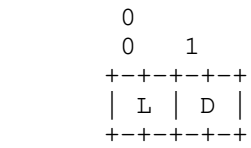


Figure 3: Mark field (M) format

where:

- L - Loss bit;
- D - Delay bit.

5. Theory of Operation

The marking method can be used in NVO3. For example, one can consider the NVO3 reference model presented in Figure 1. AMM can be applied at either ingress or egress NVE to detect performance degradation defect and localize it efficiently.

Using AMM, NVE1 creates distinct sub-flows. Each sub-flow consists of consecutive blocks that are unambiguously recognizable by a monitoring point at any component of the NVO3, e.g. NVE2 or NVE3, and can be measured to calculate packet loss and/or packet delay metrics.

Every NVO3 Header [I-D.ietf-nvo3-geneve], [I-D.ietf-nvo3-vxlan-gpe] and [I-D.ietf-nvo3-gue] can be considered for the application of AMM.

5.1. Single Mark Enabled Measurement

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

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

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

- First/Last Packet Delay calculation: whenever the marking, i.e. the value of L flag, changes a component of the NVO3 can store the timestamp of the first/last packet of the block. The timestamp can be compared with the timestamp of the packet that arrived in the same order through a monitoring point at a downstream component of the NVO3 to compute packet delay. Because timestamps collected based on order of arrival this method is sensitive to packet loss and re-ordering of packets

- Average Packet Delay calculation: an average delay is calculated by considering the average arrival time of the packets within a single block. A component of the NVO3 may collect timestamps for each packet received within a single block. Average of the timestamp is the sum of all the timestamps divided by the total number of packets received. Then the difference between averages calculated at two monitoring points is the average packet delay on that segment. This method is robust to out of order packets and also to packet loss (only a small error is introduced). This method only provides a single metric for the duration of the block and it doesn't give the minimum and maximum delay values. This limitation could be overcome by reducing the duration of the block by means of a highly optimized implementation of the method.

5.2. Double Mark Enabled Measurement

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

The first marking (L flag alternation) is needed for packet loss and also for average delay measurement. The second marking (D flag is put to one) creates a new set of marked packets that are fully identified over NVO3, so that a component can store the timestamps of these packets; these timestamps can be compared with the timestamps of the same packets on another component of the NVO3 to compute packet delay values for each packet. The number of measurements can be easily increased by changing the frequency of the second marking. But the frequency of the second marking must be not too high in order to avoid out of order issues. This method is suitable to have not only the average delay but also the minimum and maximum delay values and, in wider terms, to know more about the statistic distribution of delay values.

5.3. Multiplexed Mark Enabled Measurement

There is also a scheme that provides the benefits of Double Mark method, but uses only one bit like Single Mark. This methodology is described in [I-D.mizrahi-ippm-compact-alternate-marking]. The concept is that in the middle of each block of packets with a certain value of the L flag, a single packet has the L flag inverted. So, by examining the stream, the packets with the inverted bit can be easily identified and employed for delay measurement. This Alternate Marking variation is advantageous because it requires only one bit from each packet, and such bits are always in short supply.

6. Multipoint Measurement Considerations

The Multipoint characteristics of the traffic within a given NVO3 Domain could be considered a valuable Use Case of [I-D.fioccola-ippm-multipoint-alt-mark].

7. The Mark Field in Geneve

[I-D.ietf-nvo3-geneve] defines the format of the Geneve Header.

The design team recommendations in [I-D.ietf-nvo3-encap] section 7 concluded that Geneve is most suitable as a starting point for the proposed standard for network virtualization.

In addition, the design team recommended addressing requirements for OAM considerations for alternate marking and for performance measurements that need 2 bits in the header. This document clarifies the need for the current OAM bit in the Geneve Header.

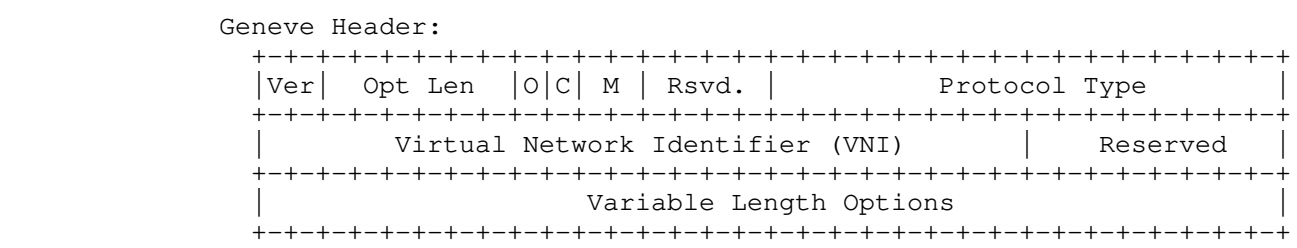


Figure 4: Geneve Header

This document defines a two-bit long field, referred to as the Mark field (M in Figure 4, as part of Geneve and designated for the alternate marking performance measurement method [RFC8321]. The Mark field MUST NOT be used in defining forwarding and/or quality of service treatment of a NVO3 packet. The Mark field MUST be used only for the performance measurement of data traffic in the NVO3 layer.

Since the field does not affect forwarding and/or quality of service treatment of packets, the alternate marking method in the NVO3 layer can be viewed as nearly-passive performance measurement method.

8. IANA Considerations

8.1. Mark Field in Geneve Header

This document requests IANA to allocate Mark field as two bits-long field from Geneve Header Reserved Bits [I-D.ietf-nvo3-geneve].

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

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

Table 1: Mark field of Geneve

9. Security Considerations

This document lists the OAM requirement for the NVO3 domain and does not raise any security concerns or issues in addition to ones common to networking and NVO3.

10. Acknowledgement

The authors would like to thank Dale R. Worley for the contribution.

11. References

11.1. Normative References

[I-D.ietf-nvo3-encap]
Boutros, S., "NVO3 Encapsulation Considerations", draft-ietf-nvo3-encap-02 (work in progress), September 2018.

[I-D.ietf-nvo3-geneve]
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-08 (work in progress), October 2018.

[I-D.ietf-nvo3-gue]
Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", draft-ietf-nvo3-gue-05 (work in progress), October 2016.

[I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-06 (work in progress), April 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

11.2. Informative References

[I-D.fioccola-ippm-multipoint-alt-mark]
Fioccola, G., Cociglio, M., Sapio, A., and R. Sisto, "Multipoint Alternate Marking method for passive and hybrid performance monitoring", draft-fioccola-ippm-multipoint-alt-mark-04 (work in progress), June 2018.

[I-D.mizrahi-ippm-compact-alternate-marking]
Mizrahi, T., Arad, C., Fioccola, G., Cociglio, M., Chen, M., Zheng, L., and G. Mirsky, "Compact Alternate Marking Methods for Passive and Hybrid Performance Monitoring", draft-mizrahi-ippm-compact-alternate-marking-03 (work in progress), October 2018.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <https://www.rfc-editor.org/info/rfc7365>.

[RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <https://www.rfc-editor.org/info/rfc8014>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <https://www.rfc-editor.org/info/rfc8321>.

Authors' Addresses

Giuseppe Fioccola
Huawei Technologies
Riesstrasse, 25
Munich 80992
Germany

Email: giuseppe.fioccola@huawei.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Tal Mizrahi
Huawei Network.IO Innovation Lab
Israel

Email: tal.mizrahi.phd@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 10, 2019

J. Gross, Ed.
I. Ganga, Ed.
Intel
T. Sridhar, Ed.
VMware
October 07, 2018

Geneve: Generic Network Virtualization Encapsulation
draft-ietf-nvo3-geneve-08

Abstract

Network virtualization involves the cooperation of devices with a wide variety of capabilities such as software and hardware tunnel endpoints, transit fabrics, and centralized control clusters. As a result of their role in tying together different elements in the system, the requirements on tunnels are influenced by all of these components. Flexibility is therefore the most important aspect of a tunnel protocol if it is to keep pace with the evolution of the system. This draft describes Geneve, a protocol designed to recognize and accommodate these changing capabilities and needs.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 10, 2019.

Table of Contents

1. Introduction 3
1.1. Requirements Language 4
1.2. Terminology 4
2. Design Requirements 5
2.1. Control Plane Independence 6
2.2. Data Plane Extensibility 7
2.2.1. Efficient Implementation 7
2.3. Use of Standard IP Fabrics 8
3. Geneve Encapsulation Details 9
3.1. Geneve Packet Format Over IPv4 9
3.2. Geneve Packet Format Over IPv6 10
3.3. UDP Header 12
3.4. Tunnel Header Fields 13
3.5. Tunnel Options 14
3.5.1. Options Processing 16
4. Implementation and Deployment Considerations 17
4.1. Encapsulation of Geneve in IP 17
4.1.1. IP Fragmentation 17
4.1.2. DSCP and ECN 17
4.1.3. Broadcast and Multicast 18
4.1.4. Unidirectional Tunnels 18
4.2. Constraints on Protocol Features 19
4.2.1. Constraints on Options 19
4.3. NIC Offloads 19
4.4. Inner VLAN Handling 20
5. Interoperability Issues 20
6. Security Considerations 21
6.1. Data Confidentiality 22
6.1.1. Inter-data center traffic 22
6.2. Data Integrity 22
6.3. Authentication of NVE peers 23
6.4. Multicast/Broadcast 23
6.5. Control plane communications 24
7. IANA Considerations 24
8. Contributors 25
9. Acknowledgements 26
10. References 26
10.1. Normative References 26

10.2. Informative References 27
Authors' Addresses 29

1. Introduction

Networking has long featured a variety of tunneling, tagging, and other encapsulation mechanisms. However, the advent of network virtualization has caused a surge of renewed interest and a corresponding increase in the introduction of new protocols. The large number of protocols in this space, ranging all the way from VLANs [IEEE.802.1Q_2014] and MPLS [RFC3031] through the more recent VXLAN [RFC7348], NVGRE [RFC7637], often leads to questions about the need for new encapsulation formats and what it is about network virtualization in particular that leads to their proliferation.

While many encapsulation protocols seek to simply partition the underlay network or bridge between two domains, network virtualization views the transit network as providing connectivity between multiple components of a distributed system. In many ways this system is similar to a chassis switch with the IP underlay network playing the role of the backplane and tunnel endpoints on the edge as line cards. When viewed in this light, the requirements placed on the tunnel protocol are significantly different in terms of the quantity of metadata necessary and the role of transit nodes.

Current work such as VL2 [VL2] and the NVO3 working group [I-D.ietf-nvo3-dataplane-requirements] have described some of the properties that the data plane must have to support network virtualization. However, one additional defining requirement is the need to carry system state along with the packet data. The use of some metadata is certainly not a foreign concept - nearly all protocols used for virtualization have at least 24 bits of identifier space as a way to partition between tenants. This is often described as overcoming the limits of 12-bit VLANs, and when seen in that context, or any context where it is a true tenant identifier, 16 million possible entries is a large number. However, the reality is that the metadata is not exclusively used to identify tenants and encoding other information quickly starts to crowd the space. In fact, when compared to the tags used to exchange metadata between line cards on a chassis switch, 24-bit identifiers start to look quite small. There are nearly endless uses for this metadata, ranging from storing input ports for simple security policies to service based context for interposing advanced middleboxes.

Existing tunnel protocols have each attempted to solve different aspects of these new requirements, only to be quickly rendered out of date by changing control plane implementations and advancements. Furthermore, software and hardware components and controllers all

have different advantages and rates of evolution - a fact that should be viewed as a benefit, not a liability or limitation. This draft describes Geneve, a protocol which seeks to avoid these problems by providing a framework for tunneling for network virtualization rather than being prescriptive about the entire system.

1.1. Requirements Language

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

1.2. Terminology

The NVO3 framework [RFC7365] defines many of the concepts commonly used in network virtualization. In addition, the following terms are specifically meaningful in this document:

Checksum offload. An optimization implemented by many NICs which enables computation and verification of upper layer protocol checksums in hardware on transmit and receive, respectively. This typically includes IP and TCP/UDP checksums which would otherwise be computed by the protocol stack in software.

Clos network. A technique for composing network fabrics larger than a single switch while maintaining non-blocking bandwidth across connection points. ECMP is used to divide traffic across the multiple links and switches that constitute the fabric. Sometimes termed "leaf and spine" or "fat tree" topologies.

ECMP. Equal Cost Multipath. A routing mechanism for selecting from among multiple best next hop paths by hashing packet headers in order to better utilize network bandwidth while avoiding reordering a single stream.

Geneve. Generic Network Virtualization Encapsulation. The tunnel protocol described in this draft.

LRO. Large Receive Offload. The receive-side equivalent function of LSO, in which multiple protocol segments (primarily TCP) are coalesced into larger data units.

NIC. Network Interface Card. A NIC could be part of a tunnel endpoint or transit device and can either process Geneve packets or aid in the processing of Geneve packets.

OAM. Operations, Administration, and Management. A suite of tools used to monitor and troubleshoot network problems.

Transit device. A forwarding element along the path of the tunnel making up part of the Underlay Network. A transit device MAY be capable of understanding the Geneve packet format but does not originate or terminate Geneve packets.

LSO. Large Segmentation Offload. A function provided by many commercial NICs that allows data units larger than the MTU to be passed to the NIC to improve performance, the NIC being responsible for creating smaller segments of size less than or equal to the MTU with correct protocol headers. When referring specifically to TCP/IP, this feature is often known as TSO (TCP Segmentation Offload).

Tunnel endpoint. A component performing encapsulation and decapsulation of packets, such as Ethernet frames or IP datagrams, in Geneve headers. As the ultimate consumer of any tunnel metadata, endpoints have the highest level of requirements for parsing and interpreting tunnel headers. Tunnel endpoints may consist of either software or hardware implementations or a combination of the two. Endpoints are frequently a component of an NVE but may also be found in middleboxes or other elements making up an NVO3 Network.

VM. Virtual Machine.

2. Design Requirements

Geneve is designed to support network virtualization use cases, where tunnels are typically established to act as a backplane between the virtual switches residing in hypervisors, physical switches, or middleboxes or other appliances. An arbitrary IP network can be used as an underlay although Clos networks composed using ECMP links are a common choice to provide consistent bisectional bandwidth across all connection points. Figure 1 shows an example of a hypervisor, top of rack switch for connectivity to physical servers, and a WAN uplink connected using Geneve tunnels over a simplified Clos network. These tunnels are used to encapsulate and forward frames from the attached components such as VMs or physical links.

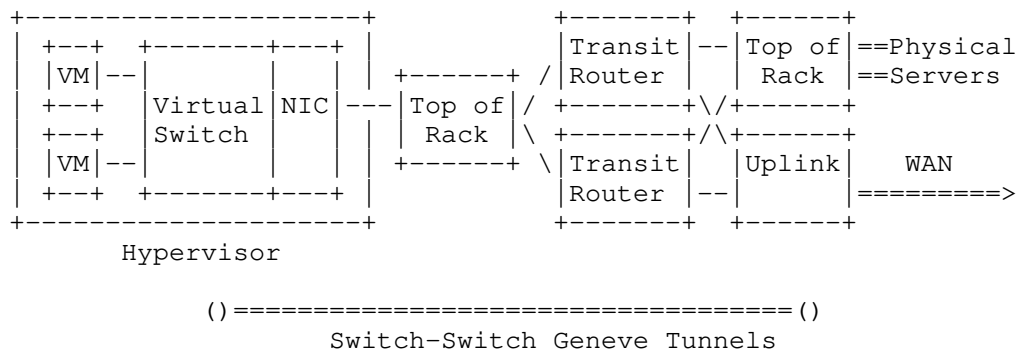


Figure 1: Sample Geneve Deployment

To support the needs of network virtualization, the tunnel protocol should be able to take advantage of the differing (and evolving) capabilities of each type of device in both the underlay and overlay networks. This results in the following requirements being placed on the data plane tunneling protocol:

- The data plane is generic and extensible enough to support current and future control planes.

- Tunnel components are efficiently implementable in both hardware and software without restricting capabilities to the lowest common denominator.

- High performance over existing IP fabrics.

These requirements are described further in the following subsections.

2.1. Control Plane Independence

Although some protocols for network virtualization have included a control plane as part of the tunnel format specification (most notably, the original VXLAN spec prescribed a multicast learning-based control plane), these specifications have largely been treated as describing only the data format. The VXLAN packet format has actually seen a wide variety of control planes built on top of it.

There is a clear advantage in settling on a data format: most of the protocols are only superficially different and there is little advantage in duplicating effort. However, the same cannot be said of control planes, which are diverse in very fundamental ways. The case for standardization is also less clear given the wide variety in requirements, goals, and deployment scenarios.

As a result of this reality, Geneve aims to be a pure tunnel format specification that is capable of fulfilling the needs of many control planes by explicitly not selecting any one of them. This simultaneously promotes a shared data format and increases the chances that it will not be obsoleted by future control plane enhancements.

2.2. Data Plane Extensibility

Achieving the level of flexibility needed to support current and future control planes effectively requires an options infrastructure to allow new metadata types to be defined, deployed, and either finalized or retired. Options also allow for differentiation of products by encouraging independent development in each vendor's core specialty, leading to an overall faster pace of advancement. By far the most common mechanism for implementing options is Type-Length-Value (TLV) format.

It should be noted that while options can be used to support non-wirespeed control packets, they are equally important on data packets as well to segregate and direct forwarding (for instance, the examples given before of input port based security policies and service interposition both require tags to be placed on data packets). Therefore, while it would be desirable to limit the extensibility to only control packets for the purposes of simplifying the datapath, that would not satisfy the design requirements.

2.2.1. Efficient Implementation

There is often a conflict between software flexibility and hardware performance that is difficult to resolve. For a given set of functionality, it is obviously desirable to maximize performance. However, that does not mean new features that cannot be run at that speed today should be disallowed. Therefore, for a protocol to be efficiently implementable means that a set of common capabilities can be reasonably handled across platforms along with a graceful mechanism to handle more advanced features in the appropriate situations.

The use of a variable length header and options in a protocol often raises questions about whether it is truly efficiently implementable in hardware. To answer this question in the context of Geneve, it is important to first divide "hardware" into two categories: tunnel endpoints and transit devices.

Endpoints must be able to parse the variable header, including any options, and take action. Since these devices are actively participating in the protocol, they are the most affected by Geneve.

However, as endpoints are the ultimate consumers of the data, transmitters can tailor their output to the capabilities of the recipient. As new functionality becomes sufficiently well defined to add to endpoints, supporting options can be designed using ordering restrictions and other techniques to ease parsing.

Transit devices MAY be able to interpret the options, however, as non-terminating devices, transit devices do not originate or terminate the Geneve packet, hence MUST NOT insert or delete options, which is the responsibility of Geneve endpoints. The participation of transit devices in interpreting options is OPTIONAL.

Further, either tunnel endpoints or transit devices MAY use offload capabilities of NICs such as checksum offload to improve the performance of Geneve packet processing. The presence of a Geneve variable length header SHOULD NOT prevent the tunnel endpoints and transit devices from using such offload capabilities.

2.3. Use of Standard IP Fabrics

IP has clearly cemented its place as the dominant transport mechanism and many techniques have evolved over time to make it robust, efficient, and inexpensive. As a result, it is natural to use IP fabrics as a transit network for Geneve. Fortunately, the use of IP encapsulation and addressing is enough to achieve the primary goal of delivering packets to the correct point in the network through standard switching and routing.

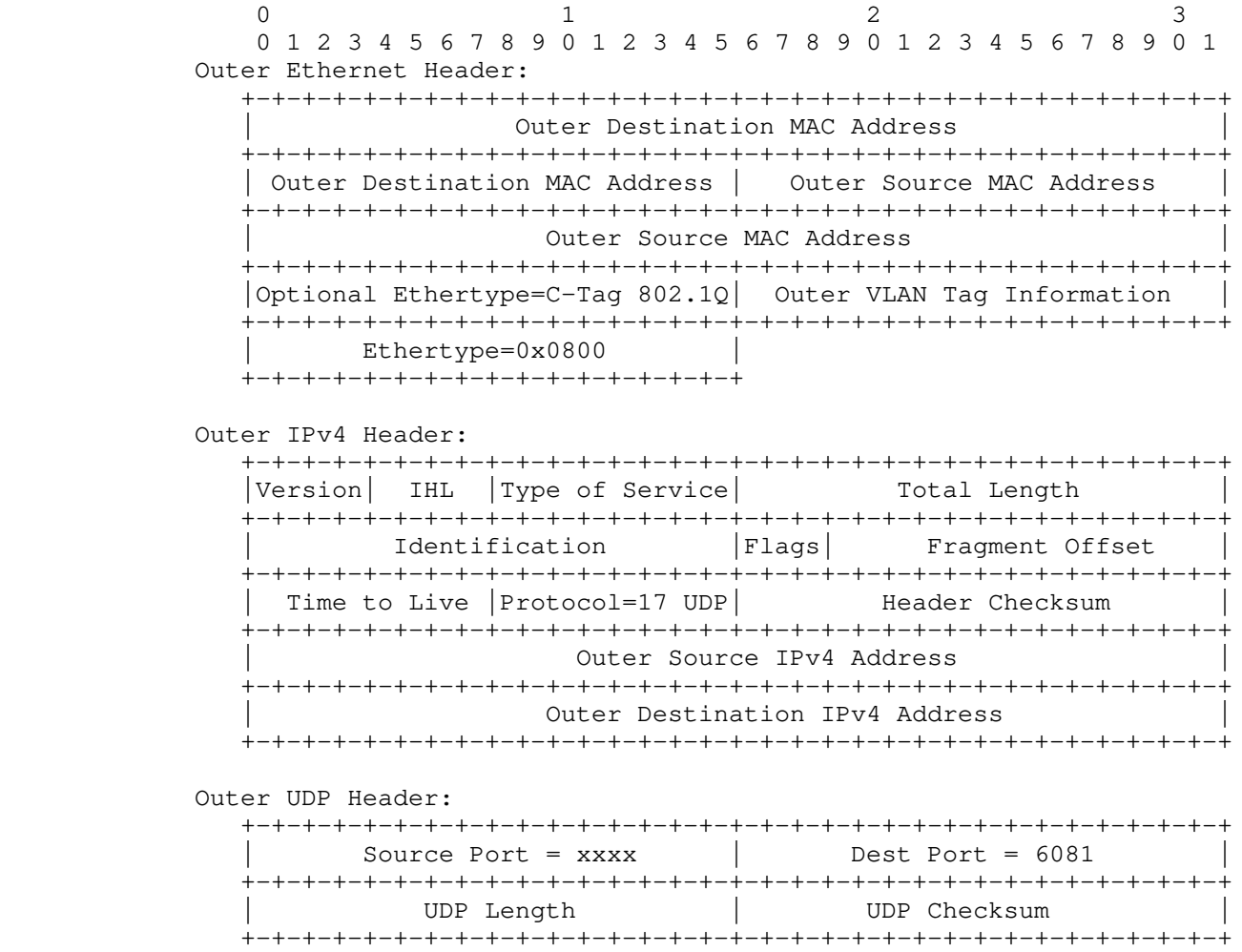
In addition, nearly all underlay fabrics are designed to exploit parallelism in traffic to spread load across multiple links without introducing reordering in individual flows. These equal cost multipathing (ECMP) techniques typically involve parsing and hashing the addresses and port numbers from the packet to select an outgoing link. However, the use of tunnels often results in poor ECMP performance without additional knowledge of the protocol as the encapsulated traffic is hidden from the fabric by design and only endpoint addresses are available for hashing.

Since it is desirable for Geneve to perform well on these existing fabrics, it is necessary for entropy from encapsulated packets to be exposed in the tunnel header. The most common technique for this is to use the UDP source port, which is discussed further in Section 3.3.

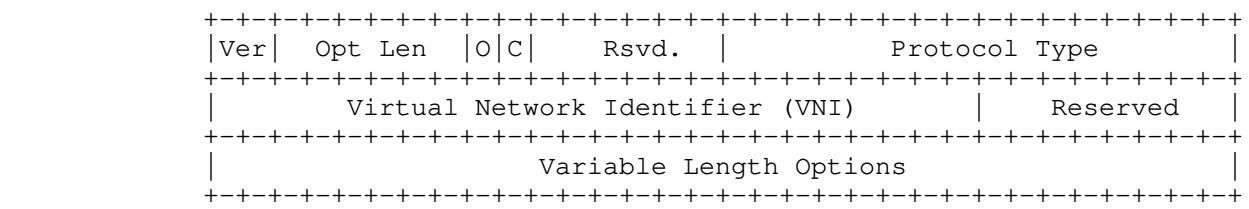
3. Geneve Encapsulation Details

The Geneve packet format consists of a compact tunnel header encapsulated in UDP over either IPv4 or IPv6. A small fixed tunnel header provides control information plus a base level of functionality and interoperability with a focus on simplicity. This header is then followed by a set of variable options to allow for future innovation. Finally, the payload consists of a protocol data unit of the indicated type, such as an Ethernet frame. Section 3.1 and Section 3.2 illustrate the Geneve packet format transported (for example) over Ethernet along with an Ethernet payload.

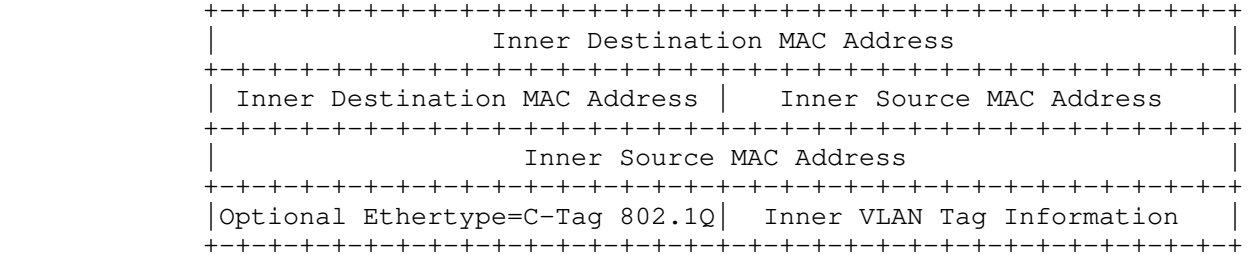
3.1. Geneve Packet Format Over IPv4



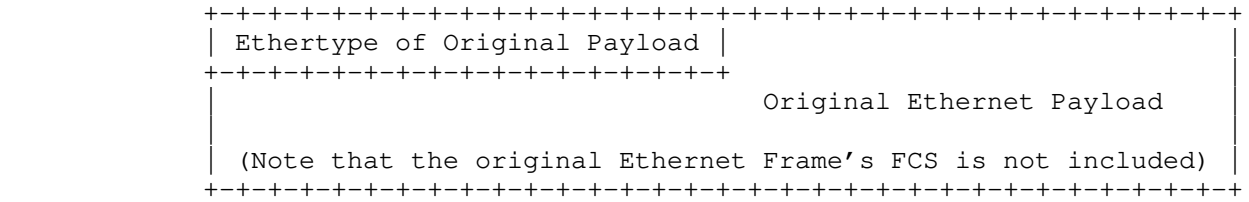
Geneve Header:



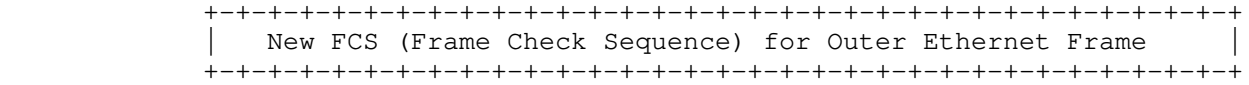
Inner Ethernet Header (example payload):



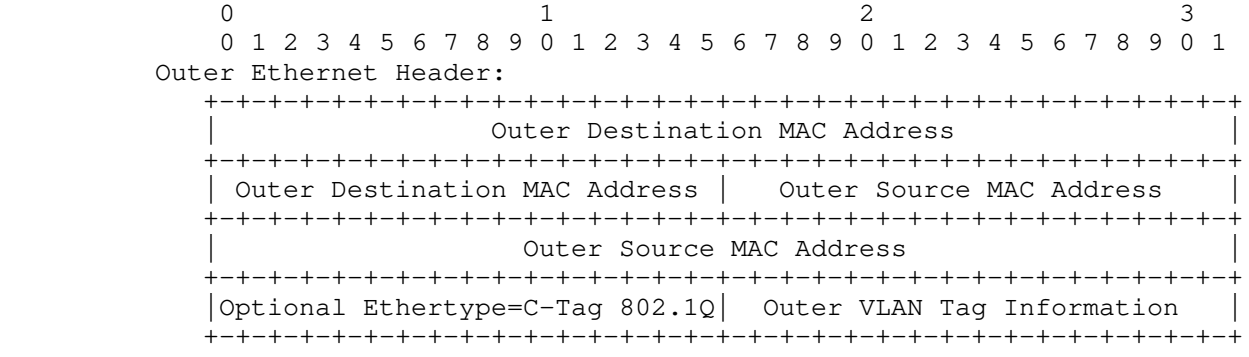
Payload:

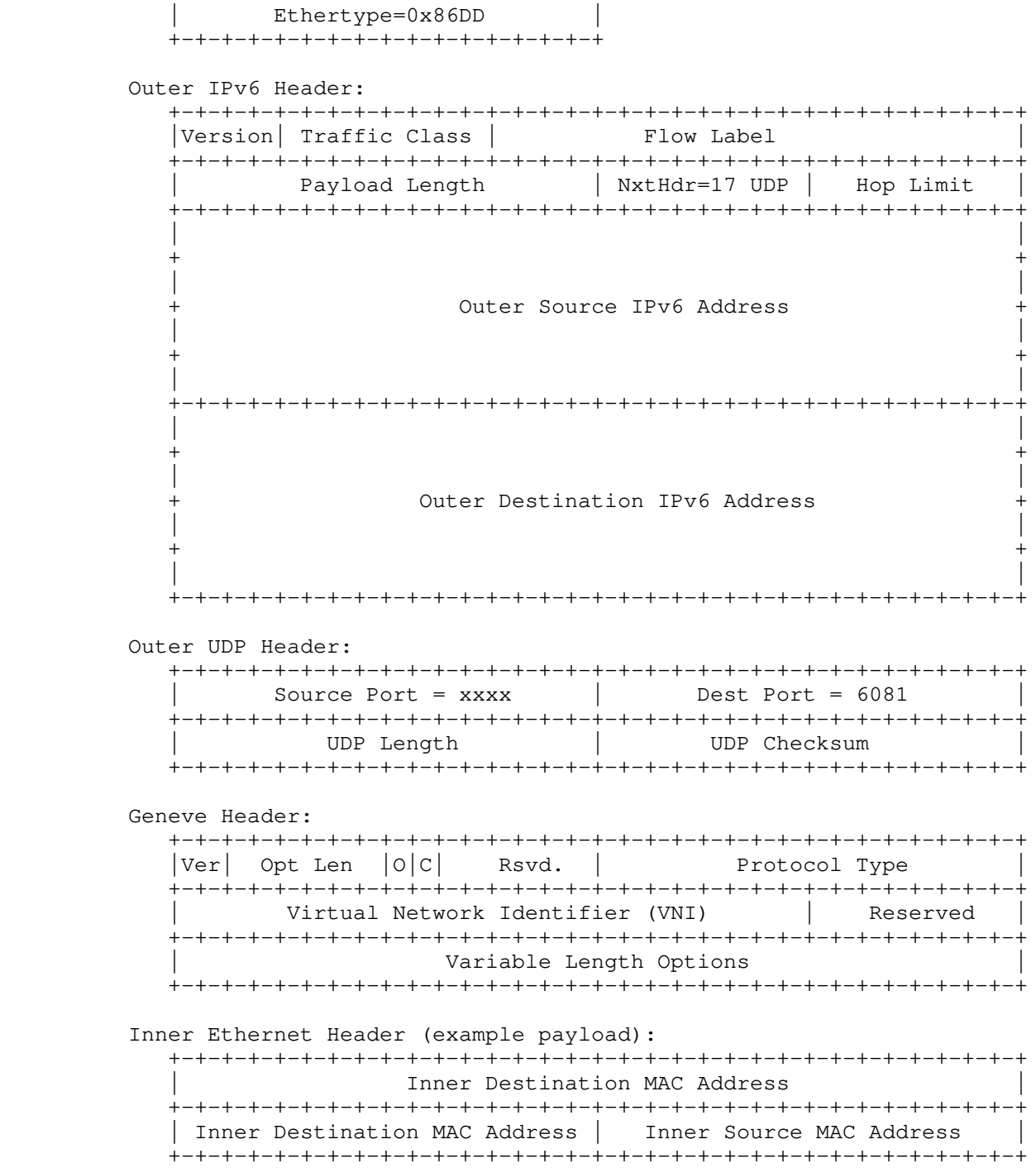


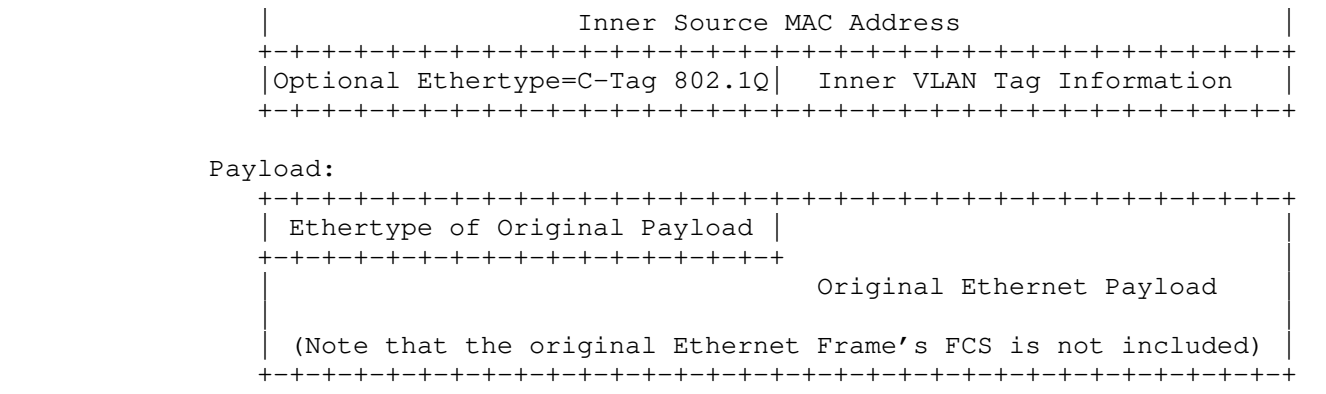
Frame Check Sequence:



3.2. Geneve Packet Format Over IPv6




Ethertype=0x86DD
Outer IPv6 Header:
Version
Traffic Class
Flow Label
Payload Length
NxtHdr=17 UDP
Hop Limit
Outer Source IPv6 Address
Outer Destination IPv6 Address
Outer UDP Header:
Source Port = xxxx
Dest Port = 6081
UDP Length
UDP Checksum
Geneve Header:
Ver
Opt Len
O
C
Rsvd.
Protocol Type
Virtual Network Identifier (VNI)
Reserved
Variable Length Options
Inner Ethernet Header (example payload):
Inner Destination MAC Address
Inner Destination MAC Address
Inner Source MAC Address



3.3. UDP Header

The use of an encapsulating UDP [RFC0768] header follows the connectionless semantics of Ethernet and IP in addition to providing entropy to routers performing ECMP. The header fields are therefore interpreted as follows:

Source port: A source port selected by the originating tunnel endpoint. This source port SHOULD be the same for all packets belonging to a single encapsulated flow to prevent reordering due to the use of different paths. To encourage an even distribution of flows across multiple links, the source port SHOULD be calculated using a hash of the encapsulated packet headers using, for example, a traditional 5-tuple. Since the port represents a flow identifier rather than a true UDP connection, the entire 16-bit range MAY be used to maximize entropy.

Dest port: IANA has assigned port 6081 as the fixed well-known destination port for Geneve. Although the well-known value should be used by default, it is RECOMMENDED that implementations make this configurable. The chosen port is used for identification of Geneve packets and MUST NOT be reversed for different ends of a connection as is done with TCP.

UDP length: The length of the UDP packet including the UDP header.

UDP checksum: The checksum MAY be set to zero on transmit for

packets encapsulated in both IPv4 and IPv6 [RFC6935]. When a packet is received with a UDP checksum of zero it MUST be accepted and decapsulated. If the originating tunnel endpoint optionally encapsulates a packet with a non-zero checksum, it MUST be a correctly computed UDP checksum. Upon receiving such a packet, the egress endpoint MUST validate the checksum. If the checksum is not correct, the packet MUST be dropped, otherwise the packet MUST be accepted for decapsulation. It is RECOMMENDED that the UDP checksum be computed to protect the Geneve header and options in situations where the network reliability is not high and the packet is not protected by another checksum or CRC.

3.4. Tunnel Header Fields

Ver (2 bits): The current version number is 0. Packets received by an endpoint with an unknown version MUST be dropped. Non-terminating devices processing Geneve packets with an unknown version number MUST treat them as UDP packets with an unknown payload.

Opt Len (6 bits): The length of the options fields, expressed in four byte multiples, not including the eight byte fixed tunnel header. This results in a minimum total Geneve header size of 8 bytes and a maximum of 260 bytes. The start of the payload headers can be found using this offset from the end of the base Geneve header.

O (1 bit): OAM packet. This packet contains a control message instead of a data payload. Control messages are sent between Geneve endpoints. Endpoints MUST NOT forward the payload and transit devices MUST NOT attempt to interpret or process it. Since these are infrequent control messages, it is RECOMMENDED that endpoints direct these packets to a high priority control queue (for example, to direct the packet to a general purpose CPU from a forwarding ASIC or to separate out control traffic on a NIC). Transit devices MUST NOT alter forwarding behavior on the basis of this bit, such as ECMP link selection.

C (1 bit): Critical options present. One or more options has the critical bit set (see Section 3.5). If this bit is set then tunnel endpoints MUST parse the options list to interpret any critical options. On endpoints where option parsing is not supported the packet MUST be dropped on the basis of the 'C' bit in the base header. If the bit is not set tunnel endpoints MAY strip all options using 'Opt Len' and forward the decapsulated packet. Transit devices MUST NOT drop packets on the basis of this bit.

The critical bit allows hardware implementations the flexibility to handle options processing in the hardware fastpath or in the exception (slow) path without the need to process all the options. For example, a critical option such as secure hash to provide Geneve header integrity check must be processed by tunnel endpoints and typically processed in the hardware fastpath.

Rsvd. (6 bits): Reserved field which MUST be zero on transmission and ignored on receipt.

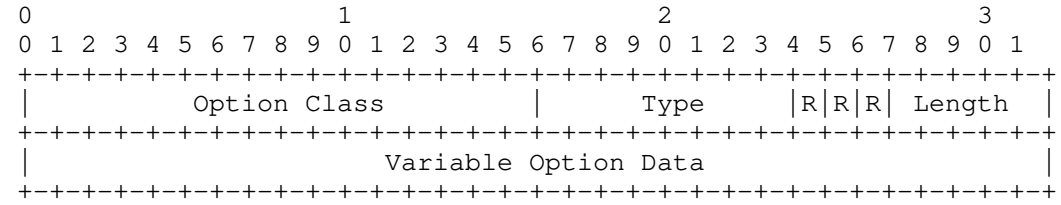
Protocol Type (16 bits): The type of the protocol data unit appearing after the Geneve header. This follows the EtherType [ETYPES] convention with Ethernet itself being represented by the value 0x6558.

Virtual Network Identifier (VNI) (24 bits): An identifier for a unique element of a virtual network. In many situations this may represent an L2 segment, however, the control plane defines the forwarding semantics of decapsulated packets. The VNI MAY be used as part of ECMP forwarding decisions or MAY be used as a mechanism to distinguish between overlapping address spaces contained in the encapsulated packet when load balancing across CPUs.

Reserved (8 bits): Reserved field which MUST be zero on transmission and ignored on receipt.

Transit devices MUST maintain consistent forwarding behavior irrespective of the value of 'Opt Len', including ECMP link selection. These devices SHOULD be able to forward packets containing options without resorting to a slow path.

3.5. Tunnel Options



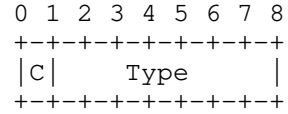
Geneve Option

The base Geneve header is followed by zero or more options in Type-Length-Value format. Each option consists of a four byte option header and a variable amount of option data interpreted according to the type.

Option Class (16 bits): Namespace for the 'Type' field. IANA will be requested to create a "Geneve Option Class" registry to allocate identifiers for organizations, technologies, and vendors that have an interest in creating types for options. Each organization may allocate types independently to allow experimentation and rapid innovation. It is expected that over time certain options will become well known and a given implementation may use option types from a variety of sources. In addition, IANA will be requested to reserve specific ranges for standardized and experimental options.

Type (8 bits): Type indicating the format of the data contained in this option. Options are primarily designed to encourage future extensibility and innovation and so standardized forms of these options will be defined in a separate document.

The high order bit of the option type indicates that this is a critical option. If the receiving endpoint does not recognize this option and this bit is set then the packet MUST be dropped. If the critical bit is set in any option then the 'C' bit in the Geneve base header MUST also be set. Transit devices MUST NOT drop packets on the basis of this bit. The following figure shows the location of the 'C' bit in the 'Type' field:



The requirement to drop a packet with an unknown critical option applies to the entire tunnel endpoint system and not a particular component of the implementation. For example, in a system comprised of a forwarding ASIC and a general purpose CPU, this does not mean that the packet must be dropped in the ASIC. An implementation may send the packet to the CPU using a rate-limited control channel for slow-path exception handling.

R (3 bits): Option control flags reserved for future use. MUST be zero on transmission and ignored on receipt.

Length (5 bits): Length of the option, expressed in four byte multiples excluding the option header. The total length of each option may be between 4 and 128 bytes. A value of 0 in the Length field implies an option with only the option header without the variable option data. Packets in which the total length of all options is not equal to the 'Opt Len' in the base header are invalid and MUST be silently dropped if received by an endpoint.

Variable Option Data: Option data interpreted according to 'Type'.

3.5.1. Options Processing

Geneve options are intended to be originated and processed by tunnel endpoints. However, options MAY be interpreted by transit devices along the tunnel path. Transit devices not processing Geneve headers SHOULD process Geneve packets as any other UDP packet and maintain consistent forwarding behavior.

In tunnel endpoints, the generation and interpretation of options is determined by the control plane, which is out of the scope of this document. However, to ensure interoperability between heterogeneous devices some requirements are imposed on options and the devices that process them:

- Receiving endpoints MUST drop packets containing unknown options with the 'C' bit set in the option type. Conversely, transit devices MUST NOT drop packets as a result of encountering unknown options, including those with the 'C' bit set.

- Some options may be defined in such a way that the position in the option list is significant. Options or their ordering, MUST NOT be changed by transit devices.

- An option MUST NOT affect the parsing or interpretation of any other option.

When designing a Geneve option, it is important to consider how the option will evolve in the future. Once an option is defined it is reasonable to expect that implementations may come to depend on a specific behavior. As a result, the scope of any future changes must be carefully described upfront.

Unexpectedly significant interoperability issues may result from changing the length of an option that was defined to be a certain size. A particular option is specified to have either a fixed length, which is constant, or a variable length, which may change over time or for different use cases. This property is part of the definition of the option and conveyed by the 'Type'. For fixed length options, some implementations may choose to ignore the length field in the option header and instead parse based on the well known length associated with the type. In this case, redefining the length will impact not only parsing of the option in question but also any options that follow. Therefore, options that are defined to be fixed length in size MUST NOT be redefined to a different length. Instead, a new 'Type' should be allocated.

4. Implementation and Deployment Considerations

4.1. Encapsulation of Geneve in IP

As an IP-based tunnel protocol, Geneve shares many properties and techniques with existing protocols. The application of some of these are described in further detail, although in general most concepts applicable to the IP layer or to IP tunnels generally also function in the context of Geneve.

4.1.1. IP Fragmentation

To prevent fragmentation and maximize performance, the best practice when using Geneve is to ensure that the MTU of the physical network is greater than or equal to the MTU of the encapsulated network plus tunnel headers. Manual or upper layer (such as TCP MSS clamping) configuration can be used to ensure that fragmentation never takes place, however, in some situations this may not be feasible.

It is strongly RECOMMENDED that Path MTU Discovery ([RFC1191], [RFC1981]) be used by setting the DF bit in the IP header when Geneve packets are transmitted over IPv4 (this is the default with IPv6). The use of Path MTU Discovery on the transit network provides the encapsulating endpoint with soft-state about the link that it may use to prevent or minimize fragmentation depending on its role in the virtualized network. For example, recommendations/guidance for handling fragmenation in similar overlay encapsulation services like PWE3 are provided in section 5.3 of [RFC3985].

Note that some implementations may not be capable of supporting fragmentation or other less common features of the IP header, such as options and extension headers.

4.1.2. DSCP and ECN

When encapsulating IP (including over Ethernet) packets in Geneve, there are several considerations for propagating DSCP and ECN bits from the inner header to the tunnel on transmission and the reverse on reception.

[RFC2983] provides guidance for mapping DSCP between inner and outer IP headers. Network virtualization is typically more closely aligned with the Pipe model described, where the DSCP value on the tunnel header is set based on a policy (which may be a fixed value, one based on the inner traffic class, or some other mechanism for grouping traffic). Aspects of the Uniform model (which treats the inner and outer DSCP value as a single field by copying on ingress and egress) may also apply, such as the ability to remark the inner

header on tunnel egress based on transit marking. However, the Uniform model is not conceptually consistent with network virtualization, which seeks to provide strong isolation between encapsulated traffic and the physical network.

[RFC6040] describes the mechanism for exposing ECN capabilities on IP tunnels and propagating congestion markers to the inner packets. This behavior MUST be followed for IP packets encapsulated in Geneve.

4.1.3. Broadcast and Multicast

Geneve tunnels may either be point-to-point unicast between two endpoints or may utilize broadcast or multicast addressing. It is not required that inner and outer addressing match in this respect. For example, in physical networks that do not support multicast, encapsulated multicast traffic may be replicated into multiple unicast tunnels or forwarded by policy to a unicast location (possibly to be replicated there).

With physical networks that do support multicast it may be desirable to use this capability to take advantage of hardware replication for encapsulated packets. In this case, multicast addresses may be allocated in the physical network corresponding to tenants, encapsulated multicast groups, or some other factor. The allocation of these groups is a component of the control plane and therefore outside of the scope of this document. When physical multicast is in use, the 'C' bit in the Geneve header may be used with groups of devices with heterogeneous capabilities as each device can interpret only the options that are significant to it if they are not critical.

4.1.4. Unidirectional Tunnels

Generally speaking, a Geneve tunnel is a unidirectional concept. IP is not a connection oriented protocol and it is possible for two endpoints to communicate with each other using different paths or to have one side not transmit anything at all. As Geneve is an IP-based protocol, the tunnel layer inherits these same characteristics.

It is possible for a tunnel to encapsulate a protocol, such as TCP, which is connection oriented and maintains session state at that layer. In addition, implementations MAY model Geneve tunnels as connected, bidirectional links, such as to provide the abstraction of a virtual port. In both of these cases, bidirectionality of the tunnel is handled at a higher layer and does not affect the operation of Geneve itself.

4.2. Constraints on Protocol Features

Geneve is intended to be flexible to a wide range of current and future applications. As a result, certain constraints may be placed on the use of metadata or other aspects of the protocol in order to optimize for a particular use case. For example, some applications may limit the types of options which are supported or enforce a maximum number or length of options. Other applications may only handle certain encapsulated payload types, such as Ethernet or IP. This could be either globally throughout the system or, for example, restricted to certain classes of devices or network paths.

These constraints may be communicated to tunnel endpoints either explicitly through a control plane or implicitly by the nature of the application. As Geneve is defined as a data plane protocol that is control plane agnostic, the exact mechanism is not defined in this document.

4.2.1. Constraints on Options

While Geneve options are more flexible, a control plane may restrict the number of option TLVs as well as the order and size of the TLVs, between tunnel endpoints, to make it simpler for a data plane implementation in software or hardware to handle [I-D.ietf-nvo3-encap]. For example, there may be some critical information such as a secure hash that must be processed in a certain order to provide lowest latency.

A control plane may negotiate a subset of option TLVs and certain TLV ordering, as well may limit the total number of option TLVs present in the packet, for example, to accommodate hardware capable of processing fewer options [I-D.ietf-nvo3-encap]. Hence, a control plane needs to have the ability to describe the supported TLVs subset and their order to the tunnel end points. In the absence of a control plane, alternative configuration mechanisms may be used for this purpose. The exact mechanism is not defined in this document.

4.3. NIC Offloads

Modern NICs currently provide a variety of offloads to enable the efficient processing of packets. The implementation of many of these offloads requires only that the encapsulated packet be easily parsed (for example, checksum offload). However, optimizations such as LSO and LRO involve some processing of the options themselves since they must be replicated/merged across multiple packets. In these situations, it is desirable to not require changes to the offload logic to handle the introduction of new options. To enable this,

some constraints are placed on the definitions of options to allow for simple processing rules:

- When performing LSO, a NIC MUST replicate the entire Geneve header and all options, including those unknown to the device, onto each resulting segment. However, a given option definition may override this rule and specify different behavior in supporting devices. Conversely, when performing LRO, a NIC MAY assume that a binary comparison of the options (including unknown options) is sufficient to ensure equality and MAY merge packets with equal Geneve headers.

- Options MUST NOT be reordered during the course of offload processing, including when merging packets for the purpose of LRO.

- NICs performing offloads MUST NOT drop packets with unknown options, including those marked as critical.

There is no requirement that a given implementation of Geneve employ the offloads listed as examples above. However, as these offloads are currently widely deployed in commercially available NICs, the rules described here are intended to enable efficient handling of current and future options across a variety of devices.

4.4. Inner VLAN Handling

Geneve is capable of encapsulating a wide range of protocols and therefore a given implementation is likely to support only a small subset of the possibilities. However, as Ethernet is expected to be widely deployed, it is useful to describe the behavior of VLANs inside encapsulated Ethernet frames.

As with any protocol, support for inner VLAN headers is OPTIONAL. In many cases, the use of encapsulated VLANs may be disallowed due to security or implementation considerations. However, in other cases trunking of VLAN frames across a Geneve tunnel can prove useful. As a result, the processing of inner VLAN tags upon ingress or egress from a tunnel endpoint is based upon the configuration of the endpoint and/or control plane and not explicitly defined as part of the data format.

5. Interoperability Issues

Viewed exclusively from the data plane, Geneve does not introduce any interoperability issues as it appears to most devices as UDP packets. However, as there are already a number of tunnel protocols deployed in network virtualization environments, there is a practical question of transition and coexistence.

Since Geneve is a superset of the functionality of the most common protocols used for network virtualization (VXLAN, NVGRE) it should be straightforward to port an existing control plane to run on top of it with minimal effort. With both the old and new packet formats supporting the same set of capabilities, there is no need for a hard transition - endpoints directly communicating with each other use any common protocol, which may be different even within a single overall system. As transit devices are primarily forwarding packets on the basis of the IP header, all protocols appear similar and these devices do not introduce additional interoperability concerns.

To assist with this transition, it is strongly suggested that implementations support simultaneous operation of both Geneve and existing tunnel protocols as it is expected to be common for a single node to communicate with a mixture of other nodes. Eventually, older protocols may be phased out as they are no longer in use.

6. Security Considerations

As encapsulated within an UDP/IP packet, Geneve does not have any inherent security mechanisms. As a result, an attacker with access to the underlay network transporting the IP packets has the ability to snoop or inject packets. Legitimate but malicious tunnel endpoints may also spoof identifiers in the tunnel header to gain access to networks owned by other tenants.

Within a particular security domain, such as a data center operated by a single service provider, the most common and highest performing security mechanism is isolation of trusted components. Tunnel traffic can be carried over a separate VLAN and filtered at any untrusted boundaries. In addition, tunnel endpoints should only be operated in environments controlled by the service provider, such as the hypervisor itself rather than within a customer VM.

When crossing an untrusted link, such as the public Internet, IPsec [RFC4301] may be used to provide authentication and/or encryption of the IP packets formed as part of Geneve encapsulation.

Geneve does not otherwise affect the security of the encapsulated packets. As per the guidelines of BCP72 [RFC3552], the following sections describe potential security risks that may be applicable to Geneve deployments and approaches to mitigate such risks. It is also noted that not all such risks are applicable to all Geneve deployment scenarios, i.e., only a subset may be applicable to certain deployments. So an operator has to make an assessment based on their network environment and determine the risks that are applicable to their specific environment and use appropriate mitigation approaches as applicable.

6.1. Data Confidentiality

Geneve is a network virtualization overlay encapsulation protocol designed to establish tunnels between network virtualization end points (NVE) over an existing IP network. It can be used to deploy multi-tenant overlay networks over an existing IP underlay network in a public or private data center. The overlay service is typically provided by a service provider, for example a cloud services provider or a private data center operator. Due to the nature of multi-tenancy in such environments, a tenant system may expect data confidentiality to ensure its packet data is not tampered with (active attack) in transit or a target of unauthorized monitoring (passive attack). A tenant may expect the overlay service provider to provide data confidentiality as part of the service or a tenant may bring its own data confidentiality mechanisms like IPsec or TLS to protect the data end to end between its tenant systems.

If an operator determines data confidentiality is necessary in their environment based on their risk analysis, for example as in multi-tenant environments, then an encryption mechanism SHOULD be used to encrypt the tenant data end to end between the NVEs. The NVEs may use existing well established encryption mechanisms such as IPsec, DTLs, etc., The operator may choose not to enable the encryption if, for example, the packet data is already encrypted by the tenant system.

6.1.1. Inter-data center traffic

A tenant system in a customer premises (private data center) may want to connect to tenant systems on their tenant overlay network in a public cloud data center or a tenant may want to have its tenant systems located in multiple geographically separated data centers for high availability. Geneve data traffic between tenant systems across such separated networks should be protected from threats when traversing public networks. Any Geneve overlay data leaving the data center network beyond the operator's security domain, for example over the public Internet, SHOULD be secured by encryption mechanisms such as IPsec or other VPN mechanisms to protect the communications between the NVEs when they are geographically separated over untrusted network links. Implementation of specific data protection mechanisms employed between data centers is beyond the scope of this document.

6.2. Data Integrity

Geneve encapsulation is used between NVEs to establish overlay tunnels over an existing IP underlay network. In a multi-tenant data center, a rogue or compromised tenant system may try to launch a

passive attack such as monitoring the traffic of other tenants, or an active attack such as spoofing or trying to inject unauthorized Geneve encapsulated traffic into the network. To prevent such attacks, an NVE MUST not propagate Geneve packets beyond the NVE to tenant systems and SHOULD employ packet filtering mechanisms so as not to forward unauthorized traffic between TSs in different tenant networks.

A compromised network node or a transit device within a data center may launch an active attack trying to tamper with the Geneve packet data between NVEs. Malicious tampering of Geneve header fields may cause the packet from one tenant to be forwarded to a different tenant network. If an operator determines the possibility of such threat in their environment, the operator may choose to employ data integrity mechanisms between NVEs. In order to prevent such risks, a data integrity mechanism SHOULD be used in such environments to protect the integrity of Geneve packets including packet headers, options and payload on communications between NVE pairs. A cryptographic data protection mechanism such as IPsec may be used to provide data integrity protection. A data center operator may choose to deploy any other data integrity mechanisms as applicable and supported in their underlay networks.

Geneve supports Geneve Options, so an operator may choose to use a Geneve option TLV to provide a cryptographic data protection mechanism, to verify the data integrity of the Geneve header, Geneve options or the entire Geneve packet including the payload. Implementation of such a mechanism is beyond the scope of this document.

6.3. Authentication of NVE peers

A rogue network device or a compromised NVE in a data center environment might be able to spoof Geneve packets as if it came from a legitimate NVE. In order to mitigate such a risk, an operator SHOULD use an Authentication mechanism, such as IPsec to ensure that the Geneve packet originated from the intended NVE peer, in environments where the operator determines spoofing or rogue devices is a potential threat. Other simpler source checks such as ingress filtering for VLAN/MAC/IP address, reverse path forwarding checks, etc., may be used in certain trusted environments to ensure Geneve packets originated from the intended NVE peer.

6.4. Multicast/Broadcast

In typical data center networks where IP multicasting is not supported in the underlay network, multicasting may be supported using multiple unicast tunnels. The same security requirements as

described in the above sections can be used to protect Geneve communications between NVE peers. If IP multicasting is supported in the underlay network and the operator chooses to use it for multicast traffic among Geneve endpoints, then the operator in such environments may use data protection mechanisms such as IPsec with Multicast extensions [RFC5374] to protect multicast traffic among Geneve NVE groups.

6.5. Control plane communications

A Network Virtualization Authority (NVA) as outlined in [RFC8014] may be used as a control plane for configuring and managing the Geneve NVEs. The data center operator is expected to use security mechanisms to protect the communications between the NVA to NVEs and use authentication mechanisms to detect any rogue or compromised NVEs within their administrative domain. Data protection mechanisms for control plane communication or authentication mechanisms between the NVA and the NVEs is beyond the scope of this document.

7. IANA Considerations

IANA has allocated UDP port 6081 as the well-known destination port for Geneve. Upon publication, the registry should be updated to cite this document. The original request was:

Service Name: geneve
Transport Protocol(s): UDP
Assignee: Jesse Gross <jgross@vmware.com>
Contact: Jesse Gross <jgross@vmware.com>
Description: Generic Network Virtualization Encapsulation (Geneve)
Reference: This document
Port Number: 6081

In addition, IANA is requested to create a "Geneve Option Class" registry to allocate Option Classes. This shall be a registry of 16-bit hexadecimal values along with descriptive strings. The identifiers 0x0-0xFF are to be reserved for standardized options for allocation by IETF Review [RFC5226] and 0xFFF0-0xFFFF for Experimental Use. Otherwise, identifiers are to be assigned to any organization with an interest in creating Geneve options on a First Come First Served basis. The registry is to be populated with the following initial values:

Option Class	Description
0x0000..0x00FF	Unassigned - IETF Review
0x0100	Linux
0x0101	Open vSwitch
0x0102	Open Virtual Networking (OVN)
0x0103	In-band Network Telemetry (INT)
0x0104	VMware
0x0105	Amazon
0x0106	Cisco
0x0107..0xFFEF	Unassigned - First Come First Served
0xFFF0..FFFF	Experimental

8. Contributors

The following individuals were authors of an earlier version of this document and made significant contributions:

Pankaj Garg
Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052
USA

Email: pankajg@microsoft.com

Chris Wright
Red Hat Inc.
1801 Varsity Drive
Raleigh, NC 27606
USA

Email: chrisw@redhat.com

Puneet Agarwal
Innovium, Inc.
6001 America Center Drive
San Jose, CA 95002
USA

Email: puneet@innovium.com

Kenneth Duda
Arista Networks
5453 Great America Parkway
Santa Clara, CA 95054

USA

Email: kduda@arista.com

Dinesh G. Dutt
Cumulus Networks
140C S. Whisman Road
Mountain View, CA 94041
USA

Email: ddutt@cumulusnetworks.com

Jon Hudson
Independent

Email: jon.hudson@gmail.com

Ariel Hendel
Facebook, Inc.
1 Hacker Way
Menlo Park, CA 94025
USA

Email: ahendel@fb.com

9. Acknowledgements

The authors wish to thank Martin Casado, Bruce Davie and Dave Thaler for their input, feedback, and helpful suggestions.

10. References

10.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[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>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

10.2. Informative References

[ETYPES] The IEEE Registration Authority, "IEEE 802 Numbers", 2013, <http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xml>.

[I-D.ietf-nvo3-dataplane-requirements] Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", draft-ietf-nvo3-dataplane-requirements-03 (work in progress), April 2014.

[I-D.ietf-nvo3-encap] Boutros, S., Ganga, I., Garg, P., Manur, R., Mizrahi, T., Mozes, D., Nordmark, E., Smith, M., Aldrin, S., and I. Bagdonas, "NVO3 Encapsulation Considerations", draft-ietf-nvo3-encap-01 (work in progress), October 2017.

[IEEE.802.1Q_2014] IEEE, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks", IEEE 802.1Q-2014, DOI 10.1109/ieeestd.2014.6991462, December 2014, <http://ieeexplore.ieee.org/servlet/opac?punumber=6991460>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, DOI 10.17487/RFC1981, August 1996, <https://www.rfc-editor.org/info/rfc1981>.

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, DOI 10.17487/RFC2983, October 2000, <https://www.rfc-editor.org/info/rfc2983>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <https://www.rfc-editor.org/info/rfc3031>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <https://www.rfc-editor.org/info/rfc3985>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC5374] Weis, B., Gross, G., and D. Ignjatic, "Multicast Extensions to the Security Architecture for the Internet Protocol", RFC 5374, DOI 10.17487/RFC5374, November 2008, <https://www.rfc-editor.org/info/rfc5374>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, DOI 10.17487/RFC6935, April 2013, <https://www.rfc-editor.org/info/rfc6935>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <https://www.rfc-editor.org/info/rfc7365>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <https://www.rfc-editor.org/info/rfc7637>.

[RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <https://www.rfc-editor.org/info/rfc8014>.

[VL2] Greenberg, A., et al., "VL2: A Scalable and Flexible Data Center Network", ACM SIGCOMM Computer Communication Review, DOI 10.1145/1594977.1592576, 2009, <http://www.sigcomm.org/sites/default/files/ccr/papers/2009/October/1594977-1592576.pdf>.

Authors' Addresses

Jesse Gross (editor)

Email: jesse@kernel.org

Ilango Ganga (editor)
Intel Corporation
2200 Mission College Blvd.
Santa Clara, CA 95054
USA

Email: ilango.s.ganga@intel.com

T. Sridhar (editor)
VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
USA

Email: tsridhar@vmware.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 3, 2018

J. Lemon, Ed.
Broadcom
F. Maino
M. Smith
Cisco

March 2, 2018

Group Policy Encoding with VXLAN-GPE
draft-lemon-vxlan-gpe-gbp-02

Abstract

This document defines a header companion for the Generic Protocol Extension for Virtual eXtensible Local Area Network (VXLAN-GPE) that is used to carry a Group Policy Identifier for the purposes of policy enforcement.

Status of This Memo

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

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

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

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

Copyright Notice

Table of Contents

1. Introduction 2
1.1. Conventions 2
1.2. Abbreviations used in this document 2
2. Group Based Policy Sub-header 2
2.1. Header Format 2
3. IANA Considerations 4
4. Security Considerations 4
5. Normative References 4
Authors' Addresses 5

1. Introduction

This document defines the group-based policy (GBP) sub-header for VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe]. The GBP sub-header carries a 16-bit group policy ID that is semantically equivalent to the 16-bit group policy ID defined in [I-D.smith-vxlan-group-policy].

1.1. Conventions

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

1.2. Abbreviations used in this document

GBP: Group-Based Policy

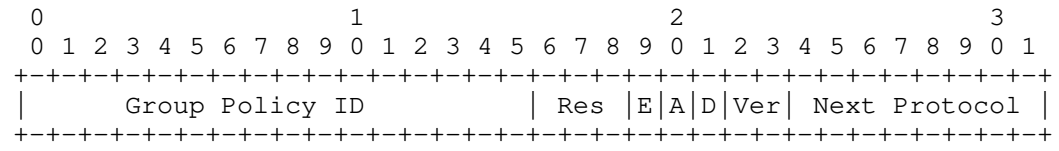
VXLAN-GPE: Virtual eXtensible Local Area Network, Generic Protocol Extension [I-D.ietf-nvo3-vxlan-gpe]

2. Group Based Policy Sub-header

The Group-Based Policy (GBP) Sub-header follows the VXLAN-GPE header, or another VXLAN-GPE subheader.

2.1. Header Format

The format of the GBP sub-header is as shown below:



- Group Policy ID: 16 bit identifier that indicates the Group Policy ID being encapsulated by this GBP sub-header. The allocation of Group Policy ID values is outside the scope of this document.

- Reserved (Res): the 3 bit field MUST be set to zero on transmission and ignored on receipt.

- End Destination bit (E bit): The E bit is set to 0 to represent the Group Policy ID associated with the source of the packet. The E bit is set to 1 to represent the Group Policy ID associated with the end destination of the packet. Note that if the packet carryies a destination group sub-header, it MUST also carry a source group sub-header.

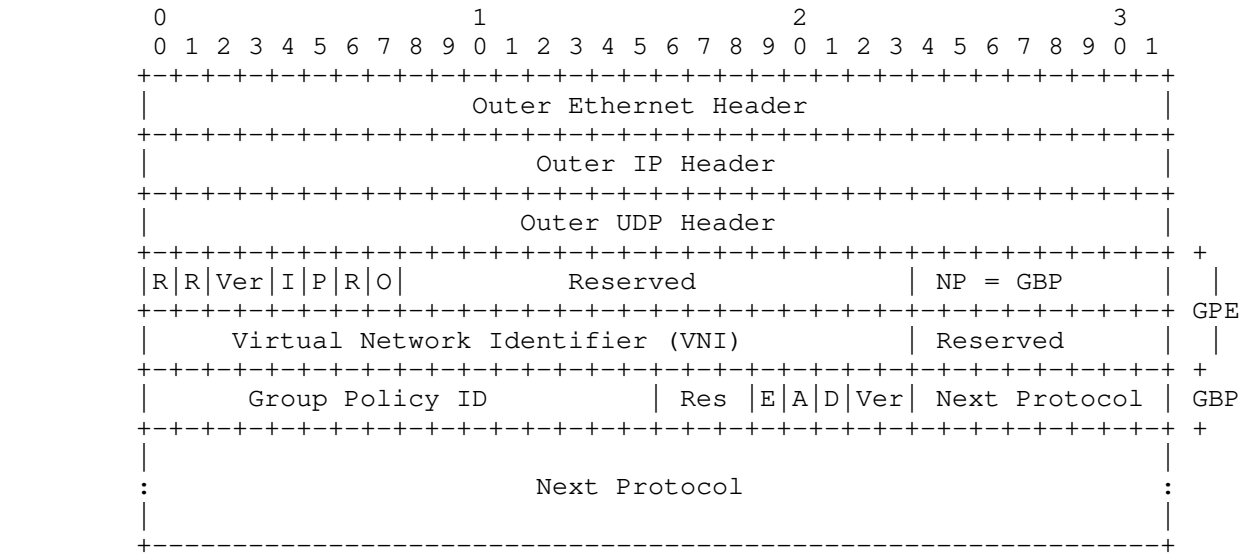
- Policy Applied bit (A bit): The A bit is set to 0 to indicate that the group policy has not (yet) been applied to this packet. Group policies MUST be applied by devices when the A bit is set to 0 and the destination Group has been determined. Devices that apply the group policy MUST set the A bit to 1 after the policy has been applied. The A bit is set to 1 to indicate that the group policy has already been applied to this packet. Policies that redirect the packet MUST NOT be applied by devices when the A bit is set. Policies that cause the packet to be dropped MAY be applied.

- Don't Learn bit (D bit): The D bit is set to 1 to indicate that the egress VTEP MUST NOT learn the source address of the encapsulated frame.

- Version (Ver): indicates the Version of the Group Policy VXLAN-GPE sub-header. The initial version is 0.

- Next Protocol: This 8 bit field indicates the protocol header immediately following this VXLAN GPE sub-header. Next Protocol types are encoded as specified in [I-D.ietf-nvo3-vxlan-gpe].

An example frame format is as shown below:



3. IANA Considerations

IANA is requested to add a new value to registry of "Next Protocol", which is defined in [I-D.ietf-nvo3-vxlan-gpe]. The new value of 6 will signify a GBP sub-header as the next protocol.

4. Security Considerations

The same security considerations applied to [I-D.ietf-nvo3-vxlan-gpe] and to [I-D.smith-vxlan-group-policy] apply to this document.

Additionally, the security policy value carried in the GBP header impacts security directly. There is a risk that this identifier could be altered. Accordingly, the network should be designed such that this header can be inserted only by trusted entities, and can not be altered before reaching the destination. This can be mitigated through physical security of the network and/or by encryption or validation of the entire packet, including the GBP.

5. Normative References

[I-D.ietf-nvo3-vxlan-gpe]
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-05 (work in progress), October 2017.

[I-D.smith-vxlan-group-policy]
Smith, M. and L. Kreeger, "VXLAN Group Policy Option", draft-smith-vxlan-group-policy-04 (work in progress), October 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

Authors' Addresses

John Lemon (editor)
Broadcom Limited
270 Innovation Drive
San Jose, CA 95134
USA

Email: john.lemon@broadcom.com

Fabio Maino
Cisco Systems

Email: fmaino@cisco.com

Michael Smith
Cisco Systems

Email: michsmit@cisco.com

NVO3
Internet-Draft
Intended status: Informational
Expires: May 8, 2019

D. Migault
Ericsson
S. Boutros
D. Wings
VMware, Inc.
S. Krishnan
Kaloom

November 04, 2018

Geneve Security Requirements
draft-mglt-nvo3-geneve-security-requirements-05

Abstract

The document defines the security requirements to protect tenants overlay traffic against security threats from the NVO3 network components that are interconnected with tunnels implemented using Generic Network Virtualization Encapsulation (Geneve).

The document provides two sets of security requirements: 1. requirements to evaluate the data plane security of a given deployment of Geneve overlay. Such requirements are intended to Geneve overlay provider to evaluate a given deployment.
2. requirement a security mechanism need to fulfill to secure any deployment of Geneve overlay deployment

Table of Contents

1. Requirements Notation 2
2. Introduction 3
3. Terminology 6
4. Security Threats 6
4.1. Passive Attacks 6
4.2. Active Attacks 7
5. Requirements for Security Mitigations 8
5.1. Protection Against Traffic Sniffing 8
5.2. Protecting Against Traffic Injection 10
5.3. Protecting Against Traffic Redirection 11
5.4. Protecting Against Traffic Replay 12
5.5. Security Management 13
6. IANA Considerations 14
7. Security Considerations 14
8. Appendix 14
8.1. TLS 14
8.2. IPsec 15
9. Acknowledgments 15
10. References 15
10.1. Normative References 15
10.2. Informative References 16
Authors' Addresses 17

1. Requirements Notation

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Introduction

The network virtualization overlay over Layer 3 (NVO3) as depicted in Figure 1, allows an overlay cloud provider to provide a logical L2/L3 interconnect for the Tenant Systems TSes that belong to a specific tenant network. A packet received from a TS is encapsulated by the ingress Network Virtualization Edge (NVE). The encapsulated packet is then sent to the remote NVE through a tunnel. When reaching the egress NVE of the tunnel, the packet is decapsulated and forwarded to the target TS. The L2/L3 address mappings to the remote NVE(s) are distributed to the NVEs by a logically centralized Network Virtualization Authority (NVA) or using a distributed control plane such as Ethernet-VPN. In a datacenter, the NVO3 tunnels can be implemented using Generic Network Virtualization Encapsulation (Geneve) [I-D.ietf-nvo3-geneve]. Such Geneve tunnels establish NVE-to-NVE communications, may transit within the data center via Transit device. The Geneve tunnels overlay network enable multiple Virtual Networks to coexist over a shared underlay infrastructure, and a Virtual Network may span a single data center or multiple data centers.

The underlay infrastructure on which the multi-tenancy overlay networks are hosted, can be owned and provided by an underlay provider who may be different from the overlay cloud provider.

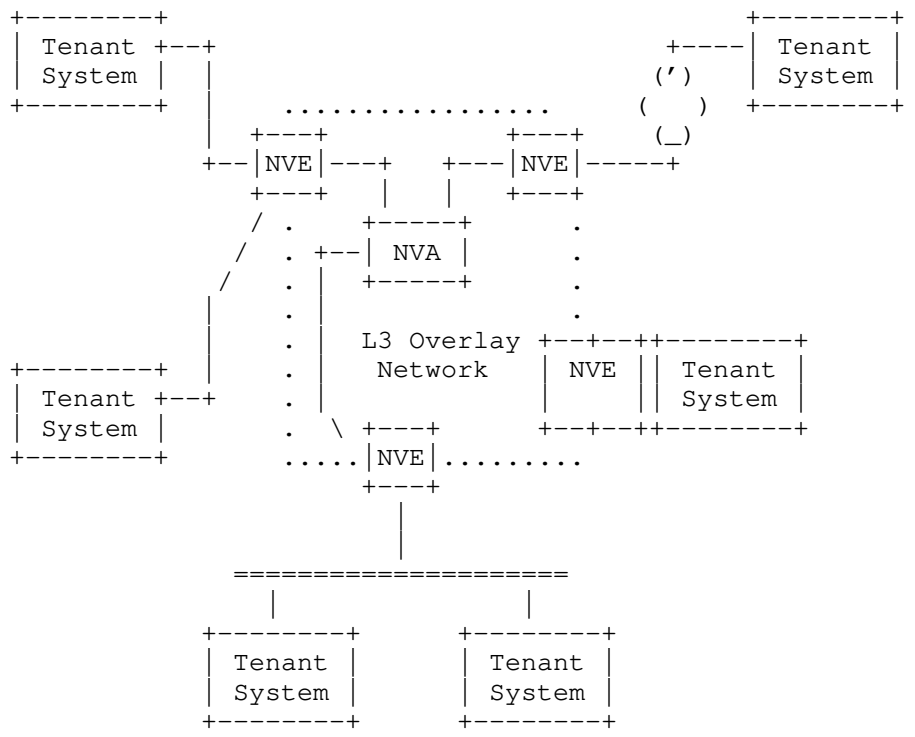


Figure 1: Generic Reference Model for Network Virtualization Overlays [RFC7365]

This document discusses the security risks that a Geneve based NVO3 network may encounter. In addition, this document lists the requirements to protect the Geneve packet components defined in [I-D.ietf-nvo3-geneve] that include the Geneve tunnel IP and UDP header, the Geneve Header, Geneve options, and inner payload.

The document provides two sets of security requirements:

1. SEC-OP: requirements to evaluate a given deployment of Geneve overlay. Such requirements are intended to Geneve overlay provider to evaluate a given deployment. Security of the Geneve packet may be achieved using various mechanisms. Typically, some deployments may use a limited subset of the capabilities provided by Geneve and rely on specific assumptions. Given these specificities, the secure deployment of a given Geneve deployment may be achieved reusing specific mechanisms such as for example DTLS [RFC6347] or IPsec [RFC4301]. On the other hand, the definition of a security mechanisms that enables to secure any Geneve deployment requires the design of a Geneve specific

mechanism. Note that the security s limited to the security of the data plane only. Additional requirements for the control plan MAY be considered in [I-D.ietf-nvo3-security-requirements]. A given Geneve deployment will be considered secured when matching with all SEC-OP requirements does not raises any concern. As such the given deployment will be considered passing SEC-OP requirements that are not applicable.

2. SEC-GEN: requirements a security mechanism need to fulfill to secure any deployment of Geneve overlay deployment. Such mechanism may require the design of a specific solution. In the case new protocol needs to be design, the document strongly recommend to re-use existing security protocols like IP Security (IPsec) [RFC4301] and Datagram Transport Layer Security (DTLS) [RFC6347], and existing encryption algorithms (such as [RFC8221]), and authentication protocols. A given candidate for a security mechanism will be considered as valid when matching with all SEC-GEN requirements does not raise any concern. In other words, at least all MUST status are met.

This document assumes the following roles are involved: - Tenant: designates the entity that connects various systems within a single virtualized network. The various system can typically be containers, VMs implementing a single or various functions.
- Geneve Overlay Provider: provides the Geneve overlay that seamlessly connect the various Tenant Systems over a given virtualized network.
- Infrastructure Provider: provides the infrastructure that runs the Geneve overlay network as well as the Tenant System. A given deployment may consider different infrastructure provider with different level of trust. Typically the Geneve overlay network may use a public cloud to extend the resource of a private cloud. Similarly, a edge computing may extend its resources using resource of the core network.

Tenant, Geneve Overlay Provider and Infrastructure Provider can be implemented by a single or various different entities with different level of trust between each other. The simplest deployment may consists in a single entity running its systems in its data center and using Geneve in order to manage its internal resources. A more complex use case may consider that a Tenant subscribe to the Geneve Overlay Provider which manage the virtualized network over various type of infrastructure. The trust between the Tenant, Geneve Overlay Provider and Infrastructure Provider may be limited.

Given the different relations between Tenant, Geneve Overlay Provider and Infrastructure Provider, this document aims providing requirements to ensure: 1. The Geneve Overlay Provider delivers

tenant payload traffic (Geneve inner payload) and ensuring privacy and integrity. 2. The Geneve Overlay Provider provides the necessary means to prevent injection or redirection of the Tenant traffic from a rogue node in the Geneve overlay network or a rogue node from the infrastructure. 3. The Geneve Overlay Provider can rely on the Geneve overlay in term of robustness and reliability of the signaling associated to the Geneve packets (Geneve tunnel header, Geneve header and Geneve options) in order to appropriately manage its overlay.

3. Terminology

This document uses the terminology of [RFC8014], [RFC7365] and [I-D.ietf-nvo3-geneve].

4. Security Threats

This section considers attacks performed by NVE, network devices or any other devices using Geneve, that is when the attackers knowing the details of the Geneve packets can perform their attacks by changing fields in the Geneve tunnel header, base header, Geneve options and Geneve inner payload. Attacks related to the control plane are outside the scope of this document. The reader is encouraged to read [I-D.ietf-nvo3-security-requirements] for a similar threat analysis of NVO3 overlay networks.

Threats include traffic analysis, sniffing, injection, redirection, and replay. Based on these threats, this document enumerates the security requirements.

Threats are divided into two categories: passive attack and active attack.

Threats are always associated with risks and the evaluation of these risks depend among other things on the environment.

4.1. Passive Attacks

Passive attacks include traffic analysis (noticing which workloads are communicating with which other workloads, how much traffic, and when those communications occur) and sniffing (examining traffic for useful information such as personally-identifyable information or protocol information (e.g., TLS certificate, overlay routing protocols).

Passive attacks may also consist in inferring information about a virtualized network or some Tenant System from observing the Geneve traffic. This could also involve the correlation between observed

traffic and additional information. For example, a passive network observer can determine two virtual machines are communicating by manipulating activity or network activity of other virtual machines on that same host. For example, the attacker could control (or be otherwise aware of) network activity of the other VMs running on the same host, and deduce other network activity is due to a victim VM.

A rogue element of the overlay Geneve network under the control of an attacker may leak and redirect the traffic from a virtual network to the attacker for passive monitoring [RFC7258].

Avoiding leaking information is hard to enforced. The security requirements provided in section {{sniffing} expect to mitigate such attacks by lowering the consequences, typically making leaked data unusable to an attacker.

4.2. Active Attacks

Active attacks involve modifying Geneve packets, injecting Geneve packets, or interfering with Geneve packet delivery (such as by corrupting packet checksum). Active attack may target the Tenant System or the Geneve overlay.

There are multiple motivations to inject illegitimate traffic into a tenants network. When the rogue element is on the path of the TS traffic, it may be able to inject and receive the corresponding messages back. On the other hand, if the attacker is not on the path of the TS traffic it may be limited to only inject traffic to a TS without receiving any response back. When rogue element have access to the traffic in both directions, the possibilities are only limited by the capabilities of the other on path elements - Transit device, NVE or TS - to detect and protect against the illegitimate traffic. On the other hand, when the rogue element is not on path, the surface for such attacks remains still quite large. For example, an attacker may target a specific TS or application by crafting a specific packet that can either generate load on the system or crash the system or application. TCP syn flood typically overload the TS while not requiring the ability to receive responses. Note that udp application are privileged target as they do not require the establishment of a session and are expected to treat any incoming packets.

Traffic injection may also be used to flood the virtual network to disrupt the communications between the TS or to introduce additional cost for the tenant, for example when pricing considers the traffic inside the virtual network. The two latest attacks may also take advantage of applications with a large factor of amplification for their responses as well as applications that upon receiving a packet

interact with multiple TS. Similarly, applications running on top of UDP are privileged targets.

Note also that an attacker that is not able to receive the response traffic, may use other channels to evaluate or measure the impact of the attack. Typically, in the case of a service, the attacker may have access, for example, to a user interface that provides indication on the level of disruption and the success of an attack, Such feed backs may also be used by the attacker to discover or scan the network.

Preventing traffic to cross virtual networks, reduce the surface of attack, but rogue element main still perform attacks within a given virtual network by replaying a legitimate packet. Some variant of such attack also includes modification of unprotected parts when available in order for example to increase the payload size.

5. Requirements for Security Mitigations

The document assumes that Security protocols, algorithms, and implementations provide the security properties for which they are designed, an attack caused by a weakness in a cryptographic algorithm is out of scope.

Protecting network connecting TSes and NVEs which could be accessible to outside attackers is out of scope.

An attacker controlling an underlying network device may break the communication of the overlays by discarding or delaying the delivery of the packets passing through it. The security consideration to prevent this type of attack is out of scope of this document.

Securing communication between NVAs and NVEs is out of scope.

Selectively providing integrity / authentication, confidentiality / encryption of only portions of the Geneve packet is in scope. This will be the case if the Tenant Systems uses security protocol to protect its communications.

5.1. Protection Against Traffic Sniffing

The inner payload, unless protection is provided by the Tenant System reveals the content of the communication. This may mitigate by the Tenant using application level security such as, for example JSON Web Encryption [RFC7516] or transport layer security such as DTLS [RFC6347] or TLS [RFC8446] or IPsec/ESP [RFC4303]. However none of these security protocols are sufficient to protect the entire inner payload. IPsec/ESP still leave in clear the optional L2 layer

information as well as the IP addresses and some IP options. In addition to these pieces of information, the use of TLS or DTLS reveals the transport layer protocol as well as ports.

A secure deployment of a Geneve overlay must fulfill the requirement below:

1. SEC-OP-1: A secure deployment of a Geneve overlay SHOULD by default encrypt the inner payload. A Geneve overlay provider MAY disable this capability for example when encryption is performed by the Tenant System and that level of confidentiality is believed to be sufficient. In order to provide additional protection to traffic already encrypted by the Tenant the Geneve network operator MAY partially encrypt the clear part of the inner payload.

A Geneve security mechanism must fulfill the requirements below:

- SEC-GEN-1: Geneve security mechanism MUST provide the capability to encrypt the inner payload.

- SEC-GEN-2: Geneve security mechanism SHOULD provide the capability to partially encrypt the inner payload header.

The Geneve Header and Geneve Options contains metadata information related to the communications. Note that a Geneve packet may have a combination of Geneve options that needs to be read by transit device, in which case this option needs to be read by the transit device while other options MAY only be accessed by the tunnel endpoint. Information revealed as well as correlation with traffic volumetry may reveal pattern traffic within a given virtualized network as well as any information revealed by the current and future Geneve Option.

A secure deployment of a Geneve overlay must fulfill the requirement below:

- SEC-OP-2: A secure deployment of a Geneve overlay MUST evaluate the information associated to the leakage of the Geneve Outer Header, Geneve Header and Geneve Option. When a risk analysis concludes that the risk of leaking sensitive information is too high, such MUST NOT be transmit in clear text.

- SEC-OP-3: A secure deployment of a Geneve overlay MUST evaluate the risk associated to traffic pattern recognition. When a risk has been identified, traffic pattern recognition MUST be addressed with padding policies as well as generation of dummy packets.

A Geneve security mechanism must fulfill the requirements below:

- SEC-GEN-3: Geneve security mechanism MUST provide the capability to encrypt a single or a set of options while leave other Geneve Option in clear. Reversely, a Geneve security mechanism MUST be able to leave a Geneve option in clear, while encrypting the others.

- SEC-GEN-4: Geneve security mechanism MUST provide means to encrypt the information of Geneve Header. Reversely, a Geneve security mechanism MUST be able to leave in clear header information while encrypting the other.

- SEC-GEN-5: Geneve security mechanism MUST provide the ability to pad a Geneve packet.

- SEC-GEN-6: Geneve security mechanism MUST provide the ability to send dummy packets.

5.2. Protecting Against Traffic Injection

Traffic injection from a rogue non legitimate NVO3 Geneve overlay device or a rogue underlay transit device can target an NVE, a transit underlay device or a Tenant System. Targeting a Tenant's System requires a valid MAC and IP addresses of the Tenant's System.

Tenant's System may protect their communications using IPsec or TLS. Such protection protects the Tenants from receiving spoofed packets, as any injected packet is expected to be discarded by the destination Tenant's System. Such protection does not protect the tenant system from receiving illegitimate packets that may disrupt the Tenant's System performance. The Geneve overlay network MAY still need to prevent such spoofed Tenant's system packets from being steered to the Tenant's system. When the Tenant's Systems are not protecting their communications, the Geneve overlay network SHOULD be able to to prevent a rogue device from injecting traffic into the overlay network.

In order to prevent traffic injection to one virtual network, the destination legitimate Geneve NVE MUST be able to authenticate the incoming Geneve packets from the source NVE. The Geneve architecture considers transit devices that MAY process some Geneve Option without affecting the Geneve packet. These transit device MAY Authenticate the Geneve packet as part of the Geneve packet processing but MAY also process other Geneve options. As a result, integrity protection and authentication SHOULD be performed by transit device, prior to any processing.

A secure deployment of a Geneve overlay must fulfill the requirement below:

- SEC-OP-4: A secure deployment of a Geneve overlay SHOULD authenticate communications between NVE to protect the Geneve Overlay infrastructure as well as the Tenants System's communications (Geneve Packet). A Geneve overlay provider MAY disable authentication of the inner packet and delegates it to the Tenant Systems when communications between Tenant's System is secured. This is NOT RECOMMENDED. To prevent injection between virtualized network, it is strongly RECOMMENDED that at least the Geneve Header is authenticated.

- SEC-OP-5: A secure deployment of a Geneve overlay SHOULD NOT process data prior authentication. If that is not possible, the Geneve overlay provider SHOULD evaluate its impact.

A Geneve security mechanism must fulfill the requirements below:

- SEC-GEN-8: Geneve Security mechanism MUST provide means for a tunnel endpoint (NVE) to authenticate data prior it is being processed. A tunnel endpoint (NVE) MUST be able to authenticate at least:

  * the Geneve Header and a subset of Geneve Options

  * the Geneve Header, a subset of Geneve options and the Geneve inner payload

  * the Geneve Header, a subset of Geneve options and the Geneve inner payload or the portion of the inner payload in case the Tenant's System provides some authentication mechanism.

- SEC-GEN-9: Geneve Security mechanism SHOULD provide means for a transit device to authenticate the Geneve Option prior processing it. Authentication MAY concern the whole Geneve packet, but MAY be limited to the Geneve Option.

5.3. Protecting Against Traffic Redirection

A rogue device of the NVO3 overlay Geneve network or the underlay network may redirect the traffic from a virtual network to the attacker for passive or active attacks. If the rogue device is in charge of the securing the Geneve packet, then Geneve security mechanisms are not intended to address this threat. More specifically, a rogue source NVE will still be able to redirect the traffic in clear text before protecting (and encrypting the packet). A rogue destination NVE will still be able to redirect the traffic in

clear text after decrypting the Geneve packets. The same occurs with a rogue transit that is in charge of encrypting and decrypting a Geneve Option, Geneve Option or any information. The security mechanisms are intended to protect a Geneve information from any on path node. Note that modern cryptography recommend the use of authenticated encryption. This section assumes such algorithms are used, and as such encrypted packets are also authenticated.

To prevent an attacker located in the middle between the NVEs and modifying the tunnel address information in the data packet header to redirect the data traffic, the solution need to provide confidentiality protection for data traffics exchanged between NVEs.

Requirements are similar as those provided in section Section 5.1 to mitigate sniffing attacks and those provided in section Section 5.2 to mitigate traffic injection attacks.

5.4. Protecting Against Traffic Replay

A rogue device of the NVO3 overlay Geneve network or the underlay network may replay a Geneve packet, to load the network and/or a specific Tenant System with a modified Geneve payload. In some cases, such attacks may target an increase of the tenants costs.

When traffic between tenants is not protected, the rogue device may forward the modified packet over a valid (authenticated) Geneve Header. The crafted packet may for example, include a specifically crafted application payload for a specific Tenant Systems application, with the intention to load the tenant specific application.

Updating the Geneve header and option parameters such as setting an OAM bit, adding bogus option TLVs, or setting a critical bit, may result in different processing behavior, that could greatly impact performance of the overlay network and the underlay infrastructure and thus affect the tenants traffic delivery.

The NVO3 overlay network and underlay network nodes that may address such attacks MUST provide means to authenticate the Geneve packet components.

A secure deployment of a Geneve overlay must fulfill the requirement below:

- SEC-OP-6: A secure deployment of a Geneve overlay MUST evaluate the communications subject to replay attacks. Communications that are subject to this attacks MUST be authenticated with an anti replay mechanism. Note that when partial authentication is

provided, the part not covered by the authentication remains a surface of attack. It is strongly RECOMMENDED that the Geneve Header is both authenticated with anti replay protection.

A Geneve security mechanism must fulfill the requirements below:

- SEC-GEN-10: Geneve Security mechanism MUST provide means for a tunnel endpoint (NVE) to validate the Geneve Header corresponds to the Geneve payload, and discard such packets.

5.5. Security Management

A secure deployment of a Geneve overlay must fulfill the requirement below:

- SEC-OP-7: A secure deployment of a Geneve overlay MUST define the security policies that associates the encryption, and authentication associated to each flow between NVEs.

- SEC-OP-8: A secure deployment of a Geneve overlay SHOULD define distinct material for each flow. The cryptographic depends on the nature of the flow (multicast, unicast) as well as on the security mechanism enabled to protect the flow.

A Geneve security mechanism must fulfill the requirements below:

- SEC-GEN-11: A Geneve security mechanism MUST be managed via security policies associated for each traffic flow to be protected. Geneve overlay provider MUST be able to configure NVEs with different security policies for different flows. A flow MUST be identified at minimum by the Geneve virtual network identifier and the inner IP and transport headers, and optionally additional fields which define a flow (e.g., inner IP DSCP, IPv6 flow id, Geneve options).

- SEC-GEN-12: A Geneve security mechanism MUST be able to assign different cryptographic keys to protect the unicast tunnels between NVEs respectively.

- SEC-GEN-13: A Geneve security mechanisms, when multicast is used, packets,MUST be able to assign distinct cryptographic group keys to protect the multicast packets exchanged among the NVEs within different multicast groups. Upon receiving a data packet, an egress Geneve NVE MUST be able to verify whether the packet is sent from a proper ingress NVE which is authorized to forward that packet.

6. IANA Considerations

There are no IANA consideration for this document.

7. Security Considerations

The whole document is about security.

Limiting the coverage of the authentication / encryption provides some means for an attack to craft special packets.

The current document details security requirements that are related to the Geneve protocol. Instead, [I-D.ietf-nvo3-security-requirements] provides generic architecture security requirement upon the deployment of an NVO3 overlay network. It is strongly recommended to read that document as architecture requirements also apply here. In addition, architecture security requirements go beyond the scope of Geneve communications, and as such are more likely to address the security needs upon deploying an Geneve overlay network.

8. Appendix

8.1. TLS

This section compares how NVE communications using TLS meet the security requirements for a secure Geneve overlay deployment. In this example TLS is used over the Geneve Outer Header and secured the Geneve Header, Geneve Options and the inner payload.

The use of TLS MAY fill the security requirements for a secure Geneve deployment. However TLS cannot be considered as the Geneve security mechanism enabling all Geneve deployments.

The use of to secure a Geneve overlay deployment TLS meets SEC-OP-1 as it protects the inner payload of the tenant. It meets SEC-OP-2 as except from the UDP port, no information concerning Geneve is leaked. SEC-OP-3 is not met as TLS does not provide the ability to send dummy traffic, nor to pad. SEC-OP-4 is met as the communication is authenticated, including the Geneve Header. SEC-OP-5 is met as the Geneve Packet is processed once it has been authenticated. SEC-OP-6 is met as TLS comes with anti replay protection. SEC-OP-7 and SEC-OP-8 may also be met with security policies established per UDP destination port where only unicast is considered.

The use of TLS as a generic Geneve Security mechanism meets SEC-GEN-1 as it encrypts the inner payload. However, TLS, but does not enable partial encryption of the inner payload. TLS does not meet SEC-GEN3

or SEC-GEN-4 that requires the ability to encrypt of a subset of the Geneve Options or the Geneve Header information. In addition, TLS does not enable that some Geneve option of Header information remain in clear text while other are encrypted. Typically TLS would not be compatible with transit device. In addition is make the Geneve option visible to the transit device, TLS does not provide the ability for a transit device to authenticate the option before processing it. SEC-GEN-5 and SEC-GEN-6 are not met as TLS does not provide padding nor the ability to generate dummy packets. TLS does not meet SEC-GEN-8 that requires the ability to authenticate some combination of Geneve Header, Geneve Options, (partial) inner payload. TLS does not meet SEC-GEN-9 that requires the ability to authenticate a single Geneve Option. TLS meets SEC-GEN-10 as it provides anti replay mechanism to the authentication. SEC-GEN-11 is not natively supported as TLS security is established by UDP destination ports, rather than by flow. If more than one security policy or flow needs to be considered a binding between flow and ports needs to be established. SEC-GEN-13 is not met for mutlicast traffic.

8.2. IPsec

The use of IPsec/ESP or IPsec/AH share most of the analysis performed for TLS. The main advantages of using IPsec would be that IPsec supports multicast communications and natively supports flow based security policies. However, the use of these security policies in a context of Geneve is not natively supported.

9. Acknowledgments

We would like to thank Ilango S Ganaga for its useful reviews and clarifications as well as Matthew Bocci, Sam Aldrin and Ignas Bagdona for moving the work forward.

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>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

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

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <https://www.rfc-editor.org/info/rfc7365>.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <https://www.rfc-editor.org/info/rfc7516>.

[RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <https://www.rfc-editor.org/info/rfc8014>.

[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>.

[RFC8221] Wouters, P., Migault, D., Mattsson, J., Nir, Y., and T. Kivinen, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 8221, DOI 10.17487/RFC8221, October 2017, <https://www.rfc-editor.org/info/rfc8221>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

10.2. Informative References

[I-D.ietf-nvo3-geneve]
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-08 (work in progress), October 2018.

[I-D.ietf-nvo3-security-requirements]
Hartman, S., Zhang, D., Wasserman, M., Qiang, Z., and M. Zhang, "Security Requirements of NVO3", draft-ietf-nvo3-security-requirements-07 (work in progress), June 2016.

Authors' Addresses

Daniel Migault
Ericsson
8275 Trans Canada Route
Saint Laurent, QC 4S 0B6
Canada

EMail: daniel.migault@ericsson.com

Sami Boutros
VMware, Inc.

EMail: boutros@vmware.com<

Dan Wings
VMware, Inc.

EMail: dwing@vmware.com

Suresh Krishnan
Kaloom

EMail: suresh@kaloom.com

NVO3 Workgroup
Internet Draft
Intended status: Informational

J. Rabadan, Ed.
M. Bocci
Nokia

S. Boutros
WMware

A. Sajassi
Cisco

Expires: August 13, 2018

February 9, 2018

Applicability of EVPN to NVO3 Networks
draft-rabadan-nvo3-evpn-applicability-01

Abstract

In NVO3 networks, Network Virtualization Edge (NVE) devices sit at the edge of the underlay network and provide Layer-2 and Layer-3 connectivity among Tenant Systems (TSes) of the same tenant. The NVEs need to build and maintain mapping tables so that they can deliver encapsulated packets to their intended destination NVE(s). While there are different options to create and disseminate the mapping table entries, NVEs may exchange that information directly among themselves via a control-plane protocol, such as EVPN. EVPN provides an efficient, flexible and unified control-plane option that can be used for Layer-2 and Layer-3 Virtual Network (VN) service connectivity. This document describes the applicability of EVPN to NVO3 networks and how EVPN solves the challenges in those networks.

This Internet-Draft will expire on August 13, 2018.

Table of Contents

1. Introduction 3
2. EVPN and NVO3 Terminology 3
3. Why Is EVPN Needed In NVO3 Networks? 6
4. Applicability of EVPN to NVO3 Networks 8
4.1. EVPN Route Types used in NVO3 Networks 8
4.2. EVPN Basic Applicability For Layer-2 Services 9
4.2.1. Auto-Discovery and Auto-Provisioning of ES, Multi-Homing PEs and NVE services 10
4.2.2. Remote NVE Auto-Discovery 11
4.2.3. Distribution Of Tenant MAC and IP Information 12
4.3. EVPN Basic Applicability for Layer-3 Services 13
4.4. EVPN as a Control Plane for NVO3 Encapsulations and GENEVE 15
4.5. EVPN OAM and application to NVO3 15
4.6. EVPN as the control plane for NVO3 security 16
4.7. Advanced EVPN Features For NVO3 Networks 16
4.7.1. Virtual Machine (VM) Mobility 16
4.7.2. MAC Protection, Duplication Detection and Loop Protection 16
4.7.3. Reduction/Optimization of BUM Traffic In Layer-2 Services 17

4.7.4. Ingress Replication (IR) Optimization For BUM Traffic . 18
4.7.5. EVPN Multi-homing 18
4.7.6. EVPN Recursive Resolution for Inter-Subnet Unicast Forwarding 19
4.7.7. EVPN Optimized Inter-Subnet Multicast Forwarding . . . 21
4.7.8. Data Center Interconnect (DCI) 21
5. Conclusion 22
6. Conventions used in this document 22
7. Security Considerations 22
8. IANA Considerations 22
9. References 22
9.1 Normative References 23
9.2 Informative References 23
10. Acknowledgments 25
11. Contributors 25
12. Authors' Addresses 25

1. Introduction

In NVO3 networks, Network Virtualization Edge (NVE) devices sit at the edge of the underlay network and provide Layer-2 and Layer-3 connectivity among Tenant Systems (TSes) of the same tenant. The NVEs need to build and maintain mapping tables so that they can deliver encapsulated packets to their intended destination NVE(s). While there are different options to create and disseminate the mapping table entries, NVEs may exchange that information directly among themselves via a control-plane protocol, such as EVPN. EVPN provides an efficient, flexible and unified control-plane option that can be used for Layer-2 and Layer-3 Virtual Network (VN) service connectivity.

In this document, we assume that the EVPN control-plane module resides in the NVEs. The NVEs can be virtual switches in hypervisors, TOR/Leaf switches or Data Center Gateways. Note that Network Virtualization Authorities (NVAs) may be used to provide the forwarding information to the NVEs, and in that case, EVPN could be used to disseminate the information across multiple federated NVAs. The applicability of EVPN would then be similar to the one described in this document. However, for simplicity, the description assumes control-plane communication among NVE(s).

2. EVPN and NVO3 Terminology

o EVPN: Ethernet Virtual Private Networks, as described in [RFC7432].

- PE: Provider Edge router.

- NVO3 or Overlay tunnels: Network Virtualization Over Layer-3 tunnels. In this document, NVO3 tunnels or simply Overlay tunnels will be used interchangeably. Both terms refer to a way to encapsulate tenant frames or packets into IP packets whose IP Source Addresses (SA) or Destination Addresses (DA) belong to the underlay IP address space, and identify NVEs connected to the same underlay network. Examples of NVO3 tunnel encapsulations are VXLAN [RFC7348], [GENEVE] or MPLSoUDP [RFC7510].

- VXLAN: Virtual eXtensible Local Area Network, an NVO3 encapsulation defined in [RFC7348].

- GENEVE: Generic Network Virtualization Encapsulation, an NVO3 encapsulation defined in [GENEVE].

- CLOS: a multistage network topology described in [CLOS1953], where all the edge switches (or Leafs) are connected to all the core switches (or Spines). Typically used in Data Centers nowadays.

- ECMP: Equal Cost Multi-Path.

- NVE: Network Virtualization Edge is a network entity that sits at the edge of an underlay network and implements L2 and/or L3 network virtualization functions. The network-facing side of the NVE uses the underlying L3 network to tunnel tenant frames to and from other NVEs. The tenant-facing side of the NVE sends and receives Ethernet frames to and from individual Tenant Systems. In this document, an NVE could be implemented as a virtual switch within a hypervisor, a switch or a router, and runs EVPN in the control-plane.

- EVI: or EVPN Instance. It is a Layer-2 Virtual Network that uses an EVPN control-plane to exchange reachability information among the member NVEs. It corresponds to a set of MAC-VRFs of the same tenant. See MAC-VRF in this section.

- BD: or Broadcast Domain, it corresponds to a tenant IP subnet. If no suppression techniques are used, a BUM frame that is injected in a BD will reach all the NVEs that are attached to that BD. An EVI may contain one or multiple BDs depending on the service model [RFC7432]. This document will use the term BD to refer to a tenant subnet.

- EVPN VLAN-based service model: it refers to one of the three service models defined in [RFC7432]. It is characterized as a BD that uses a single VLAN per physical access port to attach tenant traffic to the BD. In this service model, there is only one BD per

EVI.

- EVPN VLAN-bundle service model: similar to VLAN-based but uses a bundle of VLANs per physical port to attach tenant traffic to the BD. As in VLAN-based, in this model there is a single BD per EVI.

- EVPN VLAN-aware bundle service model: similar to the VLAN-bundle model but each individual VLAN value is mapped to a different BD. In this model there are multiple BDs per EVI for a given tenant. Each BD is identified by an "Ethernet Tag", that is a control-plane value that identifies the routes for the BD within the EVI.

- IP-VRF: an IP Virtual Routing and Forwarding table, as defined in [RFC4364]. It stores IP Prefixes that are part of the tenant's IP space, and are distributed among NVEs of the same tenant by EVPN. Route-Distinghisher (RD) and Route-Target(s) (RTs) are required properties of an IP-VRF. An IP-VRF is instantiated in an NVE for a given tenant, if the NVE is attached to multiple subnets of the tenant and local inter-subnet-forwarding is required across those subnets.

- MAC-VRF: a MAC Virtual Routing and Forwarding table, as defined in [RFC7432]. The instantiation of an EVI (EVPN Instance) in an NVE. Route-distinghisher (RD) and Route-Target(s) (RTs) are required properties of a MAC-VRF and they are normally different than the ones defined in the associated IP-VRF (if the MAC-VRF has an IRB interface).

- BT: a Bridge Table, as defined in [RFC7432]. A BT is the instantiation of a BD in an NVE. When there is a single BD on a given EVI, the MAC-VRF is equivalent to the BT on that NVE.

- AC: Attachment Circuit or logical interface associated to a given BT. To determine the AC on which a packet arrived, the NVE will examine the physical/logical port and/or VLAN tags (where the VLAN tags can be individual c-tags, s-tags or ranges of both).

- IRB: Integrated Routing and Bridging interface. It refers to the logical interface that connects a BD instance (or a BT) to an IP-VRF and allows to forward packets with destination in a different subnet.

- ES: Ethernet Segment. When a Tenant System (TS) is connected to one or more NVEs via a set of Ethernet links, then that set of links is referred to as an 'Ethernet segment'. Each ES is represented by a unique Ethernet Segment Identifier (ESI) in the NVO3 network and the ESI is used in EVPN routes that are specific to that ES.

- o DF and NDF: they refer to Designated Forwarder and Non-Designated Forwarder, which are the roles that a given PE can have in a given ES.

- o VNI: Virtual Network Identifier. Irrespective of the NVO3 encapsulation, the tunnel header always includes a VNI that is added at the ingress NVE (based on the mapping table lookup) and identifies the BT at the egress NVE. This VNI is called VNI in VXLAN or GENEVE, VSID in nvGRE or Label in MPLSoGRE or MPLSoUDP. This document will refer to VNI as a generic Virtual Network Identifier for any NVO3 encapsulation.

- o BUM: Broadcast, Unknown unicast and Multicast frames.

- o SA and DA: they refer to Source Address and Destination Address. They are used along with MAC or IP, e.g. IP SA or MAC DA.

- o RT and RD: they refer to Route Target and Route Distinguisher.

- o PTA: Provider Multicast Service Interface Tunnel Attribute.

- o RT-1, RT-2, RT-3, etc.: they refer to Route Type followed by the type number as defined in the IANA registry for EVPN route types.

- o TS: Tenant System.

- o ARP and ND: they refer to Address Resolution Protocol and Neighbor Discovery protocol.

3. Why Is EVPN Needed In NVO3 Networks?

Data Centers have adopted NVO3 architectures mostly due to the issues discussed in [RFC7364]. The architecture of a Data Center is nowadays based on a CLOS design, where every Leaf is connected to a layer of Spines, and there is a number of ECMP paths between any two leaf nodes. All the links between Leaf and Spine nodes are routed links, forming what we also know as an underlay IP Fabric. The underlay IP Fabric does not have issues with loops or flooding (like old Spanning Tree Data Center designs did), convergence is fast and ECMP provides a fairly optimal bandwidth utilization on all the links.

On this architecture and as discussed by [RFC7364] multi-tenant intra-subnet and inter-subnet connectivity services are provided by NVO3 tunnels, being VXLAN [RFC7348] or [GENEVE] two examples of such tunnels.

Why is a control-plane protocol along with NVO3 tunnels required?

There are three main reasons:

a) Auto-discovery of the remote NVEs that are attached to the same VPN instance (Layer-2 and/or Layer-3) as the ingress NVE is.

b) Dissemination of the MAC/IP host information so that mapping tables can be populated on the remote NVEs.

c) Advanced features such as MAC Mobility, MAC Protection, BUM and ARP/ND traffic reduction/suppression, Multi-homing, Prefix Independent Convergence (PIC) like functionality, Fast Convergence, etc.

A possible approach to achieve points (a) and (b) above for multipoint Ethernet services, is "Flood and Learn". "Flood and Learn" refers to not using a specific control-plane on the NVEs, but rather "Flood" BUM traffic from the ingress NVE to all the egress NVEs attached to the same BD. The egress NVEs may then use data path MAC SA "Learning" on the frames received over the NVO3 tunnels. When the destination host replies back and the frames arrive at the NVE that initially flooded BUM frames, the NVE will also "Learn" the MAC SA of the frame encapsulated on the NVO3 tunnel. This approach has the following drawbacks:

- In order to Flood a given BUM frame, the ingress NVE must know the IP addresses of the remote NVEs attached to the same BD. This may be done as follows:
  - The remote tunnel IP addresses can be statically provisioned on the ingress NVE. If the ingress NVE receives a BUM frame for the BD on an ingress AC, it will do ingress replication and will send the frame to all the configured egress NVE IP DAs in the BD.
  - All the NVEs attached to the same BD can subscribe to an underlay IP Multicast Group that is dedicated to that BD. When an ingress NVE receives a BUM frame on an ingress AC, it will send a single copy of the frame encapsulated into an NVO3 tunnel, using the multicast address as IP DA of the tunnel. This solution requires PIM in the underlay network and the association of individual BDs to underlay IP multicast groups.
- "Flood and Learn" solves the issues of auto-discovery and learning of the MAC to VNI/tunnel IP mapping on the NVEs for a given BD. However, it does not provide a solution for advanced features and it does not scale well.

EVPN provides a unified control-plane that solves the NVE auto-

discovery, tenant MAP/IP dissemination and advanced features in a scalable way and keeping the independence of the underlay IP Fabric, i.e. there is no need to enable PIM in the underlay network and maintain multicast states for tenant BDs.

Section 4 describes how to apply EVPN to meet the control-plane requirements in an NVO3 network.

4. Applicability of EVPN to NVO3 Networks

This section discusses the applicability of EVPN to NVO3 networks. The intend is not to provide a comprehensive explanation of the protocol itself but give an introduction and point at the corresponding reference document, so that the reader can easily find more details if needed.

4.1. EVPN Route Types used in NVO3 Networks

EVPN supports multiple Route Types and each type has a different function. For convenience, Table 1 shows a summary of all the existing EVPN route types and its usage. We will refer to these route types as RT-x throughout the rest of the document, where x is the type number included in the first column of Table 1.

Type	Description	Usage
1	Ethernet Auto-Discovery	Multi-homing: Per-ES: Mass withdrawal Per-EVI: aliasing/backup
2	MAC/IP Advertisement	Host MAC/IP dissemination Supports MAC mobility and protection
3	Inclusive Multicast Ethernet Tag	NVE discovery and BUM flooding tree setup
4	Ethernet Segment	Multi-homing: ES auto-discovery and DF Election
5	IP Prefix	IP Prefix dissemination
6	Selective Multicast Ethernet Tag	Indicate interest for a multicast S,G or *,G
7	IGMP Join Synch	Multi-homing: S,G or *,G state synch
8	IGMP Leave Synch	Multi-homing: S,G or *,G leave synch
9	Per-Region I-PMSI A-D	BUM tree creation across regions
10	S-PMSI A-D	Multicast tree for S,G or *,G states
11	Leaf A-D	Used for responses to explicit tracking

Table 1 EVPN route types

4.2. EVPN Basic Applicability For Layer-2 Services

Although the applicability of EVPN to NVO3 networks spans multiple documents, EVPN's baseline specification is [RFC7432]. [RFC7432] allows multipoint layer-2 VPNs to be operated as [RFC4364] IP-VPNs, where MACs and the information to setup flooding trees are distributed by MP-BGP. Based on [RFC7432], [EVPN-OVERLAY] describes how to use EVPN to deliver Layer-2 services specifically in NVO3 Networks.

Figure 1 represents a Layer-2 service deployed with an EVPN BD in an NVO3 network.

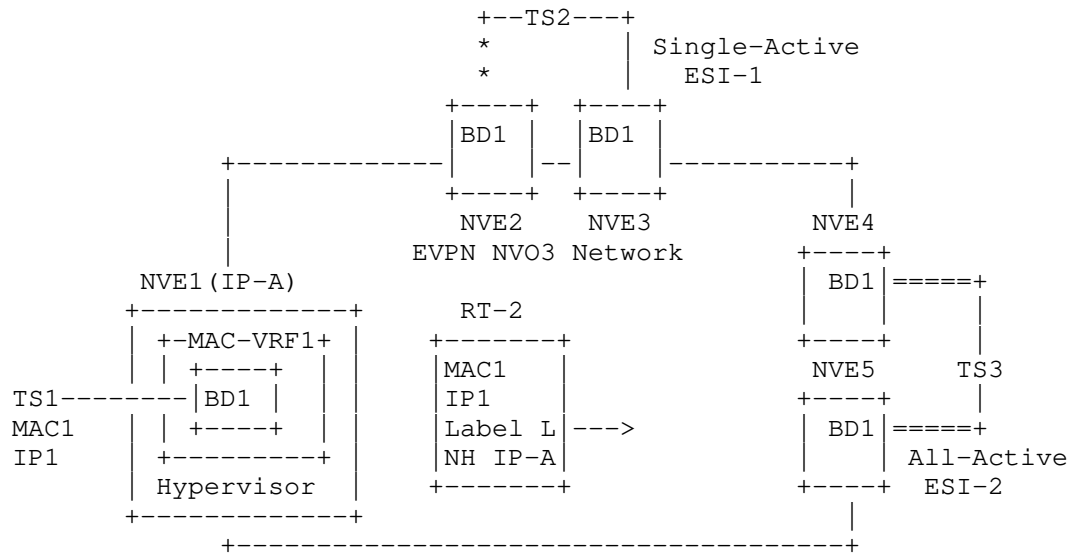


Figure 1 EVPN for L2 in an NVO3 Network - example

In a simple NVO3 network, such as the example of Figure 1, these are the basic constructs that EVPN uses for Layer-2 services (or Layer-2 Virtual Networks):

- o BD1 is an EVPN Broadcast Domain for a given tenant and TS1, TS2 and TS3 are connected to it. The five represented NVEs are attached to BD1 and are connected to the same underlay IP network. That is, each NVE learns the remote NVEs' loopback addresses via underlay routing protocol.

- o NVE1 is deployed as a virtual switch in a Hypervisor with IP-A as underlay loopback IP address. The rest of the NVEs in Figure 1 are physical switches and TS2/TS3 are multi-homed to them. TS1 is a virtual machine, identified by MAC1 and IP1.

4.2.1. Auto-Discovery and Auto-Provisioning of ES, Multi-Homing PEs and NVE services

Auto-discovery is one of the basic capabilities of EVPN. The provisioning of EVPN components in NVEs is significantly automated, simplifying the deployment of services and minimizing manual operations that are prone to human error.

These are some of the Auto-Discovery and Auto-Provisioning capabilities available in EVPN:

- o Automation on Ethernet Segments (ES): an ES is defined as a group of NVEs that are attached to the same TS or network. An ES is identified by an Ethernet Segment Identifier (ESI) in the control plane, but neither the ESI nor the NVEs that share the same ES are required to be manually provisioned in the local NVE:

  - If the multi-homed TS or network are running protocols such as LACP (Link Aggregation Control Protocol), MSTP (Multiple-instance Spanning Tree Protocol), G.8032, etc. and all the NVEs in the ES can listen to the protocol PDUs to uniquely identify the multi-homed TS/network, then the ESI can be "auto-sensed" or "auto-provisioned" following the guidelines in [RFC7432] section 5.

  - As described in [RFC7432], EVPN can also auto-derive the BGP parameters required to advertise the presence of a local ES in the control plane (RT and RD). Local ESes are advertised using RT-4s and the ESI-import Route-Target used by RT-4s can be auto-derived based on the procedures of [RFC7432], section 7.6.

  - By listening to other RT-4s that match the local ESI and import RT, an NVE can also auto-discover the other NVEs participating in the multi-homing for the ES.

  - Once the NVE has auto-discovered all the NVEs attached to the same ES, the NVE can automatically perform the DF Election algorithm (which determines the NVE that will forward traffic to the multi-homed TS/network). EVPN guarantees that all the NVEs in the ES have a consistent DF Election.

- o Auto-provisioning of services: when deploying a Layer-2 Service for a tenant in an NVO3 network, all the NVEs attached to the same subnet must be configured with a MAC-VRF and the BD for the subnet, as well as certain parameters for them. Note that, if the EVPN service model is VLAN-based or VLAN-bundle, implementations do not normally have a specific provisioning for the BD (since it is in that case the same construct as the MAC-VRF). EVPN allows auto-deriving as many MAC-VRF parameters as possible. As an example, the MAC-VRF's RT and RD for the EVPN routes may be auto-derived. Section 5.1.2.1 in [EVPN-OVERLAY] specifies how to auto-derive a MAC-VRF's RT as long as VLAN-based service model is implemented. [RFC7432] specifies how to auto-derive the RD.

4.2.2. Remote NVE Auto-Discovery

Auto-discovery via MP-BGP is used to discover the remote NVEs attached to a given BD, NVEs participating in a given redundancy group, the tunnel encapsulation types supported by an NVE, etc.

In particular, when a new MAC-VRF and BD are enabled, the NVE will advertise a new RT-3. Besides other fields, the RT-3 will encode the IP address of the advertising NVE, the Ethernet Tag (which is zero in case of VLAN-based and VLAN-bundle models) and also a PMSI Tunnel Attribute (PTA) that indicates the information about the intended way to deliver BUM traffic for the BD.

In the example of Figure 1, when MAC-VRF1/BD1 are enabled, NVE1 will send an RT-3 including its own IP address, Ethernet-Tag for BD1 and the PTA. Assuming Ingress Replication (IR), the RT-3 will include an identification for IR in the PTA and the VNI the NVEs must use to send BUM traffic to the advertising NVE. The other NVEs in the BD, will import the RT-3 and will add NVE1's IP address to the flooding list for BD1. Note that the RT-3 is also sent with a BGP encapsulation attribute [TUNNEL-ENCAP] that indicates what NVO3 encapsulation the remote NVEs should use when sending BUM traffic to NVE1.

Refer to [RFC7432] for more information about the RT-3 and forwarding of BUM traffic, and to [EVPN-OVERLAY] for its considerations on NVO3 networks.

4.2.3. Distribution Of Tenant MAC and IP Information

Tenant MAC/IP information is advertised to remote NVEs using RT-2s. Following the example of Figure 1:

- In a given EVPN BD, TSes' MAC addresses are first learned at the NVE they are attached to, via data path or management plane learning. In Figure 1 we assume NVE1 learns MAC1/IP1 in the management plane (for instance, via Cloud Management System) since the NVE is a virtual switch. NVE2, NVE3, NVE4 and NVE4 are TOR/Leaf switches and they normally learn MAC addresses via data path.

- Once NVE1's BD1 learns MAC1/IP1, NVE1 advertises that information along with a VNI and Next Hop IP-A in an RT-2. The EVPN routes are advertised using the RD/RTs of the MAC-VRF where the BD belongs. All the NVEs in BD1 learn local MAC/IP addresses and advertise them in RT-2 routes in a similar way.

- The remote NVEs can then add MAC1 to their mapping table for BD1 (BT). For instance, when TS3 sends frames to NVE4 with MAC DA = MAC1, NVE4 does a MAC lookup on the BT that yields IP-A and Label L. NVE4 can then encapsulate the frame into an NVO3 tunnel with IP-A as the tunnel IP DA and L as the Virtual Network Identifier. Note that the RT-2 may also contain the host's IP address (as in the example of Figure 1). While the MAC of the received RT-2 is

installed in the BT, the IP address may be installed in the Proxy-ARP/ND table (if enabled) or in the ARP/IP-VRF tables if the BD has an IRB. See section 4.7.3. to see more information about Proxy-ARP/ND and section 4.3. for more details about IRB and Layer-3 services.

Refer to [RFC7432] and [EVPN-OVERLAY] for more information about the RT-2 and forwarding of known unicast traffic.

4.3. EVPN Basic Applicability for Layer-3 Services

[IP-PREFIX] and [INTER-SUBNET] are the reference documents that describe how EVPN can be used for Layer-3 services. Inter Subnet Forwarding in EVPN networks is implemented via IRB interfaces between BDs and IP-VRFs. As discussed, an EVPN BD corresponds to an IP subnet. When IP packets generated in a BD are destined to a different subnet (different BD) of the same tenant, the packets are sent to the IRB attached to local BD in the source NVE. As discussed in [INTER-SUBNET], depending on how the IP packets are forwarded between the ingress NVE and the egress NVE, there are two forwarding models: Asymmetric and Symmetric.

The Asymmetric model is illustrated in the example of Figure 2 and it requires the configuration of all the BDs of the tenant in all the NVEs attached to the same tenant. In that way, there is no need to advertise IP Prefixes between NVEs since all the NVEs are attached to all the subnets. It is called Asymmetric because the ingress and egress NVEs do not perform the same number of lookups in the data plane. In Figure 2, if TS1 and TS2 are in different subnets, and TS1 sends IP packets to TS2, the following lookups are required in the data path: a MAC lookup (on BD1's table), an IP lookup (on the IP-VRF) and a MAC lookup (on BD2's table) at the ingress NVE1 and then only a MAC lookup at the egress NVE. The two IP-VRFs in Figure 2 are not connected by tunnels and all the connectivity between the NVEs is done based on tunnels between the BDs.

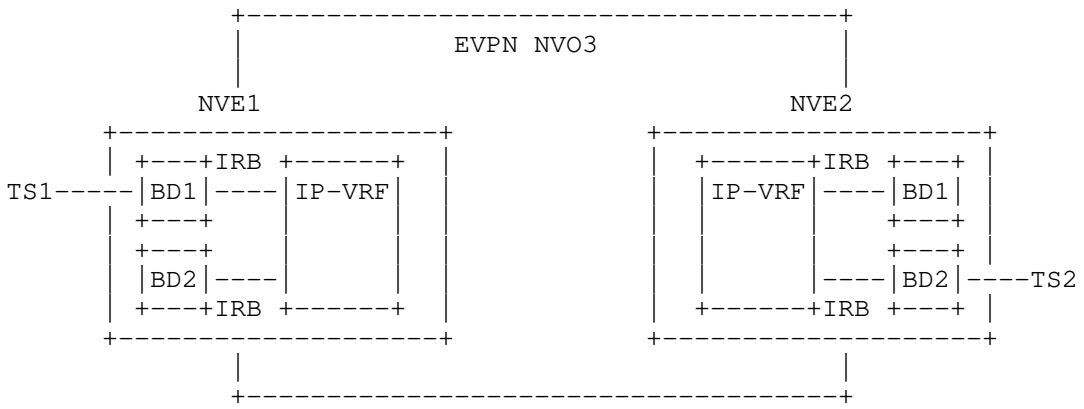


Figure 2 EVPN for L3 in an NVO3 Network - Asymmetric model

In the Symmetric model, depicted in Figure 3, there are the same data path lookups at the ingress and egress NVEs. For example, if TS1 sends IP packets to TS3, the following data path lookups are required: a MAC lookup at NVE1's BD1 table, an IP lookup at NVE1's IP-VRF and then IP lookup and MAC lookup at NVE2's IP-VRF and BD3 respectively. In the Symmetric model, the Inter Subnet connectivity between NVEs is done based on tunnels between the IP-VRFs.

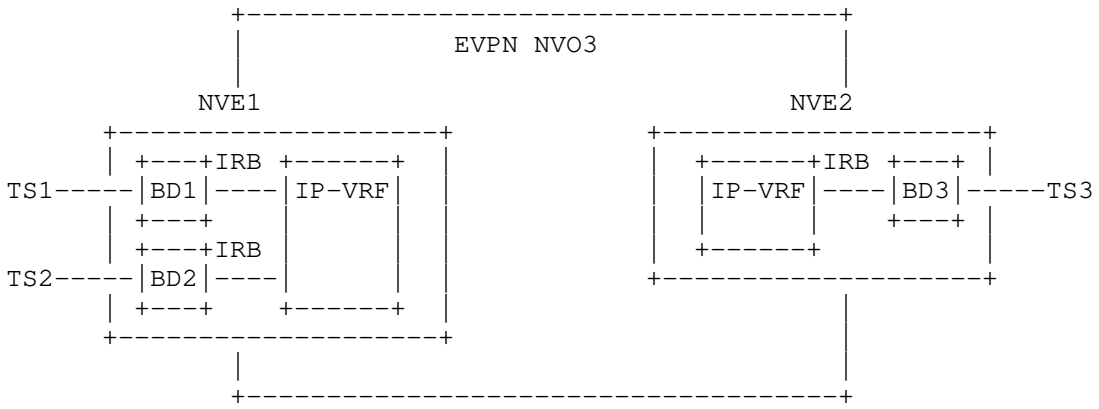


Figure 3 EVPN for L3 in an NVO3 Network - Symmetric model

The Symmetric model scales better than the Asymmetric model because it does not require the NVEs to be attached to all the tenant's subnets. However, it requires the use of NVO3 tunnels on the IP-VRFs

and the exchange of IP Prefixes between the NVEs in the control plane. EVPN uses RT-2 and RT-5 routes for the exchange of host IP routes (in the case of RT-2 and RT-5) and IP Prefixes (RT-5s) of any length. As an example, in Figure 3, NVE2 needs to advertise TS3's host route and/or TS3's subnet, so that the IP lookup on NVE1's IP-VRF succeeds.

[INTER-SUBNET] specifies the use of RT-2s for the advertisement of host routes. Section 4.4.1 in [IP-PREFIX] specifies the use of RT-5s for the advertisement of IP Prefixes in an "Interface-less IP-VRF-to-IP-VRF Model".

4.4. EVPN as a Control Plane for NVO3 Encapsulations and GENEVE

[EVPN-OVERLAY] describes how to use EVPN for NVO3 encapsulations, such us VXLAN, nvGRE or MPLSoGRE. The procedures can be easily applicable to any other NVO3 encapsulation, in particular GENEVE.

The NVO3 working group has been working on different data plane encapsulations. The Generic Network Virtualization Encapsulation [GENEVE] has been recommended to be the proposed standard for NVO3 Encapsulation. The EVPN control plane can signal the GENEVE encapsulation type in the BGP Tunnel Encapsulation Extended Community (see [TUNNEL-ENCAP]).

The NVO3 encapsulation design team has made a recommendation in [NVO3-ENCAP] for a control plane to:

1- Negotiate a subset of GENEVE option TLVs that can be carried on a GENEVE tunnel

2- Enforce an order for GENEVE option TLVs and

3- Limit the total number of options that could be carried on a GENEVE tunnel.

The EVPN control plane can easily extend the BGP Tunnel Encapsulation Attribute sub-TLV [TUNNEL-ENCAP] to specify the GENEVE tunnel options that can be received or transmitted over a GENEVE tunnels by a given NVE. [EVPN-GENEVE] describes the EVPN control plane extensions to support GENEVE.

4.5. EVPN OAM and application to NVO3

EVPN OAM (as in [EVPN-LSP-PING]) defines mechanisms to detect data

plane failures in an EVPN deployment over an MPLS network. These mechanisms detect failures related to P2P and P2MP connectivity, for multi-tenant unicast and multicast L2 traffic, between multi-tenant access nodes connected to EVPN PE(s), and in a single-homed, single-active or all-active redundancy model.

In general, EVPN OAM mechanisms defined for EVPN deployed in MPLS networks are equally applicable for EVPN in NVO3 networks.

4.6. EVPN as the control plane for NVO3 security

EVPN can be used to signal the security protection capabilities of a sender NVE, as well as what portion of an NVO3 packet (taking a GENEVE packet as an example) can be protected by the sender NVE, to ensure the privacy and integrity of tenant traffic carried over the NVO3 tunnels.

4.7. Advanced EVPN Features For NVO3 Networks

This section describes how EVPN can be used to deliver advanced capabilities in NVO3 networks.

4.7.1. Virtual Machine (VM) Mobility

[RFC7432] replaces the traditional Ethernet Flood-and-Learn behavior among NVEs with BGP-based MAC learning, which in return provides more control over the location of MAC addresses in the BD and consequently advanced features, such as MAC Mobility. If we assume that VM Mobility means the VM's MAC and IP addresses move with the VM, EVPN's MAC Mobility is the required procedure that facilitates VM Mobility. According to [RFC7432] section 15, when a MAC is advertised for the first time in a BD, all the NVEs attached to the BD will store Sequence Number zero for that MAC. When the MAC "moves" within the same BD but to a remote NVE, the NVE that just learned locally the MAC, increases the Sequence Number in the RT-2's MAC Mobility extended community to indicate that it owns the MAC now. That makes all the NVE in the BD change their tables immediately with no need to wait for any aging timer. EVPN guarantees a fast MAC Mobility without flooding or black-holes in the BD.

4.7.2. MAC Protection, Duplication Detection and Loop Protection

The advertisement of MACs in the control plane, allows advanced features such as MAC protection, Duplication Detection and Loop Protection.

[RFC7432] MAC Protection refers to EVPN's ability to indicate - in an RT-2 - that a MAC must be protected by the NVE receiving the route. The Protection is indicated in the "Sticky bit" of the MAC Mobility extended community sent along the RT-2 for a MAC. NVEs' ACs that are connected to subject-to-be-protected servers or VMs may set the Sticky bit on the RT-2s sent for the MACs associated to the ACs. Also statically configured MAC addresses should be advertised as Protected MAC addresses, since they are not subject to MAC Mobility procedures.

[RFC7432] MAC Duplication Detection refers to EVPN's ability to detect duplicate MAC addresses. A "MAC move" is a relearn event that happens at an access AC or through an RT-2 with a Sequence Number that is higher than the stored one for the MAC. When a MAC moves a number of times N within an M-second window between two NVEs, the MAC is declared as Duplicate and the detecting NVE does not re-advertise the MAC anymore.

While [RFC7432] provides MAC Duplication Detection, it does not protect the BD against loops created by backdoor links between NVEs. However, the same principle (based on the Sequence Number) may be extended to protect the BD against loops. When a MAC is detected as duplicate, the NVE may install it as a black-hole MAC and drop received frames with MAC SA and MAC DA matching that duplicate MAC. Loop Protection is described in [LOOP].

4.7.3. Reduction/Optimization of BUM Traffic In Layer-2 Services

In BDs with a significant amount of flooding due to Unknown unicast and Broadcast frames, EVPN may help reduce and sometimes even suppress the flooding.

In BDs where most of the Broadcast traffic is caused by ARP (Address Resolution Protocol) and ND (Neighbor Discovery) protocols on the TSes, EVPN's Proxy-ARP and Proxy-ND capabilities may reduce the flooding drastically. The use of Proxy-ARP/ND is specified in [PROXY-ARP-ND].

Proxy-ARP/ND procedures along with the assumption that TSes always issue a GARP (Gratuitous ARP) or an unsolicited Neighbor Advertisement message when they come up in the BD, may drastically reduce the unknown unicast flooding in the BD.

The flooding caused by TSes' IGMP/MLD or PIM messages in the BD may also be suppressed by the use of IGMP/MLD and PIM Proxy functions, as specified in [IGMP-MLD-PROXY] and [PIM-PROXY]. These two documents also specify how to forward IP multicast traffic efficiently within the same BD, translate soft state IGMP/MLD/PIM messages into hard

state BGP routes and provide fast-convergence redundancy for IP Multicast on multi-homed Ethernet Segments (ESes).

4.7.4. Ingress Replication (IR) Optimization For BUM Traffic

When an NVE attached to a given BD needs to send BUM traffic for the BD to the remote NVEs attached to the same BD, IR is a very common option in NVO3 networks, since it is completely independent of the multicast capabilities of the underlay network. Also, if the optimization procedures to reduce/suppress the flooding in the BD are enabled (section 4.7.3), in spite of creating multiple copies of the same frame at the ingress NVE, IR may be good enough. However, in BDs where Multicast (or Broadcast) traffic is significant, IR may be very inefficient and cause performance issues on virtual-switch-based NVEs.

[OPT-IR] specifies the use of AR (Assisted Replication) NVO3 tunnels in EVPN BDs. AR retains the independence of the underlay network while providing a way to forward Broadcast and Multicast traffic efficiently. AR uses AR-REPLICATORs that can replicate the Broadcast/Multicast traffic on behalf of the AR-LEAF NVEs. The AR-LEAF NVEs are typically virtual-switches or NVEs with limited replication capabilities. AR can work in a single-stage replication mode (Non-Selective Mode) or in a dual-stage replication mode (Selective Mode). Both modes are detailed in [OPT-IR].

In addition, [OPT-IR] also describes a procedure to avoid sending Broadcast, Multicast or Unknown unicast to certain NVEs that don't need that type of traffic. This is done by enabling PFL (Pruned Flood Lists) on a given BD. For instance, an virtual-switch NVE that learns all its local MAC addresses for a BD via Cloud Management System, does not need to receive the BD's Unknown unicast traffic. PFLs help optimize the BUM flooding in the BD.

4.7.5. EVPN Multi-homing

Another fundamental concept in EVPN is multi-homing. A given TS can be multi-homed to two or more NVEs for a given BD, and the set of links connected to the same TS is defined as Ethernet Segment (ES). EVPN supports single-active and all-active multi-homing. In single-active multi-homing only one link in the ES is active. In all-active multi-homing all the links in the ES are active for unicast traffic. Both modes support load-balancing:

- Single-active multi-homing means per-service load-balancing

to/from the TS, for example, in Figure 1, for BD1 only one of the NVEs can forward traffic from/to TS2. For a different BD, the other NVE may forward traffic.

- All-active multi-homing means per-flow load-balanding for unicast frames to/from the TS. That is, in Figure 1 and for BD1, both NVE4 and NVE5 can forward known unicast traffic to/from TS3. For BUM traffic only one of the two NVEs can forward traffic to TS3, and both can forward traffic from TS3.

There are two key aspects of EVPN multi-homing:

- DF (Designated Forwarder) election: the DF is the NVE that forwards the traffic to the ES in single-active mode. In case of all-active, the DF is the NVE that forwards the BUM traffic to the ES.

- Split-horizon function: prevents the TS from receiving echoed BUM frames that the TS itself sent to the ES. This is especially relevant in all-active ESes, where the TS may forward BUM frames to a non-DF NVE that can flood the BUM frames back to the DF NVE and then the TS. As an example, in Figure 1, assuming NVE4 is the DF for ES-2 in BD1, BUM frames sent from TS3 to NVE5 will be received at NVE4 and, since NVE4 is the DF for DB1, it will forward them back to TS3. Split-horizon allows NVE4 (and any multi-homed NVE for that matter) to identify if an EVPN BUM frame is coming from the same ES or different, and if the frame belongs to the same ES2, NVE4 will not forward the BUM frame to TS3, in spite of being the DF.

While [RFC7432] describes the default algorithm for the DF Election, [HRW-DF], [PREF-DF] and [AC-DF] specify other algorithms and procedures that optimize the DF Election.

The Split-horizon function is specified in [RFC7432] and it is carried out by using a special ESI-label that it identifies in the data path, all the BUM frames being originated from a given NVE and ES. Since the ESI-label is an MPLS label, it cannot be used in all the non-MPLS NVO3 encapsulations, therefore [EVPN-OVERLAY] defines a modified Split-horizon procedure that is based on the IP SA of the NVO3 tunnel, known as "Local-Bias". It is worth noting that Local-Bias only works for all-active multi-homing, and not for single-active multi-homing.

4.7.6. EVPN Recursive Resolution for Inter-Subnet Unicast Forwarding

Section 4.3. describes how EVPN can be used for Inter Subnet Forwarding among subnets of the same tenant. RT-2s and RT-5s allow the advertisement of host routes and IP Prefixes (RT-5) of any length. The procedures outlined by section 4.3. are similar to the ones in [RFC4364], only for NVO3 tunnels. However, [EVPN-PREFIX] also defines advanced Inter Subnet Forwarding procedures that allow the resolution of RT-5s to not only BGP next-hops but also "overlay indexes" that can be a MAC, a GW IP or an ESI, all of them in the tenant space.

Figure 4 illustrates an example that uses Recursive Resolution to a GWIP as per [IP-PREFIX] section 4.4.2. In this example, IP-VRFs in NVE1 and NVE2 are connected by a SBD (Supplementary BD). An SBD is a BD that connects all the IP-VRFs of the same tenant, via IRB, and has no ACs. NVE1 advertises the host route TS2-IP/L (IP address and Prefix Length of TS2) in an RT-5 with overlay index GWIP=IP1. Also, IP1 is advertised in an RT-2 associated to M1, VNI-S and BGP next-hop NVE1. Upon importing the two routes, NVE2 installs TS2-IP/L in the IP-VRF with a next-hop that is the GWIP IP1. NVE2 also installs M1 in the SBD, with VNI-S and NVE1 as next-hop. If TS3 sends a packet with IP DA=TS2, NVE2 will perform a Recursive Resolution of the RT-5 prefix information to the forwarding information of the correlated RT-2. The RT-5's Recursive Resolution has several advantages such as better convergence in scaled networks (since multiple RT-5s can be invalidated with a single withdrawal of the overlay index route) or the ability to advertise multiple RT-5s from an overlay index that can move or change dynamically. [EVPN-PREFIX] describes a few use-cases.

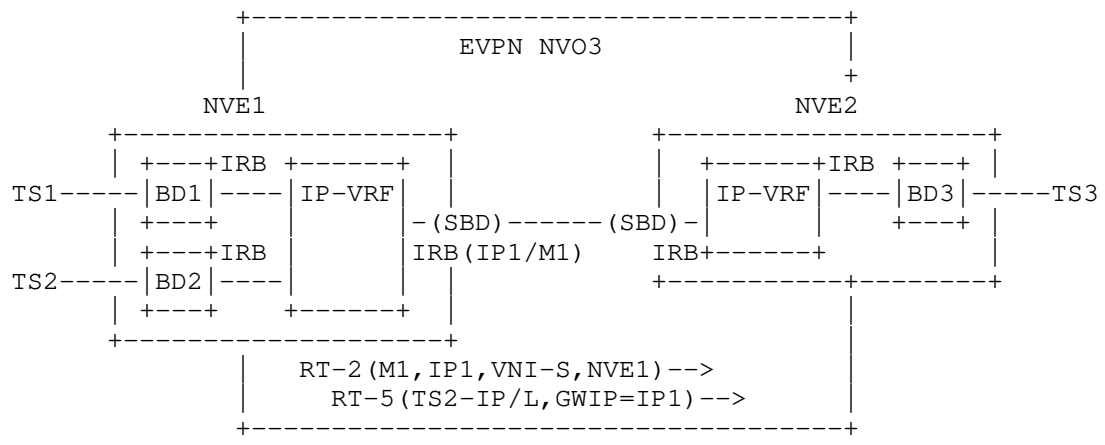


Figure 4 EVPN for L3 - Recursive Resolution example

4.7.7. EVPN Optimized Inter-Subnet Multicast Forwarding

The concept of the SBD described in section 4.7.6 is also used in [OISM] for the procedures related to Inter Subnet Multicast Forwarding across BDs of the same tenant. For instance, [OISM] allows the efficient forwarding of IP multicast traffic from any BD to any other BD (or even to the same BD where the Source resides). The [OISM] procedures are supported along with EVPN multi-homing, and for any tree allowed on NVO3 networks, including IR or AR. [OISM] also describes the interoperability between EVPN and other multicast technologies such as MVPN (Multicast VPN) and PIM for inter-subnet multicast.

[EVPN-MVPN] describes another potential solution to support EVPN to MVPN interoperability.

4.7.8. Data Center Interconnect (DCI)

Tenant Layer-2 and Layer-3 services deployed on NVO3 networks must be extended to remote NVO3 networks that are connected via non-NOV3 WAN networks (mostly MPLS based WAN networks). [EVPN-DCI] defines some architectural models that can be used to interconnect NVO3 networks via MPLS WAN networks.

When NVO3 networks are connected by MPLS WAN networks, [EVPN-DCI] specifies how EVPN can be used end-to-end, in spite of using a

different encapsulation in the WAN.

Even if EVPN can also be used in the WAN for Layer-2 and Layer-3 services, there may be a need to provide a Gateway function between EVPN for NVO3 encapsulations and IPVPN for MPLS tunnels. [EVPN-IPVPN] specifics the interworking function between EVPN and IPVPN for unicast Inter Subnet Forwarding. If Inter Subnet Multicast Forwarding is also needed across an IPVPN WAN, [OISM] describes the required interworking between EVPN and MVPN.

5. Conclusion

EVPN provides a unified control-plane that solves the NVE auto-discovery, tenant MAP/IP dissemination and advanced features required by NVO3 networks, in a scalable way and keeping the independence of the underlay IP Fabric, i.e. there is no need to enable PIM in the underlay network and maintain multicast states for tenant BDs.

This document justifies the use of EVPN for NVO3 networks, discusses its applicability to basic Layer-2 and Layer-3 connectivity requirements, as well as advanced features such as MAC-mobility, MAC Protection and Loop Protection, multi-homing, DCI and much more.

6. Conventions used in this document

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.

7. Security Considerations

This section will be added in future versions.

8. IANA Considerations

None.

9. References

9.1 Normative References

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <http://www.rfc-editor.org/info/rfc7432>.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <http://www.rfc-editor.org/info/rfc7365>.

[RFC7364] Narten, T., Ed., Gray, E., Ed., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", RFC 7364, DOI 10.17487/RFC7364, October 2014, <http://www.rfc-editor.org/info/rfc7364>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2 Informative References

[IP-PREFIX] Rabadan et al., "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-08, work in progress, October, 2017.

[INTER-SUBNET] Sajassi et al., "IP Inter-Subnet Forwarding in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-03, work in progress, February, 2017

[EVPN-USAGE] Rabadan et al., "Usage and applicability of BGP MPLS based Ethernet VPN", work in progress, draft-ietf-bess-evpn-usage-06, August 2017

[EVPN-OVERLAY] Sajassi-Drake et al., "A Network Virtualization Overlay Solution using EVPN", work in progress, draft-ietf-bess-evpn-overlay-08, March 2017

[GENEVE] Gross et al., "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-05, work in progress, September 2017

[NVO3-ENCAP] Boutros et al., "NVO3 Encapsulation Considerations", draft-ietf-nvo3-encap-01, work in progress, October 2017

[TUNNEL-ENCAP] Rosen et al., "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-03, work in progress, May 31, 2016.

[EVPN-LSP-PING] Jain et al., "LSP-Ping Mechanisms for EVPN and PBB-EVPN", draft-jain-bess-evpn-lsp-ping-05, work in progress, July 2017

[LOOP] Rabadan et al., "Loop Protection in EVPN networks", draft-snr-bess-evpn-loop-protect-00, work in progress, July 2017

[PROXY-ARP-ND] Rabadan et al., "Operational Aspects of Proxy-ARP/ND in EVPN Networks", draft-ietf-bess-evpn-proxy-arp-nd-03, work in progress, October 2017

[IGMP-MLD-PROXY] Sajassi et al., "IGMP and MLD Proxy for EVPN", draft-ietf-bess-evpn-igmp-mld-proxy-00, work in progress, March 2017

[PIM-PROXY] Rabadan et al., "PIM Proxy in EVPN Networks", draft-skr-bess-evpn-pim-proxy-01, work in progress, October 2017

[OPT-IR] Rabadan et al., "Optimized Ingress Replication solution for EVPN", draft-ietf-bess-evpn-optimized-ir-02, work in progress, August 2017

[HRW-DF] Mohanty et al., "A new Designated Forwarder Election for the EVPN", draft-ietf-bess-evpn-df-election-03, work in progress, October 2017

[PREF-DF] Rabadan et al., "Preference-based EVPN DF Election", draft-ietf-bess-evpn-pref-df-00, work in progress, June 2017

[AC-DF] Rabadan et al., "AC-Influenced Designated Forwarder Election for EVPN", draft-ietf-bess-evpn-ac-df-02, work in progress, October 2017

[OISM] Lin at al., "EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding", draft-lin-bess-evpn-irb-mcast-04, work in progress, October 2017

[EVPN-DCI] Rabadan et al., "Interconnect Solution for EVPN Overlay networks", draft-ietf-bess-dci-evpn-overlay-05, work in progress, July 2017

[BUM-UPDATE] Zhang et al., "Updates on EVPN BUM Procedures", draft-

ietf-bess-evpn-bum-procedure-updates-02, work in progress, September 2017

[EVPN-IPVPN] Rabadan-Sajassi et al., "EVPN Interworking with IPVPN", draft-rabadan-sajassi-bess-evpn-ipvpn-interworking-00, work in progress, October 2017

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <http://www.rfc-editor.org/info/rfc7348>.

[RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <http://www.rfc-editor.org/info/rfc7510>.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <http://www.rfc-editor.org/info/rfc4364>.

[CLOS1953] Clos, C., "A Study of Non-Blocking Switching Networks", The Bell System Technical Journal, Vol. 32(2), DOI 10.1002/j.1538-7305.1953.tb01433.x, March 1953.

[EVPN-GENEVE] Boutros et al., "EVPN control plane for Geneve", draft-boutros-bess-evpn-geneve-01, work in progress, February 2018.

[EVPN-MVPN] Sajassi et al., "Seamless Multicast Interoperability between EVPN and MVPN PEs", draft-sajassi-bess-evpn-mvpn-seamless-interop-00, work in progress, July 2017.

10. Acknowledgments

11. Contributors

12. Authors' Addresses

Jorge Rabadan (Editor)
Nokia
777 E. Middlefield Road
Mountain View, CA 94043 USA

Email: jorge.rabadan@nokia.com

Sami Boutros
VMware
Email: sboutros@vmware.com

Matthew Bocci
Nokia
Email: matthew.bocci@nokia.com

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2018

S. Pallagatti, Ed.
Independent Contributor
S. Paragiri
Juniper Networks
V. Govindan
M. Mudigonda
Cisco
G. Mirsky
ZTE Corp.
October 13, 2017

BFD for VXLAN
draft-spallagatti-bfd-vxlan-06

Abstract

This document describes use of Bidirectional Forwarding Detection (BFD) protocol in Virtual eXtensible Local Area Network (VXLAN) overlay network.

Status of This Memo

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

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

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

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

Table of Contents

1. Introduction 2
2. Conventions used in this document 3
2.1. Terminology 3
2.2. Requirements Language 3
3. Use cases 3
4. Deployment 4
5. BFD Packet Transmission over VXLAN Tunnel 5
5.1. BFD Packet Encapsulation in VXLAN 6
6. Reception of BFD packet from VXLAN Tunnel 7
6.1. Demultiplexing of the BFD packet 8
7. Use of reserved VNI 8
8. Echo BFD 8
9. IANA Considerations 8
10. Security Considerations 8
11. Contributors 8
12. Acknowledgments 9
13. Normative References 9
Authors' Addresses 10

1. Introduction

"Virtual eXtensible Local Area Network (VXLAN)" has been described in [RFC7348]. VXLAN provides an encapsulation scheme that allows virtual machines (VMs) to communicate in a data center network.

VXLAN is typically deployed in data centers interconnecting virtualized hosts, which may be spread across multiple racks. The individual racks may be part of a different Layer 3 network or they could be in a single Layer 2 network. The VXLAN segments/overlay networks are overlaid on top of these Layer 2 or Layer 3 networks.

A VM can communicate with another VM only if they are on the same VXLAN. VMs are unaware of VXLAN tunnels as VXLAN tunnel is terminated on VXLAN Tunnel End Point (VTEP) (hypervisor/TOR). VTEPs (hypervisor/TOR) are responsible for encapsulating and decapsulating frames exchanged among VMs.

Since underlay is a L3 network, ability to monitor path continuity, i.e. perform proactive continuity check (CC) for these tunnels is important. Asynchronous mode of BFD, as defined in [RFC5880], can be

used to monitor a VXLAN tunnel. Use of [I-D.ietf-bfd-multipoint] is for future study.

Also BFD in VXLAN can be used to monitor special service nodes that are designated to properly handle Layer 2 broadcast, unknown unicast, and multicast traffic. Such nodes, often referred "replicators", are usually virtual VTEPs can be monitored by physical VTEPs in order to minimize BUM traffic directed to unavialable replicator.

This document describes use of Bidirectional Forwarding Detection (BFD) protocol VXLAN to enable continuity monitoring between Network Virtualization Edges (NVEs) and/or availability of a replicator service node using BFD.

2. Conventions used in this document

2.1. Terminology

BFD - Bidirectional Forwarding Detection

CC - Continuity Check

NVE - Network Virtualization Edge

TOR - Top of Rack

VM - Virtual Machine

VTEP - VXLAN Tunnel End Point

VXLAN - Virtual eXtensible Local Area Network

2.2. Requirements Language

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

3. Use cases

Main use case of BFD for VXLAN is for continuity check of a tunnel. By exchanging BFD control packets between VTEPs an operator exercises the VXLAN path in both in underlay and overlay thus ensuring the VXLAN path availability and VTEPs reachability. BFD failure detection can be used for maintenance. There are other use cases such as

Layer 2 VMs:

Most deployments will have VMs with only L2 capabilities that may not support L3. BFD being a L3 protocol can be used as tunnel CC mechanism, where BFD will start and terminate at the NVEs, e.g. VTEPs.

It is possible to aggregate the CC sessions for multiple tenants by running a BFD session between the VTEPs over VxLAN tunnel. In rest of this document terms NVE and VTEP are used interchangeably.

Fault localization:

It is also possible that VMs are L3 aware and can possibly host a BFD session. In these cases BFD sessions can be established among VMs for CC. In addition, BFD sessions can be established among VTEPs for tunnel CC. Having a hierarchical OAM model helps localize faults though requires additional consideration.

Service node reachability:

Service node is responsible for sending BUM traffic. In case of service node tunnel terminates at VTEP and it might not even host VM. BFD session between TOR/hypervisor and service node can be used to monitor service node reachability.

4. Deployment

Figure 1 illustrates the scenario with two servers, each of them hosting two VMs. These servers host VTEPs that terminate two VXLAN tunnels with VNI number 100 and 200. Separate BFD sessions can be established between the VTEPs (IP1 and IP2) for monitoring each of the VXLAN tunnels (VNI 100 and 200). No BFD packets, intended to Hypervisor VTEP, should be forwarded to a VM as VM may drop BFD packets leading to false negative. This method is applicable whether VTEP is a virtual or physical device.

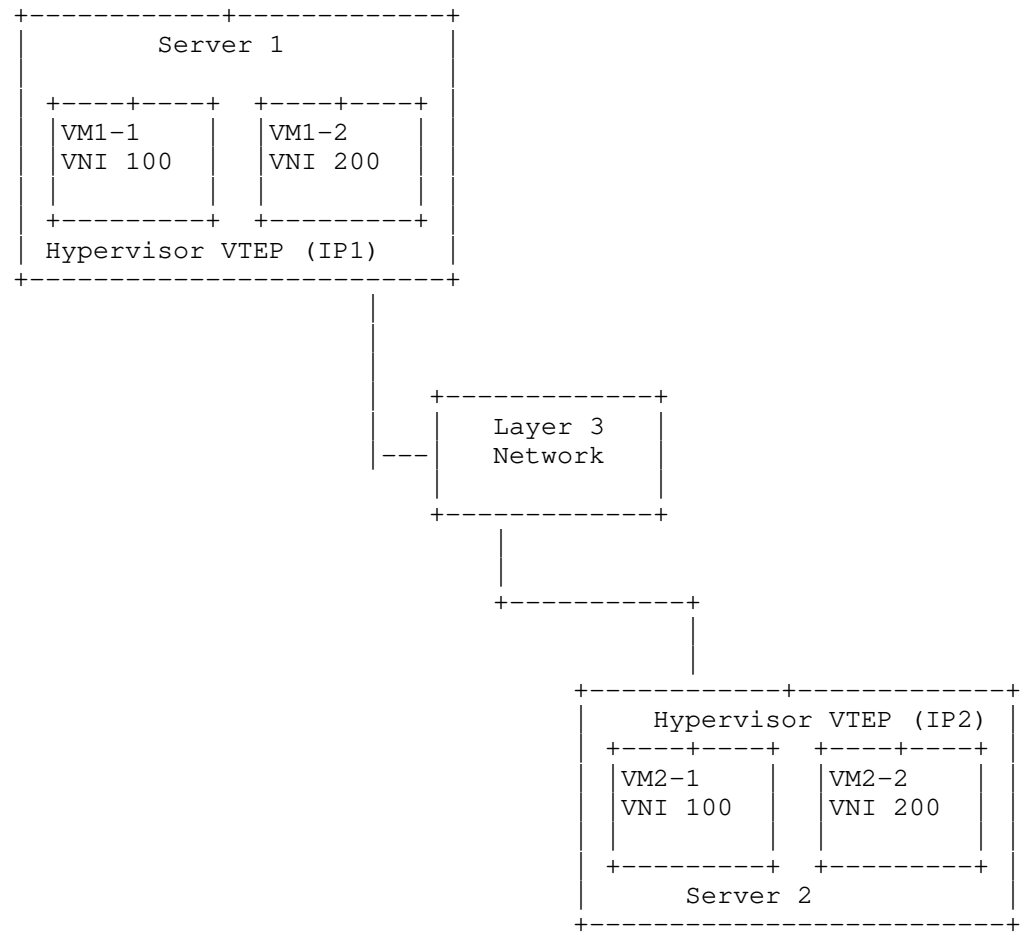


Figure 1: Reference VXLAN domain

5. BFD Packet Transmission over VXLAN Tunnel

BFD packet MUST be encapsulated and sent to a remote VTEP as explained in Section 5.1. Implementations SHOULD ensure that the BFD packets follow the same lookup path of VXLAN packets within the sender system.

5.1. BFD Packet Encapsulation in VXLAN

VXLAN packet format has been described in Section 5 of [RFC7348]. The Outer IP/UDP and VXLAN headers MUST be encoded by the sender as per [RFC7348].

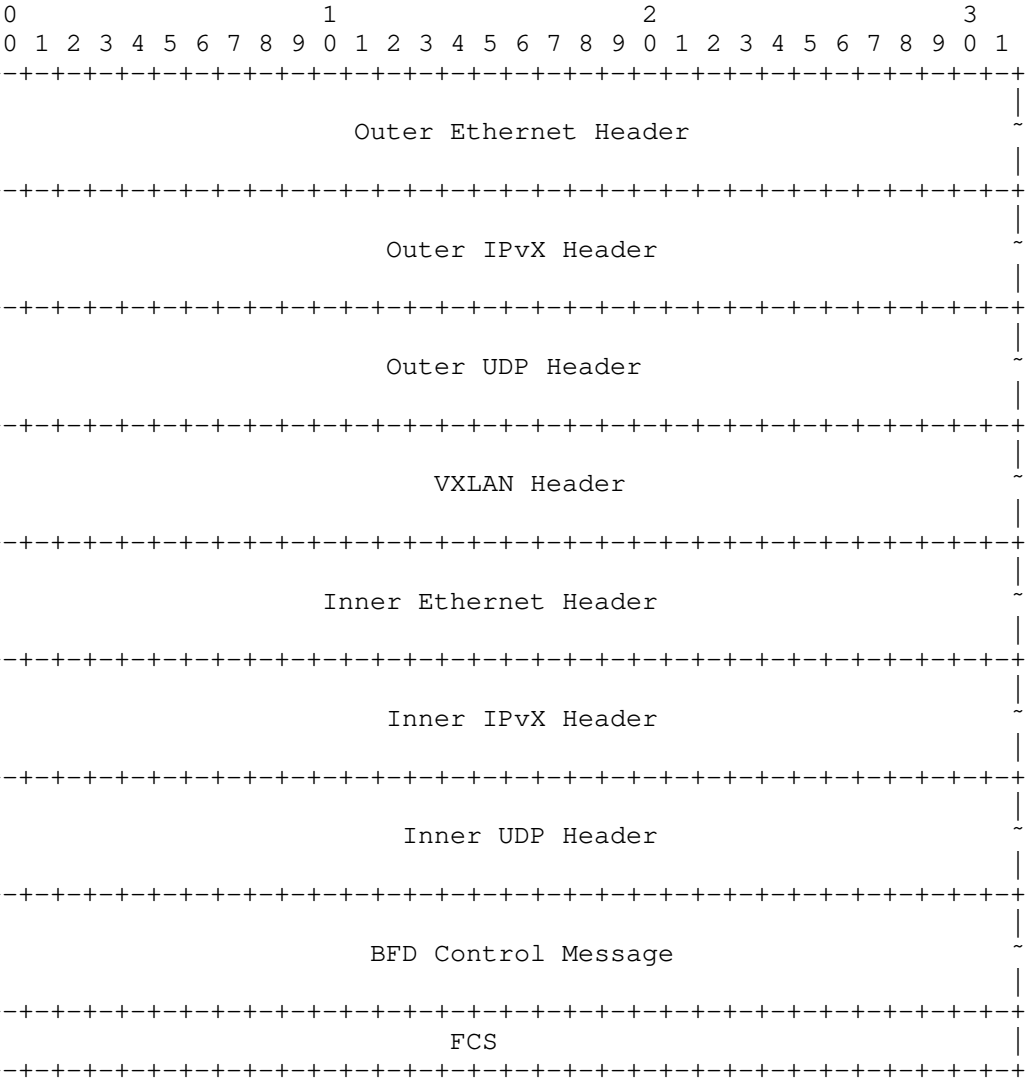


Figure 2: VXLAN Encapsulaion of BFD Control Message

The BFD packet MUST be carried inside the inner MAC frame of the VXLAN packet. The inner MAC frame carrying the BFD payload has the following format:

Ethernet Header:

Destination MAC: This MUST be a dedicated MAC (TBA) Section 9 or the MAC address of the destination VTEP. The details of how the MAC address of the destination VTEP is obtained are outside the scope of this document.

Source MAC: MAC address of the originating VTEP

IP header:

Source IP: IP address of the originating VTEP.

Destination IP: IP address of the terminating VTEP.

TTL: This MUST be set to 1. This is to ensure that the BFD packet is not routed within the L3 underlay network.

[Ed.Note]:Use of inner source and destination IP addresses needs more discussion by the WG.

The fields of the UDP header and the BFD control packet are encoded as specified in [RFC5881] for p2p VXLAN tunnels.

6. Reception of BFD packet from VXLAN Tunnel

Once a packet is received, VTEP MUST validate the packet as described in Section 4.1 of [RFC7348]. If the Destination MAC of the inner MAC frame matches the dedicated MAC or the MAC address of the VTEP the packet MUST be processed further.

The UDP destination port and the TTL of the inner Ethernet frame MUST be validated to determine if the received packet can be processed by BFD. BFD packet with inner MAC set to VTEP or dedicated MAC address MUST NOT be forwarded to VMs.

To ensure BFD detects the proper configuration of VXLAN Network Identifier (VNI) in a remote VTEP, a lookup SHOULD be performed with the MAC-DA and VNI as key in the Virtual Forwarding Instance (VFI) table of the originating/ terminating VTEP in order to exercise the VFI associated with the VNI.

6.1. Demultiplexing of the BFD packet

Demultiplexing of IP BFD packet has been defined in Section 3 of [RFC5881]. Since multiple BFD sessions may be running between two VTEPs, there needs to be a mechanism for demultiplexing received BFD packets to the proper session. The procedure for demultiplexing packets with Your Discriminator equal to 0 is different from [RFC5880]. For such packets, the BFD session MUST be identified using the inner headers, i.e. the source IP and the destination IP present in the IP header carried by the payload of the VXLAN encapsulated packet. The VNI of the packet SHOULD be used to derive interface related information for demultiplexing the packet. If BFD packet is received with non-zero Your Discriminator then BFD session MUST be demultiplexed only with Your Discriminator as the key.

7. Use of reserved VNI

BFD session MAY be established for the reserved VNI 0. One way to aggregate BFD sessions between VTEP's is to establish a BFD session with VNI 0. A VTEP MAY also use VNI 0 to establish a BFD session with a service node.

8. Echo BFD

Support for echo BFD is outside the scope of this document.

9. IANA Considerations

IANA is requested to assign a dedicated MAC address to be used as the Destination MAC address of the inner Ethernet which carries BFD control packet in IP/UDP encapsulation.

10. Security Considerations

Document recommends setting of inner IP TTL to 1 which could lead to DDoS attack, implementation MUST have throttling in place. Throttling MAY be relaxed for BFD packets based on port number.

Other than inner IP TTL set to 1 this specification does not raise any additional security issues beyond those of the specifications referred to in the list of normative references.

11. Contributors

Reshad Rahman
rrahman@cisco.com
Cisco

12. Acknowledgments

Authors would like to thank Jeff Hass of Juniper Networks for his reviews and feedback on this material.

Authors would also like to thank Nobo Akiya, Marc Binderberger and Shahram Davari for the extensive review.

13. Normative References

[I-D.ietf-bfd-multipoint] Katz, D., Ward, D., and J. Networks, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-10 (work in progress), April 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[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>.

[RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <https://www.rfc-editor.org/info/rfc5881>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[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>.

Authors' Addresses

Santosh Pallagatti (editor)
Independent Contributor

Email: santosh.pallagatti@gmail.com

Sudarsan Paragiri
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206
USA

Email: sparagiri@juniper.net

Vengada Prasad Govindan
Cisco

Email: venggovi@cisco.com

Mallik Mudigonda
Cisco

Email: mmudigon@cisco.com

Greg Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com