

NVO3 Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 27, 2018

G. Fioccola  
Telecom Italia  
G. Mirsky  
ZTE Corp.  
T. Mizrahi  
Marvell  
June 25, 2018

Performance Measurement (PM) with Alternate Marking in Network  
Virtualization Overlays (NVO3)  
draft-fmm-nvo3-pm-alt-mark-02

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 December 27, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

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

## 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].

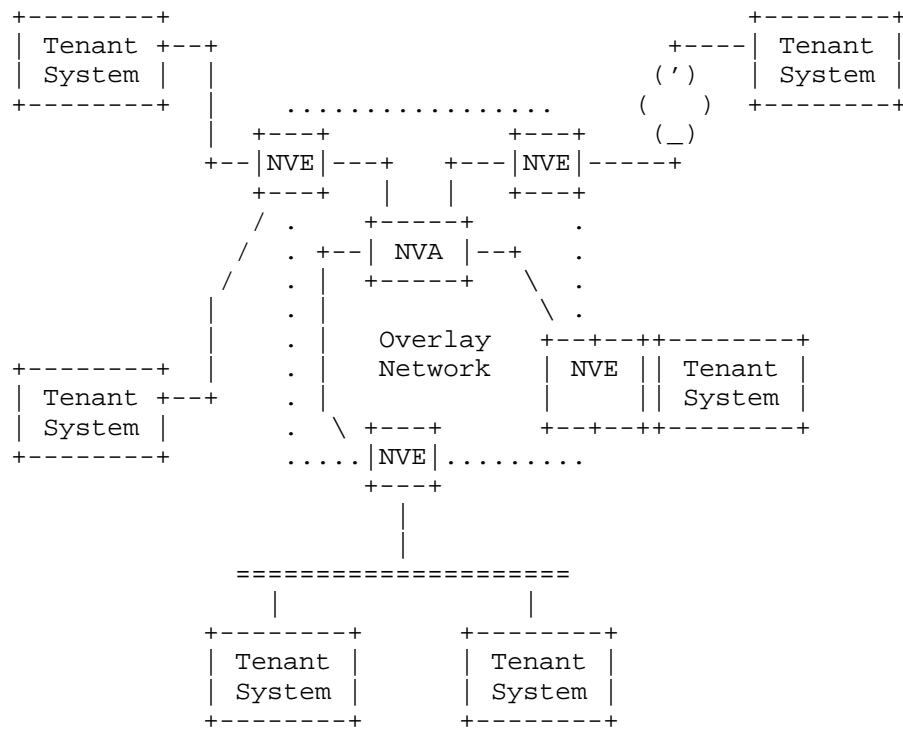


Figure 1: Generic Reference Model for DC Network Virtualization Overlays (RFC7365)

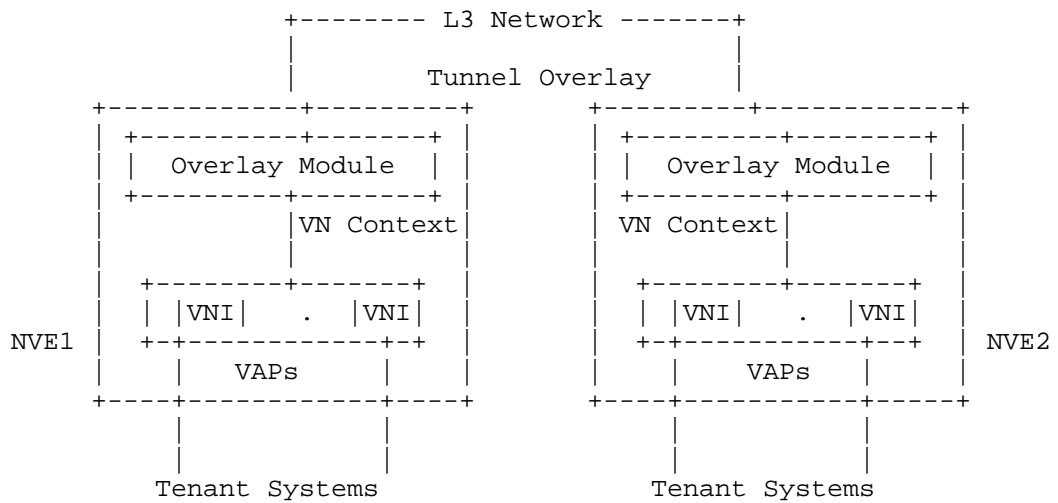


Figure 2: Generic NVE Reference Model (RFC7365)

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.

```

0
0  1
+--+--+--+
| L | D |
+--+--+--+

```

Figure 3: Mark field (M) format

where:

- o L - Loss bit;
- o 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:

- o 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
- o 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.

Geneve Header:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|Ver|  Opt Len  |O|C| M | Rsvd. |               Protocol Type               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Virtual Network Identifier (VNI)               |   Reserved   |
+-----+-----+-----+-----+-----+-----+-----+-----+
|               Variable Length Options               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

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., 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.

[I-D.ietf-nvo3-geneve]

Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-06 (work in progress), March 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-03 (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-01 (work in progress), March 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  
Telecom Italia  
Via Reiss Romoli, 274  
Torino 10148  
Italy

Email: [giuseppe.fioccola@telecomitalia.it](mailto:giuseppe.fioccola@telecomitalia.it)

Greg Mirsky  
ZTE Corp.

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)

Tal Mizrahi  
Marvell  
6 Hamada St.  
Yokneam  
Israel

Email: [talmi@marvell.com](mailto:talmi@marvell.com)

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.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

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

## 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].

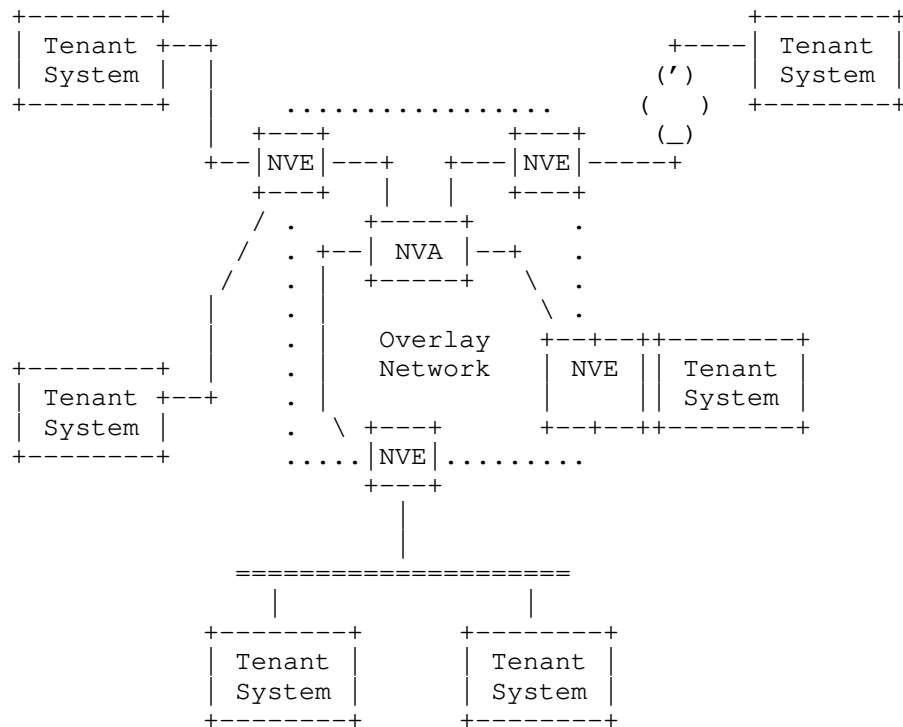


Figure 1: Generic Reference Model for DC Network Virtualization Overlays (RFC7365)

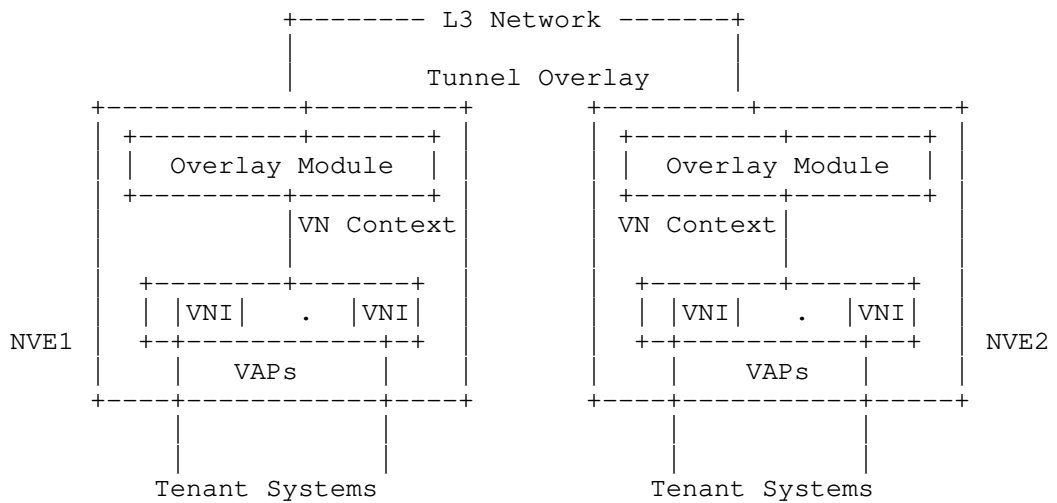


Figure 2: Generic NVE Reference Model (RFC7365)

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.



```

0
0  1
+--+--+--+
| L | D |
+--+--+--+

```

Figure 3: Mark field (M) format

where:

- o L - Loss bit;
- o 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:

- o 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
- o 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.

Geneve Header:

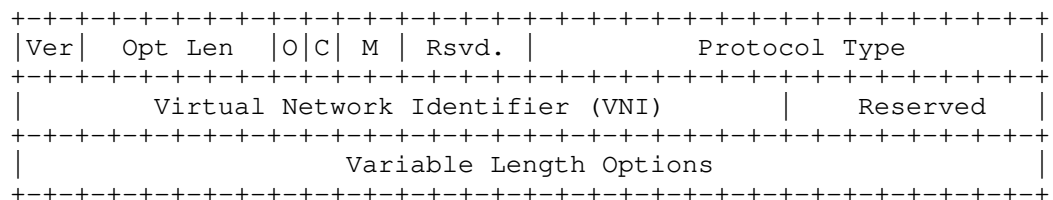


Figure 4: Geneve Header

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](mailto:giuseppe.fioccola@huawei.com)

Greg Mirsky  
ZTE Corp.

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)

Tal Mizrahi  
Huawei Network.IO Innovation Lab  
Israel

Email: [tal.mizrahi.phd@gmail.com](mailto:tal.mizrahi.phd@gmail.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 3, 2019

J. Gross, Ed.  
I. Ganga, Ed.  
Intel  
T. Sridhar, Ed.  
VMware  
July 02, 2018

Geneve: Generic Network Virtualization Encapsulation  
draft-ietf-nvo3-geneve-07

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 January 3, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## 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 . . . . .	21
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.

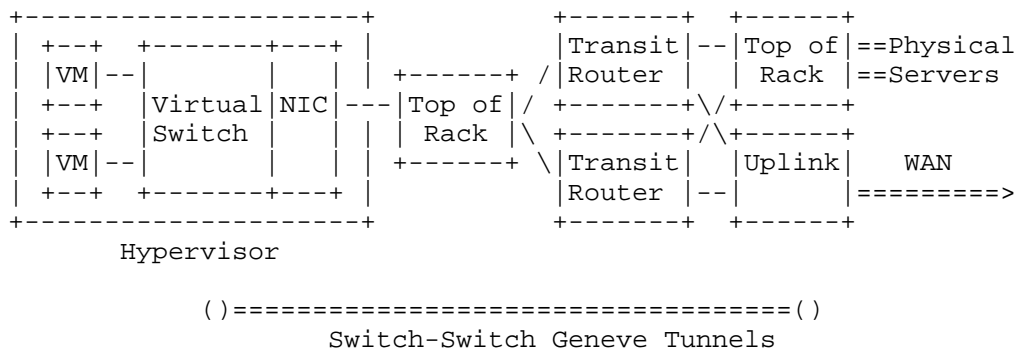


Figure 1: Sample Geneve Deployment

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:

- o The data plane is generic and extensible enough to support current and future control planes.
- o Tunnel components are efficiently implementable in both hardware and software without restricting capabilities to the lowest common denominator.
- o 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

[illegible]

```
Outer IPv4 Header:
```

	Version	IHL  Type of Service	Total Length	
	Identification		Flags	Fragment Offset
	Time to Live  Protocol=17 UDP	Header Checksum		
	Outer Source IPv4 Address			
	Outer Destination IPv4 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 |
+-----+
|          Inner Source MAC Address          |
+-----+
|Optional Ethertype=C-Tag 802.1Q| Inner VLAN Tag Information |
+-----+

```

## Payload:

```

+-----+
| Ethertype of Original Payload |
+-----+
|          Original Ethernet Payload          |
|
| (Note that the original Ethernet Frame's FCS is not included) |
+-----+

```

## Frame Check Sequence:

```

+-----+
| New FCS (Frame Check Sequence) for Outer Ethernet Frame |
+-----+

```

## 3.2. Geneve Packet Format Over IPv6

```

      0          1          2          3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Outer Ethernet Header:
+-----+
|          Outer Destination MAC Address          |
+-----+
| Outer Destination MAC Address | Outer Source MAC Address |
+-----+
|          Outer Source MAC Address          |
+-----+
|Optional Ethertype=C-Tag 802.1Q| Outer VLAN Tag Information |
+-----+

```



```

|      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 |
+-----+

```

```

|                               Inner Source MAC Address                               |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Optional Ethertype=C-Tag 802.1Q | Inner VLAN Tag Information |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Payload:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Ethertype of Original Payload |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Original Ethernet Payload                               |
| (Note that the original Ethernet Frame's FCS is not included) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Frame Check Sequence:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| New FCS (Frame Check Sequence) for Outer Ethernet Frame |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

### 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.

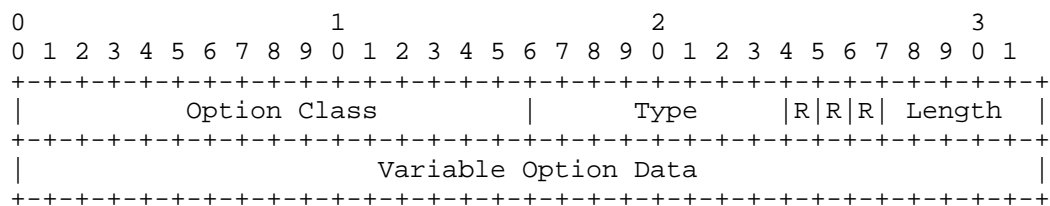
Protocol Type (16 bits): The type of the protocol data unit appearing after the Geneve header. This follows the EtherType [ETYPES] convention with Ethernet itself being represented by the value 0x6558.

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:

```

0 1 2 3 4 5 6 7 8
+---+---+---+---+---+
|C|   Type   |
+---+---+---+---+---+

```

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:

- o 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.
- o 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.
- o 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 fragmentation 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:

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

### 6.1. Data Confidentiality

Geneve is a network virtualization overlay encapsulation protocol designed to establish tunnels between network virtualization endpoints (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.

An NVE, used in multi-tenant environments, **MUST** have the capability 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 NVEs **SHOULD** have a configurable option to disable 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 operators security domain, for example over a 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 Geneve NVE **MUST** have the capability 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. The NVE **SHOULD** have a configuration option to enable or disable the data integrity protection, based on the presence of threats in their environment. 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, a Geneve NVE **MUST** support an Authentication mechanism, such as IPsec AH, to ensure that the Geneve packet originated from the intended NVE peer, in environments where 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 can 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 Geneve NVEs used in such environments **SHOULD** support 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
0xFFFF0..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](mailto:jesse@kernel.org)

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

Email: [ilango.s.ganga@intel.com](mailto:ilango.s.ganga@intel.com)

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

Email: [tsridhar@vmware.com](mailto:tsridhar@vmware.com)

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: September 8, 2020

J. Gross, Ed.  
I. Ganga, Ed.  
Intel  
T. Sridhar, Ed.  
VMware  
March 07, 2020

Geneve: Generic Network Virtualization Encapsulation  
draft-ietf-nvo3-geneve-16

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 document describes Geneve, an encapsulation 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 September 8, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
1.1. Requirements Language . . . . .	4
1.2. Terminology . . . . .	4
2. Design Requirements . . . . .	6
2.1. Control Plane Independence . . . . .	7
2.2. Data Plane Extensibility . . . . .	7
2.2.1. Efficient Implementation . . . . .	8
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 . . . . .	11
3.3. UDP Header . . . . .	13
3.4. Tunnel Header Fields . . . . .	14
3.5. Tunnel Options . . . . .	15
3.5.1. Options Processing . . . . .	17
4. Implementation and Deployment Considerations . . . . .	18
4.1. Applicability Statement . . . . .	18
4.2. Congestion Control Functionality . . . . .	19
4.3. UDP Checksum . . . . .	19
4.3.1. UDP Zero Checksum Handling with IPv6 . . . . .	19
4.4. Encapsulation of Geneve in IP . . . . .	21
4.4.1. IP Fragmentation . . . . .	21
4.4.2. DSCP, ECN and TTL . . . . .	22
4.4.3. Broadcast and Multicast . . . . .	23
4.4.4. Unidirectional Tunnels . . . . .	23
4.5. Constraints on Protocol Features . . . . .	24
4.5.1. Constraints on Options . . . . .	24
4.6. NIC Offloads . . . . .	25
4.7. Inner VLAN Handling . . . . .	25
5. Transition Considerations . . . . .	26
6. Security Considerations . . . . .	26
6.1. Data Confidentiality . . . . .	27
6.1.1. Inter-Data Center Traffic . . . . .	27
6.2. Data Integrity . . . . .	28
6.3. Authentication of NVE peers . . . . .	29
6.4. Options Interpretation by Transit Devices . . . . .	29

6.5. Multicast/Broadcast . . . . .	29
6.6. Control Plane Communications . . . . .	29
7. IANA Considerations . . . . .	30
8. Contributors . . . . .	31
9. Acknowledgements . . . . .	32
10. References . . . . .	33
10.1. Normative References . . . . .	33
10.2. Informative References . . . . .	34
Authors' Addresses . . . . .	37

## 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, for example, ranging all the way from VLANs [IEEE.802.1Q\_2018] and MPLS [RFC3031] through the more recent VXLAN [RFC7348] (Virtual eXtensible Local Area Network) and NVGRE [RFC7637] (Network Virtualization Using Generic Routing Encapsulation), 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. Note that the list of protocols presented above is non-exhaustive.

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.

Work such as [VL2] (A Scalable and Flexible Data Center Network) and the NVO3 Data Plane Requirements [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 metadata (e.g. system state) along with the packet data; example use cases of metadata are noted below. The use of some metadata is certainly not a foreign concept - nearly all protocols used for network 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 port identifiers for simple security policies to sending service based context for advanced middlebox applications that terminate and re-encapsulate Geneve traffic.

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", "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.

### 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 (Network Interface Controller) 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 of packets within a flow.

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

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

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).

Middlebox. The term middlebox in the context of this document refers to network service functions or appliances for service interposition that would typically implement NVE functionality, which terminate or re-encapsulate Geneve traffic.

NIC. Network Interface Controller. Also called as Network Interface Card or Network Adapter. 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.

Transit device. A forwarding element (e.g. router or switch) 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.

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, tunnel 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. Tunnel endpoints are frequently a component of an NVE (Network Virtualization Edge) 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 for data center environments, 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. Many of the concepts of network virtualization overlays over Layer 3 IP networks are described in the NVO3 Framework [RFC7365]. 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.

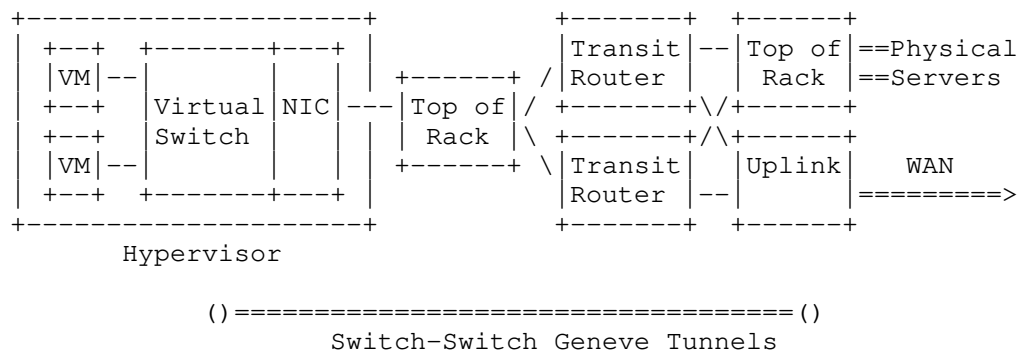


Figure 1: Sample Geneve Deployment

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:

- o The data plane is generic and extensible enough to support current and future control planes.
- o Tunnel components are efficiently implementable in both hardware and software without restricting capabilities to the lowest common denominator.
- o 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, VXLAN [RFC7348] 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 is 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 reduces the chance of obsolescence 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 terminating/re-encapsulating 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 a desired 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.

Tunnel 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 tunnel endpoints are the ultimate consumers of the data, transmitters can tailor their output to the capabilities of the recipient.

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 modify Geneve headers and MUST NOT insert or delete options, which is the responsibility of tunnel endpoints. Options, if present in the packet, MUST only be generated and terminated by tunnel 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.

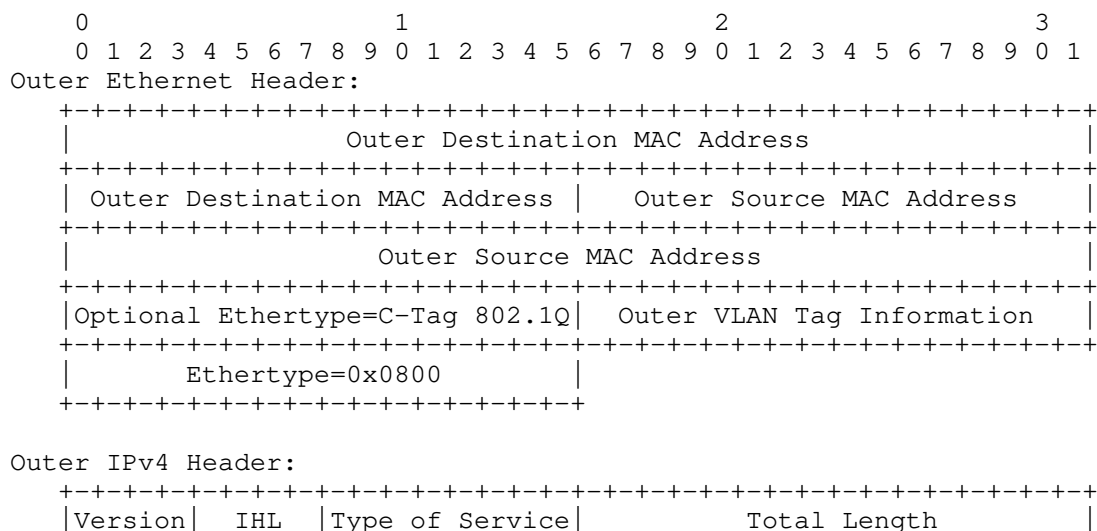
In addition, nearly all underlay fabrics are designed to exploit parallelism in traffic to spread load across multiple links without introducing reordering in individual flows. These equal cost multipathing (ECMP) techniques typically involve parsing and hashing the addresses and port numbers from the 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 tunnel 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



```

+-----+-----+-----+-----+-----+-----+-----+-----+
| Identification | Flags | Fragment Offset |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Time to Live | Protocol=17 UDP | Header Checksum |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Outer Source IPv4 Address |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Outer Destination IPv4 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 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Inner Source MAC Address |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Optional Ethertype=C-Tag 802.1Q | Inner VLAN Tag Information |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

## Payload:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| Ethertype of Original Payload |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Original Ethernet Payload |
|
| (Note that the original Ethernet Frame's Preamble, Start Frame
| Delimiter(SFD) & Frame Check Sequence(FCS) are not included
| and the Ethernet Payload need not be 4-byte aligned)
|
+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

+-----+
Frame Check Sequence:
+-----+
|   New Frame Check Sequence (FCS) for Outer Ethernet Frame   |
+-----+

```

### 3.2. Geneve Packet Format Over IPv6

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Outer Ethernet Header:
+-----+
|                               Outer Destination MAC Address                               |
+-----+
| Outer Destination MAC Address | Outer Source MAC Address |
+-----+
|                               Outer Source MAC Address                               |
+-----+
| Optional Ethertype=C-Tag 802.1Q | Outer VLAN Tag Information |
+-----+
|                               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 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Inner Source MAC Address          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Optional Ethertype=C-Tag 802.1Q | Inner VLAN Tag Information |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

## Payload:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
| Ethertype of Original Payload |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Original Ethernet Payload          |
+-----+-----+-----+-----+-----+-----+-----+-----+
| (Note that the original Ethernet Frame's Preamble, Start Frame |
| Delimiter(SFD) & Frame Check Sequence(FCS) are not included  |
| and the Ethernet Payload need not be 4-byte aligned)          |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

## Frame Check Sequence:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|          New Frame Check Sequence (FCS) for Outer Ethernet Frame          |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

### 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. In addition to setting the source port, for IPv6, flow label MAY also be used for providing entropy. For an example of using IPv6 flow label for tunnel use cases, see [RFC6438].

If Geneve traffic is shared with other UDP listeners on the same IP address, tunnel endpoints SHOULD implement a mechanism to ensure ICMP return traffic arising from network errors is directed to the correct listener. The definition of such a mechanism is beyond the scope of this document.

**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. It is the responsibility of the control plane for any reconfiguration of the assigned port and its interpretation by respective devices. The definition of the control plane is beyond the scope of this document.

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

**UDP checksum:** In order to protect the Geneve header, options and payload from potential data corruption, UDP checksum SHOULD be generated as specified in [RFC0768] and [RFC1112] when Geneve is encapsulated in IPv4. To protect the IP header, Geneve header, options and payload from potential data corruption, the UDP checksum MUST be generated by default as specified in [RFC0768] and [RFC8200] when Geneve is encapsulated in IPv6, except for certain conditions, which are outlined in the next paragraph. Upon receiving such packets with non-zero UDP checksum, the



receiving tunnel endpoints MUST validate the checksum. If the checksum is not correct, the packet MUST be dropped, otherwise the packet MUST be accepted for decapsulation.

Under certain conditions, the UDP checksum MAY be set to zero on transmit for packets encapsulated in both IPv4 and IPv6 [RFC8200]. See Section 4.3 for additional requirements that apply when using zero UDP checksum with IPv4 and IPv6. Disabling the use of UDP checksums is an operational consideration that should take into account the risks and effects of packet corruption.

### 3.4. Tunnel Header Fields

Ver (2 bits): The current version number is 0. Packets received by a tunnel endpoint with an unknown version MUST be dropped. Transit devices interpreting 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.

Transit devices MUST maintain consistent forwarding behavior irrespective of the value of 'Opt Len', including ECMP link selection.

O (1 bit): Control packet. This packet contains a control message. Control messages are sent between tunnel endpoints. Tunnel endpoints MUST NOT forward the payload and transit devices MUST NOT attempt to interpret it. Since control messages are less frequent, it is RECOMMENDED that tunnel 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 tunnel 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.

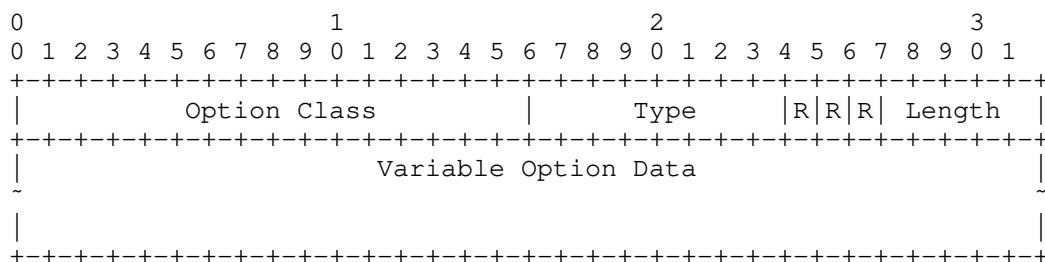
Rsvd. (6 bits): Reserved field, which MUST be zero on transmission and MUST be ignored on receipt.

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.

### 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 allocation by IETF Review and for Experimental Use (see Section 7).

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 separate documents.

The high order bit of the option type indicates that this is a critical option. If the receiving tunnel endpoint does not recognize this option and this bit is set then the packet MUST be dropped. If this 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:

```

0 1 2 3 4 5 6 7 8
+---+---+---+---+
|C|       Type       |
+---+---+---+---+

```

The requirement to drop a packet with an unknown option with the 'C' bit set 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. These bits MUST be zero on transmission and MUST be 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 an option header and no 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 a tunnel endpoint that processes the options.

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 interpreting Geneve headers (which may or may not include options) MUST handle 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 beyond the the scope of this document. However, to ensure interoperability between heterogeneous devices some requirements are imposed on options and the devices that process them:

- o Receiving tunnel 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.
- o The contents of the options and their ordering MUST NOT be modified by transit devices.
- o If a tunnel endpoint receives a Geneve packet with 'Opt Len' (total length of all options) that exceeds the options processing capability of the tunnel endpoint then the tunnel endpoint MUST drop such packets. An implementation may raise an exception to the control plane of such an event. It is the responsibility of the control plane to ensure the communicating peer tunnel endpoints have the processing capability to handle the total length of options. The definition of the control plane is beyond the scope of this document.

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.

Architecturally, options are intended to be self-descriptive and independent. This enables parallelism in option processing and reduces implementation complexity. However, the control plane may impose certain ordering restrictions as described in Section 4.5.1.

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. Actual definition of the option type is beyond the scope of this document. The option type and its interpretation should be defined by the entity that owns the option class.

Options may be processed by NIC hardware utilizing offloads (e.g. LSO and LRO) as described in Section 4.6. Careful consideration should be given to how the offload capabilities outlined in Section 4.6 impact an option's design.

#### 4. Implementation and Deployment Considerations

##### 4.1. Applicability Statement

Geneve is a network virtualization overlay encapsulation protocol designed to establish tunnels between NVEs over an existing IP network. It is intended for use in public or private data center environments, for deploying multi-tenant overlay networks over an existing IP underlay network.

Geneve is a UDP based encapsulation protocol transported over existing IPv4 and IPv6 networks. Hence, as a UDP based protocol, Geneve adheres to the UDP usage guidelines as specified in [RFC8085]. The applicability of these guidelines are dependent on the underlay IP network and the nature of Geneve payload protocol (example TCP/IP, IP/Ethernet).

Geneve is intended to be deployed in a data center network environment operated by a single operator or adjacent set of cooperating network operators that fits with the definition of controlled environments in [RFC8085]. A network in a controlled environment can be managed to operate under certain conditions whereas in the general Internet this cannot be done. Hence requirements for a tunnel protocol operating under a controlled environment can be less restrictive than the requirements of the general Internet.

For the purpose of this document, a traffic-managed controlled environment (TMCE) is defined as an IP network that is traffic-engineered and/or otherwise managed (e.g., via use of traffic rate

limiters) to avoid congestion. The concept of TMCE is outlined in [RFC8086]. Significant portions of the text in Section 4.1 through Section 4.3 are based on [RFC8086] as applicable to Geneve.

It is the responsibility of the operator to ensure that the guidelines/requirements in this section are followed as applicable to their Geneve deployment(s).

#### 4.2. Congestion Control Functionality

Geneve does not natively provide congestion control functionality and relies on the payload protocol traffic for congestion control. As such Geneve **MUST** be used with congestion controlled traffic or within a network that is traffic managed to avoid congestion (TMCE). An operator of a traffic managed network (TMCE) may avoid congestion by careful provisioning of their networks, rate-limiting of user data traffic and traffic engineering according to path capacity.

#### 4.3. UDP Checksum

In order to provide integrity of Geneve headers, options and payload, (for example to avoid misdelivery of payload to different tenant systems) in case of data corruption, the outer UDP checksum **SHOULD** be used with Geneve when transported over IPv4. The UDP checksum provides a statistical guarantee that a payload was not corrupted in transit. These integrity checks are not strong from a coding or cryptographic perspective and are not designed to detect physical-layer errors or malicious modification of the datagram (see Section 3.4 of [RFC8085]). In deployments where such a risk exists, an operator **SHOULD** use additional data integrity mechanisms such as offered by IPsec (see Section 6.2).

An operator **MAY** choose to disable UDP checksums and use zero checksums if Geneve packet integrity is provided by other data integrity mechanisms such as IPsec or additional checksums or if one of the conditions in Section 4.3.1 a, b, c are met.

By default, UDP checksums **MUST** be used when Geneve is transported over IPv6. A tunnel endpoint **MAY** be configured for use with zero UDP checksum if additional requirements in Section 4.3.1 are met.

##### 4.3.1. UDP Zero Checksum Handling with IPv6

When Geneve is used over IPv6, the UDP checksum is used to protect IPv6 headers, UDP headers and Geneve headers, options and payload from potential data corruption. As such by default Geneve **MUST** use UDP checksums when transported over IPv6. An operator **MAY** choose to configure to operate with zero UDP checksum if operating in a traffic

managed controlled environment as stated in Section 4.1 if one of the following conditions are met.

- a. It is known that the packet corruption is exceptionally unlikely (perhaps based on knowledge of equipment types in their underlay network) and the operator is willing to take a risk of undetected packet corruption
- b. It is judged through observational measurements (perhaps through historic or current traffic flows that use non zero checksum) that the level of packet corruption is tolerably low and where the operator is willing to take the risk of undetected corruption.
- c. Geneve payload is carrying applications that are tolerant of misdelivered or corrupted packets (perhaps through higher layer checksum validation and/or reliability through retransmission)

In addition Geneve tunnel implementations using zero UDP checksum MUST meet the following requirements:

1. Use of UDP checksum over IPv6 MUST be the default configuration for all Geneve tunnels.
2. If Geneve is used with zero UDP checksum over IPv6 then such tunnel endpoint implementation MUST meet all the requirements specified in Section 4 of [RFC6936] and requirement 1 as specified in Section 5 of [RFC6936] as that is relevant to Geneve.
3. The Geneve tunnel endpoint that decapsulates the tunnel SHOULD check the source and destination IPv6 addresses are valid for the Geneve tunnel that is configured to receive zero UDP checksum and discard other packets for which such check fails.
4. The Geneve tunnel endpoint that encapsulates the tunnel MAY use different IPv6 source addresses for each Geneve tunnel that uses zero UDP checksum mode in order to strengthen the decapsulator's check of the IPv6 source address (i.e the same IPv6 source address is not to be used with more than one IPv6 destination address, irrespective of whether that destination address is a unicast or multicast address). When this is not possible, it is RECOMMENDED to use each source address for as few Geneve tunnels that use zero UDP checksum as is feasible.

Note that (for requirements 3 and 4) the receiving tunnel endpoint can apply these checks only if it has out-of-band knowledge that the encapsulating tunnel endpoint is applying the

indicated behavior. One possibility to obtain this out-of-band knowledge is through signaling by the control plane. The definition of the control plane is beyond the scope of this document.

5. Measures SHOULD be taken to prevent Geneve traffic over IPv6 with zero UDP checksum from escaping into the general Internet. Examples of such measures include employing packet filters at the gateways or edge of Geneve network and/or keeping logical or physical separation of the Geneve network from networks carrying the general Internet traffic.

The above requirements do not change either the requirements specified in [RFC8200] or the requirements specified in [RFC6936].

The use of the source IPv6 address in addition to the destination IPv6 address, plus the recommendation against reuse of source IPv6 addresses among Geneve tunnels collectively provide some mitigation for the absence of UDP checksum coverage of the IPv6 header. A traffic-managed controlled environment that satisfies at least one of three conditions listed at the beginning of this section provides additional assurance.

Editorial Note (The following paragraph to be removed by the RFC Editor before publication)

It was discussed during TSVART early review if the level of requirement for using different IPv6 source addresses for different tunnel destinations would need to be "MAY" or "SHOULD". The discussion concluded that it was appropriate to keep this as "MAY", since it was considered not realistic for control planes having to maintain a high level of state on a per tunnel destination basis. In addition, the text above provides sufficient guidance to operators and implementors on possible mitigations.

#### 4.4. 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.

##### 4.4.1. IP Fragmentation

It is strongly RECOMMENDED that Path MTU Discovery ([RFC1191], [RFC8201]) be used to prevent or minimize fragmentation. The use of Path MTU Discovery on the transit network provides the encapsulating



tunnel endpoint with soft-state about the link that it may use to prevent or minimize fragmentation depending on its role in the virtualized network. The NVE can maintain this state (the MTU size of the tunnel link(s) associated with the tunnel endpoint), so if a tenant system sends large packets that when encapsulated exceed the MTU size of the tunnel link, the tunnel endpoint can discard such packets and send exception messages to the tenant system(s). If the tunnel endpoint is associated with a routing or forwarding function and/or has the capability to send ICMP messages, the encapsulating tunnel endpoint MAY send ICMP fragmentation needed [RFC0792] or Packet Too Big [RFC4443] messages to the tenant system(s). When determining the MTU size of a tunnel link, maximum length of options MUST be assumed as options may vary on a per-packet basis. For example, recommendations/guidance for handling fragmentation 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. For example, some of the issues associated with MTU size and fragmentation in IP tunneling and use of ICMP messages is outlined in Section 4.2 of [I-D.ietf-intarea-tunnels].

#### 4.4.2. DSCP, ECN and TTL

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.

Though Uniform or Pipe models could be used for TTL (or Hop Limit in case of IPv6) handling when tunneling IP packets, the Pipe model is more aligned with network virtualization. [RFC2003] provides guidance on handling TTL between inner IP header and outer IP tunnels; this model is more aligned with the Pipe model and is RECOMMENDED for use with Geneve for network virtualization applications.

#### 4.4.3. Broadcast and Multicast

Geneve tunnels may either be point-to-point unicast between two tunnel 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 is beyond the scope of this document.

When physical multicast is in use, devices with heterogeneous capabilities may be present in the same group. Some options may only be interpretable by a subset of the devices in the group. Other devices can safely ignore such options unless the 'C' bit is set to mark the unknown option as critical. Requirements outlined in Section 3.4 apply for critical options.

In addition, [RFC8293] provides examples of various mechanisms that can be used for multicast handling in network virtualization overlay networks.

#### 4.4.4. Unidirectional Tunnels

Generally speaking, a Geneve tunnel is a unidirectional concept. IP is not a connection oriented protocol and it is possible for two tunnel 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.

#### 4.5. 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, definition of such mechanisms are beyond the scope of this document.

##### 4.5.1. Constraints on Options

While Geneve options are 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 or there may be other scenarios where the options must be processed in a certain order due to protocol semantics.

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 endpoints. In the absence of a control plane, alternative configuration mechanisms may be used for this purpose. Such mechanisms are beyond the scope of this document.

#### 4.6. 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:

- o When performing LSO, a NIC **MUST** replicate the entire Geneve header and all options, including those unknown to the device, onto each resulting segment unless an option allows an exception. 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.
- o Options **MUST NOT** be reordered during the course of offload processing, including when merging packets for the purpose of LRO.
- o NICs performing offloads **MUST NOT** drop packets with unknown options, including those marked as critical, unless explicitly configured.

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.

#### 4.7. 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 tunnel

endpoint and/or control plane and not explicitly defined as part of the data format.

## 5. Transition Considerations

Viewed exclusively from the data plane, Geneve is compatible with existing IP networks 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 builds on the base data plane functionality provided by 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 - tunnel endpoints directly communicating with each other can 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 a 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, alter or inject packets. Compromised tunnel endpoints or transit devices 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.

When crossing an untrusted link, such as the general Internet, VPN technologies such as IPsec [RFC4301] should be used to provide authentication and/or encryption of the IP packets formed as part of Geneve encapsulation (See Section 6.1.1).

Geneve does not otherwise affect the security of the encapsulated packets. As per the guidelines of BCP 72 [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 NVEs 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, this may or not may be the same provider as an underlay service provider. 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) for example by other tenant systems or underlay service provider. A compromised network node or a transit device within a data center may passively monitor Geneve packet data between NVEs; or route traffic for further inspection. 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. The overlay provider is expected to provide cryptographic protection in cases where the underlay provider is not the same as the overlay provider to ensure the payload is not exposed to the underlay.

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.

#### 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 SHOULD be secured by encryption mechanisms such as IPsec or other VPN technologies to protect the communications between the NVEs when they are geographically separated over untrusted network links. Specification of data protection mechanisms employed between data centers is beyond the scope of this document.

The principles described in Section 4 regarding controlled environments still apply to the geographically separated data center usage outlined in this section.

## 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 trying to inject unauthorized Geneve encapsulated traffic such as spoofing, replay, etc., 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 tenant systems in different tenant networks. An NVE MUST NOT interpret Geneve packets from tenant systems other than as frames to be encapsulated.

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, although non-cryptographic mechanisms may not protect the Geneve portion of the packet from tampering.

### 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. Options Interpretation by Transit Devices

Options, if present in the packet, are generated and terminated by tunnel endpoints. As indicated in Section 2.2.1, transit devices may interpret the options. However, if the packet is protected by tunnel endpoint to tunnel endpoint encryption, for example through IPsec, transit devices will not have visibility into the Geneve header or options in the packet. In such cases transit devices MUST handle Geneve packets as any other IP packet and maintain consistent forwarding behavior. In cases where options are interpreted by transit devices, the operator MUST ensure that transit devices are trusted and not compromised. The definition of a mechanism to ensure this trust is beyond the scope of this document.

### 6.5. 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 tunnel 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.

### 6.6. 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 are beyond the scope of this document.

## 7. IANA Considerations

IANA has allocated UDP port 6081 in the Service Name and Transport Protocol Port Number Registry [IANA-SN] as the well-known destination port for Geneve based on early registration.

Upon publication of this document, this registration will have its reference changed to cite this document [RFC-to-be] and inline with [RFC6335] the assignee and contact of the port entry should be changed to IESG <iesg@ietf.org> and IETF Chair <chair@ietf.org> respectively:

```
Service Name: geneve
Transport Protocol(s): UDP
Assignee: IESG <iesg@ietf.org>
Contact: IETF Chair <chair@ietf.org>
Description: Generic Network Virtualization Encapsulation (Geneve)
Reference: [RFC-to-be]
Port Number: 6081
```

In addition, IANA is requested to create a new "Geneve Option Class" registry to allocate Option Classes. This registry is to be placed under a new Network Virtualization Overlay (NVO3) protocols page (to be created) in IANA protocol registries [IANA-PR]. The Geneve Option Class registry shall consist of 16-bit hexadecimal values along with descriptive strings, assignee/contact information and references. The registration rules for the new registry are (as defined by [RFC8126]):

Range	Registration Procedures
0x0000..0x00FF	IETF Review
0x0100..0xFEFF	First Come First Served
0xFF00..0xFFFF	Experimental Use

Initial registrations in the new registry are as follows:

Option Class	Description	Assignee/Contact	References
0x0100	Linux		
0x0101	Open vSwitch (OVS)		
0x0102	Open Virtual Networking (OVN)		
0x0103	In-band Network Telemetry (INT)		
0x0104	VMware, Inc.		
0x0105	Amazon.com, Inc.		
0x0106	Cisco Systems, Inc.		
0x0107	Oracle Corporation		
0x0108..0x0110	Amazon.com, Inc.		

## 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

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

Email: kduda@arista.com

Dinesh G. Dutt  
Independent

Email: didutt@gmail.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 acknowledge Puneet Agarwal, David Black, Sami Boutros, Scott Bradner, Martin Casado, Alissa Cooper, Roman Danyliw, Bruce Davie, Anoop Ghanwani, Benjamin Kaduk, Suresh Krishnan, Mirja Kuhlewind, Barry Leiba, Daniel Migault, Greg Mirksy, Tal Mizrahi,

Kathleen Moriarty, Magnus Nystrom, Adam Roach, Sabrina Tanamal, Dave Thaler, Eric Vyncke, Magnus Westerlund and many other members of the NVO3 WG for their reviews, comments and suggestions.

The authors would like to thank Sam Aldrin, Alia Atlas, Matthew Bocci, Benson Schliesser, and Martin Vigoureux for their guidance throughout the process.

## 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>>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.
- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <<https://www.rfc-editor.org/info/rfc1112>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <<https://www.rfc-editor.org/info/rfc2003>>.
- [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>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <<https://www.rfc-editor.org/info/rfc6040>>.

- [RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <<https://www.rfc-editor.org/info/rfc6936>>.
- [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>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.

## 10.2. Informative References

- [ETYPES] The IEEE Registration Authority, "IEEE 802 Numbers", <<https://www.iana.org/assignments/ieee-802-numbers>>.
- [I-D.ietf-intarea-tunnels]  
Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", draft-ietf-intarea-tunnels-10 (work in progress), September 2019.
- [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., "NVO3 Encapsulation Considerations", draft-ietf-nvo3-encap-05 (work in progress), February 2020.
- [IANA-PR] IANA, "Protocol Registries", <<https://www.iana.org/protocols>>.
- [IANA-SN] IANA, "Service Name and Transport Protocol Port Number Registry", <<https://www.iana.org/assignments/service-names-port-numbers>>.
- [IEEE.802.1Q\_2018] IEEE, "IEEE Standard for Local and Metropolitan Area Networks--Bridges and Bridged Networks", IEEE 802.1Q-2018, DOI 10.1109/ieeestd.2018.8403927, July 2018, <<http://ieeexplore.ieee.org/servlet/opac?punumber=8403925>>.
- [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>>.

- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [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>>.
- [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>>.
- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<https://www.rfc-editor.org/info/rfc8086>>.
- [RFC8293] Ghanwani, A., Dunbar, L., McBride, M., Bannai, V., and R. Krishnan, "A Framework for Multicast in Network Virtualization over Layer 3", RFC 8293, DOI 10.17487/RFC8293, January 2018, <<https://www.rfc-editor.org/info/rfc8293>>.
- [VL2] "VL2: A Scalable and Flexible Data Center Network", ACM SIGCOMM Computer Communication Review, DOI 10.1145/1594977.1592576, 2009, <<https://www.sigcomm.org/sites/default/files/ccr/papers/2009/October/1594977-1592576.pdf>>.

Authors' Addresses

Jesse Gross (editor)

Email: [jesse@kernel.org](mailto:jesse@kernel.org)

Ilango Ganga (editor)

Intel Corporation

2200 Mission College Blvd.

Santa Clara, CA 95054

USA

Email: [ilango.s.ganga@intel.com](mailto:ilango.s.ganga@intel.com)

T. Sridhar (editor)

VMware, Inc.

3401 Hillview Ave.

Palo Alto, CA 94304

USA

Email: [tsridhar@vmware.com](mailto:tsridhar@vmware.com)



NVO3  
Internet-Draft  
Intended status: Informational  
Expires: August 30, 2018

D. Migault  
Ericsson  
S. Boutros  
D. Wing  
VMware  
S. Krishnan  
Kaloom  
February 26, 2018

Geneve Protocol Security Requirements  
draft-mglt-nvo3-geneve-security-requirements-03

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).

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 August 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Requirements Notation . . . . .	2
2. Introduction . . . . .	2
3. Terminology . . . . .	5
4. Security Threats . . . . .	5
4.1. Passive Attacks . . . . .	6
4.2. Active Attacks . . . . .	6
5. Requirements for Security Mitigations . . . . .	7
5.1. Protection Against Traffic Sniffing . . . . .	7
5.2. Protecting Against Traffic Injection . . . . .	8
5.3. Protecting Against Traffic Redirection . . . . .	10
5.4. Protecting Against Traffic Replay . . . . .	12
6. IANA Considerations . . . . .	12
7. Security Considerations . . . . .	13
8. Acknowledgments . . . . .	13
9. References . . . . .	14
9.1. Normative References . . . . .	14
9.2. Informational References . . . . .	14
Authors' Addresses . . . . .	15

## 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 Geneve Transit Nodes (GTN). 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.

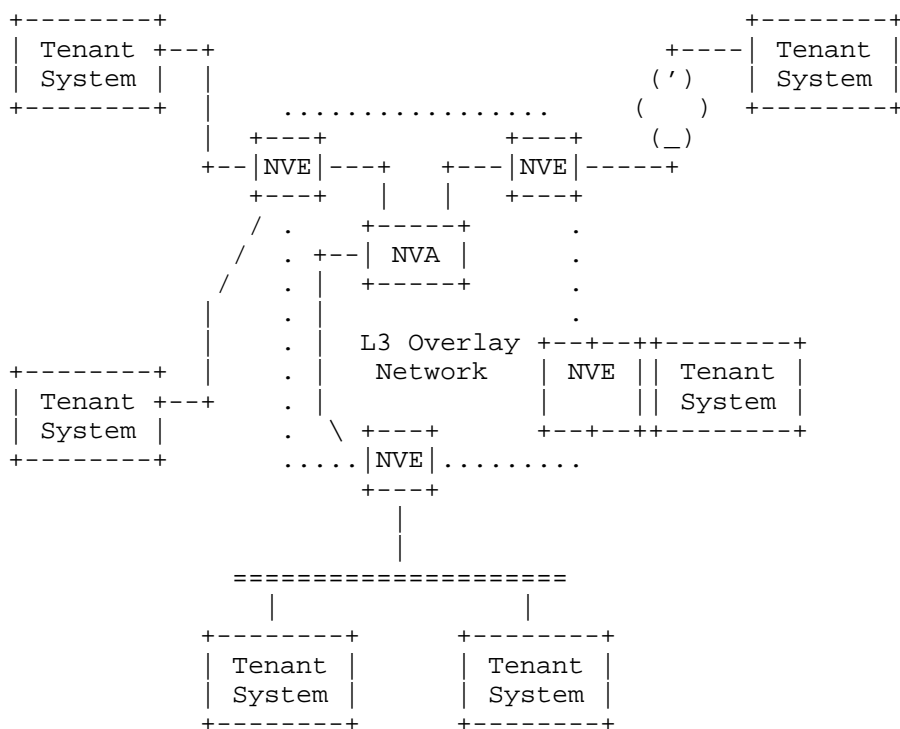


Figure 1: Generic Reference Model for Network Virtualization Overlays  
[RFC7365]

This document discusses the security risks that a Geneve based NVO3 network may encounter and tries to provide a list of essential security requirements that needs to be fulfilled. 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. Protecting the complete Geneve packet - that is the full IP packet or the full outer UDP payload for example - is out of

scope of this document, given that this can be supported using existing mechanisms.

This document assumes that a tenant subscribes to an overlay cloud provider for hosting its Tenant Systems, the cloud provider manages the Geneve overlay network on behalf of the tenant. The overlay network will be hosted on an underlay network infrastructure, that may be managed by another underlay cloud provider.

The security requirements in this document aims at providing the overlay cloud provider the necessary options to ensure:

1. Delivering tenant payload traffic and ensuring privacy and integrity of the overlay traffic, and isolation between the overlay and underlay networks, as well preventing tenant traffic from being redirected or injected to other tenants.
2. Protecting tenant traffic from rogue devices in the providers of Geneve overlay or underlay networks.

In summary, 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 Geneve.

The security requirements in the document are expressed regarding the threats to mitigate. It is expected that a security mechanism designed for NVO3 overlay network implementing Geneve is able to mitigate all the threats and as such fulfills all the security requirements expressed in the document. The document RECOMMENDS that the definition of a Geneve security mechanism fulfills all requirements expressed in this document

On the other hand, the specificities and the context of some Geneve deployments may consider the risk associated to some threats as very low and as such ignore the threats. In such cases, a specific security mechanism designed for that specific deployment may not fulfill all the requirements associated to that given threat. This document RECOMMENDS to consider all the threats while designing a security mechanism for the Geneve overlay network. In addition, some deployments may not take advantage of some features provided by Geneve, in which case, a specific security mechanisms designed for a specific deployment may not fulfill the requirements associated to that feature. This document RECOMMENDS that such specific security mechanisms be an intermediary approach toward the deployment of a Geneve security mechanism. In fact, such specific mechanisms present the risk to ossifying Geneve as well as the security being lowered in favor of Geneve features.

The document strongly recommend to re-use existing security protocols like IP Security (IPsec) [RFC4301] and Transport Layer Security (TLS) [RFC5246], and existing encryption algorithms ( such as [RFC8221]), and authentication protocols.

Authentication requirements for NVO3 devices, automated key management, as well as packet level security providing confidentiality, integrity and authorization requirements defined in [I-D.ietf-nvo3-security-requirements] are also requirements for this document.

### 3. Terminology

This document uses the terminology of [RFC8014], [RFC7365] and [I-D.ietf-nvo3-geneve]. In addition to these document the following term is used:

- o Immutable Geneve Option: designate a Geneve Option that are not expected to be modified by any on path element, such as a GTN.
- o Geneve Transit Node (GTN): A transit device that is not Geneve termination point. GTN MAY understand and Geneve packet and MAY process Geneve Option.

### 4. Security Threats

Attacks from compromised NVO3 and underlay network devices, and attacks from compromised tenant systems defined in [I-D.ietf-nvo3-security-requirements] are considered for the Geneve overlay network. Furthermore, 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 packet inner payload.

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-identifiable information or protocol information (e.g., TLS certificate, overlay routing protocols)).

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 and the security requirements 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 packets, injecting packets, or interfering with packet delivery (such as by corrupting packet checksum).

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 - GTN, 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

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. Comparing application TLS to guest IPsec ESP to NVE IPsec ESP, each provides stronger protection from traffic analysis in

the same order. Application TLS exposes TCP port numbers to a passive observer, guest IPsec ESP encrypts the inner transport header but still identifies the communicating VM's IP address, while NVE IPsec ESP encrypts the entire inner payload.

To protect packet payloads from passive listeners, application-level encryption (e.g., JSON Web Encryption [RFC7516]), application TLS, guest IPsec ESP, or hypervisor IPsec ESP can be used. Each provides the same protection against a passive listener.

To protect against the above-described traffic sniffing attacks, we require:

GEN-REQ1: The NVE MUST ensure the traffic leaving the NVE has its payload encrypted. The encryption operation MAY be performed by the NVE, but could also be performed, for example, by the TS.

GEN-REQ2: To provide best protection from traffic analysis, the NVE SHOULD encrypt the payload fields that appears in clear. Typically, this could include VM's inner IP address, transport header, and IP payload when Geneve carries IP packets.

GEN-REQ1 and GEN-REQ2 are inline with NVE-NVE and NVE-Hypervisor data plane security requirements for confidentiality in [I-D.ietf-nvo3-security-requirements] like REQ 10, 11 and 16.

## 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 System are not protecting their communications, the Geneve overlay network SHOULD be able 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. Authenticated Geneve Packet MAY be checked by underlay intermediary nodes.

Based on a policy partial authentication MAY be performed on Geneve packets if tenant's system is protecting it's communication. In situations where the tenant system is already encrypting its traffic with application-level encryption (e.g., S/MIME), transport encryption (e.g., TLS), or IP encryption (e.g., IPsec ESP), it is redundant for the NVE to apply additional encryption. Note that relaying on upper security layers, results from a compromise between security and performance as it may introduce cut and paste vulnerability.

The Geneve architecture considers intermediary nodes designated as GTN. A protection established between NVE SHOULD NOT prevent GTN to perform their operations, such as the insertion of a Geneve Option, authenticating a Geneve Option or steering Geneve packets. In the later case, in order to ease the transition from a non secured to secure Geneve overlay network, it is expected that GTN that are not aware of Geneve security mechanisms can steer authenticated Geneve packets the same way as non protected Geneve packets. Similarly, the transition from non secure to secure Geneve overlay network may also be performed by introducing GTN that performs the security functionalities - such as authentication of Geneve packets- on behalf of NVE.

This leads to the following security requirements:

- GEN-REQ3: A Geneve NVE MUST be able to authenticate at least one of the Geneve tunnel Header, the Geneve base header, the immutable Geneve Options, or the Geneve payload. The combination of fields that are authenticated is defined by security policies.
- GEN-REQ4: A Geneve NVE MAY be able to authenticate only a portion of the Geneve payload if the Tenant's system is protecting its communication.
- GEN-REQ5: A GTN MAY be able to validate the authentication before the packet reaches the Geneve destination NVE.
- GEN-REQ6: A GTN MUST be able to insert an authenticated Geneve Option into a authenticated Geneve Packet - protected by the source Geneve NVE.

- GEN-REQ7: A GTN capable of forwarding non-authenticated Geneve packets MUST be capable of forwarding the Geneve authenticated packet without any additional security specific functionalities. In other words, forwarding authenticated Geneve packet MUST done the same way as authenticated Geneve packets.
- GEN-REQ8: A Geneve NVE SHOULD be able to set 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).
- GEN-REQ9: In the case when Tenant systems secure their communications using protocols such as TLS or IPsec. A Geneve NVE MAY be able to selectively encrypt and/or authenticate only the sections that are not encrypted/authenticated by the Tenant System. For example, only the IP, transport (TCP / UDP) in case of TLS/DTLS MAY be encrypted/authenticated, while only the IP header and ESP header MAY be encrypted/authenticated.

Requirements listed in this section are inline with authentication and integrity requirements in [I-D.ietf-nvo3-security-requirements], like REQ 9, 10, 11, 14 and 16.

The requirements further define mechanisms to fully and partially authenticate Geneve Header, and Geneve options, as well fully and partially encrypt the same.

### 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 GTN that are in charge of encrypting and decrypting a Geneve Packet, Geneve Option or any information. The security mechanisms are intended to protect a Geneve information from any on path node.

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.

Based on a policy partial encryption MAY be performed on Geneve packets if tenant's system is protecting it's communication.

The Geneve architecture considers intermediary nodes designated as GTN. A protection established between NVE SHOULD NOT prevent GTN to perform their operations, such as the insertion of a Geneve Option, encrypting a Geneve Option or steering Geneve packets. In the later case, in order to ease the transition from a non secured to secure Geneve overlay network, it is expected that GTN that are not aware of Geneve security mechanisms can steer encrypted Geneve packets the same way as non protected Geneve packets. Similarly, the transition from non secure to secure Geneve overlay network may also be performed by introducing GTN that performs the security functionalities - such as encryption of Geneve packets- on behalf of NVE.

This leads to the following security requirements:

GEN-REQ10: A Geneve NVE MUST be able encrypt Geneve base Header, and/or Geneve Payload and/or Geneve Options not intended for the GTN.

GEN-REQ11: A Geneve NVE MAY be able encrypt portion of Geneve Payload as well as as Geneve Options not intended for the GTN.

GEN-REQ12: A transit underlay intermediary node MUST be able to insert an encrypted Geneve Option into an encrypted/ authenticated Geneve Packet - protected by the source Geneve NVE.

GEN-REQ13: A Geneve NVE SHOULD be able to assign different cryptographic keys to protect the unicast tunnels between NVEs respectively.

GEN-REQ14: If there are multicast packets, a Geneve NVE SHOULD 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.

Requirements listed in this section are inline with the requirements in the data plane sections in [I-D.ietf-nvo3-security-requirements] to protect against traffic redirection and man in the middle attacks.

The requirements further define mechanisms for a transit intermediary node to insert an encrypted Geneve option to an encrypted/authenticated Geneve packet.

#### 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 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.

This leads to the following security requirements:

GEN-REQ15: A Geneve NVE or a GTN MUST be able to validate the Geneve Header corresponds to the Geneve payload, and discard such packets.

GEN-REQ16: A Geneve NVE or a GTN SHOULD provide anti replay mechanisms and discard replayed packet.

The requirements in this section are inline with REQ 10 and 14 in [I-D.ietf-nvo3-security-requirements], and they further specifies requirements to validate that a Geneve Header corresponds to the Geneve payload.

#### 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. Their purpose is to design appropriated Geneve security Options or to appropriately secure NVE-NVE communication based on Geneve. 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 adress the security needs upon deploying an Geneve overlay network.

More precisely, REQ 1 to REQ 8 are focused on the control plane which is outside the scope of this document.

REQ 9 is a data plane security requirement, but focused on the establishment of a NVO3 tunnels. This is outside the scope of Geneve which only address data in motion. As such REQ 9 is outside the scope of this document.

REQ 10 to REQ 14 are in the scope of this document. REQ 12 and REQ 13 are identical as GEN-REQ13 and GEN-REQ14. All other requirements from GEN-REQ1 to GEN-REQ16 are declinasons of REQ 10, REQ 11 and REQ 14. These requirements are the declination of architecture requirements in a context for Geneve, which includes the presence of GTN, Geneve Options as well as the possibility to split the security oporation between tenants and teh overlay infrastructure.

REQ 15 to REQ 18 from [I-D.ietf-nvo3-security-requirements] are focused on the NVE-Hypervisor Data Plane which is not based on Geneve and thus is outside the scope of the document.

## 8. 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.

## 9. References

### 9.1. Normative References

- [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>>.
- [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>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [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>>.

### 9.2. Informational References

- [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.
- [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>>.

Authors' Addresses

Daniel Migault  
Ericsson  
8400 boulevard Decarie  
Montreal, QC H4P 2N2  
Canada

Email: [daniel.migault@ericsson.com](mailto:daniel.migault@ericsson.com)

Sami Boutros  
VMware, Inc.

Email: [sboutros@vmware.com](mailto:sboutros@vmware.com)

Dan Wing  
VMware, Inc.

Email: [dwing@vmware.com](mailto:dwing@vmware.com)

Suresh Krishnan  
Kaloom

Email: [suresh@kaloom.com](mailto:suresh@kaloom.com)

NVO3  
Internet-Draft  
Intended status: Informational  
Expires: September 1, 2019

D. Migault  
Ericsson  
S. Boutros  
D. Wings  
VMware, Inc.  
S. Krishnan  
Kaloom  
February 28, 2019

Geneve Security Requirements  
draft-mglt-nvo3-geneve-security-requirements-06

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 evaluate a given deployment. 2. requirement a security mechanism need to fulfill to secure any deployment of Geneve overlay deployment

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 1, 2019.



## Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Requirements Notation . . . . .	3
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
5.1.1. Operational Security Requirements . . . . .	9
5.1.2. Geneve Security Requirements . . . . .	10
5.2. Protecting Against Traffic Injection . . . . .	10
5.2.1. Operational Security Requirements . . . . .	12
5.2.2. Geneve Security Requirements . . . . .	13
5.3. Protecting Against Traffic Redirection . . . . .	13
5.4. Protecting Against Traffic Replay . . . . .	14
5.4.1. Geneve Security Requirements . . . . .	14
5.4.2. Geneve Security Requirements . . . . .	15
5.5. Security Management . . . . .	15
5.5.1. Operational Security Requirements . . . . .	15
5.5.2. Geneve Security Requirements . . . . .	15
6. IANA Considerations . . . . .	16
7. Security Considerations . . . . .	16
8. Appendix . . . . .	16
8.1. DTLS . . . . .	16
8.1.1. Operational Security Requirements . . . . .	16
8.1.2. Geneve Security Requirements . . . . .	18
8.2. IPsec . . . . .	20
8.2.1. Operational Security Requirements . . . . .	20
8.2.2. Geneve Security Requirements . . . . .	21
9. Acknowledgments . . . . .	23
10. References . . . . .	23

10.1. Normative References . . . . .	23
10.2. Informative References . . . . .	25
Authors' Addresses . . . . .	25

## 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.

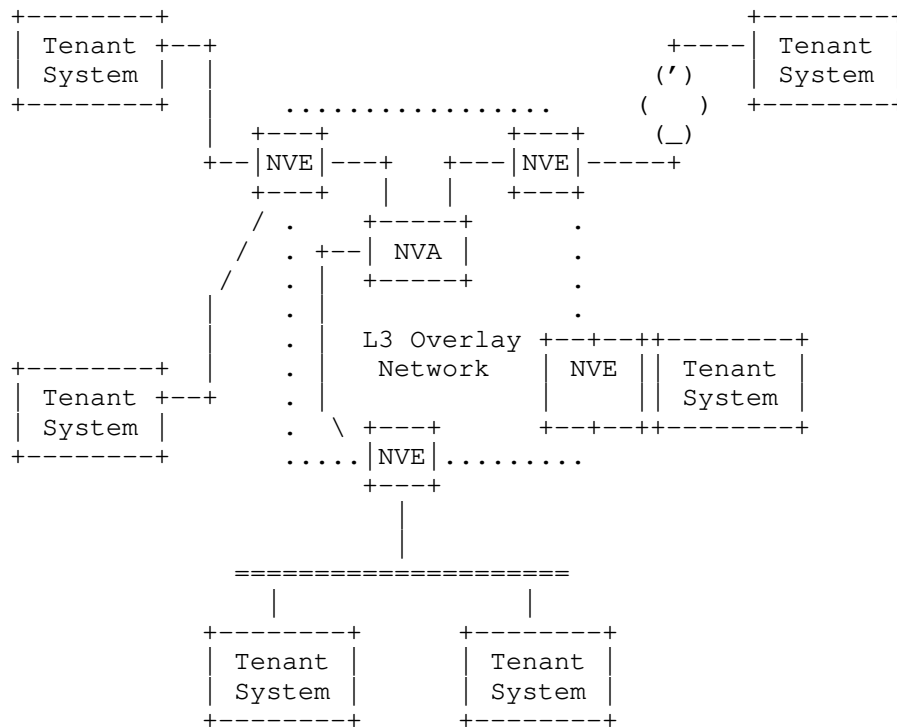


Figure 1: Generic Reference Model for Network Virtualization Overlays [RFC7365]

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 is limited to the security of the data plane only. Additional requirements for the control plane 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 raise 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-identifiable 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. The algorithm used MUST follow the cryptographic guidance such as [RFC8247], [RFC8221] or [RFC7525]. In this context, when the document mentions encryption, it assumes authenticated encryption.

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 be 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. As a result, the confidentiality protection of the inner packet may be handled either entirely by the Geneve Overlay Provider, or partially by the Tenant or handled by both the Tenant and the Geneve Overlay Provider.

The Geneve Header contains information related to the Geneve communications or metadata designated as Geneve Information. Geneve Information is carried on the Geneve Outer Header, the Geneve Header (excluding Geneve Options) as well as in the Geneve Options. Geneve Information needs to be accessed solely by a NVE or transit device while other Geneve Information may need to be accessed by other transit devices. More specifically, a subset of the information contained in the Geneve Header (excluding Geneve Options) as well as a subset of (none, one or multiple Geneve Option) may be accessed by a transit device or the NVE while the others needs to be accessed by other transit devices. The confidentiality protection of the Geneve Information is handled by the Geneve Overlay Provider.

In addition to Geneve Information, the traffic generated for the Geneve overlay may be exposed to traffic volumetry and pattern analysis within a virtualized network. Confidentiality protection against traffic pattern recognition is handled by the Geneve Overlay Provider.

#### 5.1.1.1. Operational Security Requirements

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

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



- o 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.

#### 5.1.2. Geneve Security Requirements

A Geneve security mechanism must fulfill the requirements below:

- o SEC-GEN-1: Geneve security mechanism MUST provide the capability to encrypt the inner payload.
- o SEC-GEN-2: Geneve security mechanism SHOULD provide the capability to partially encrypt the inner payload header.
- o SEC-GEN-3: Geneve security mechanism MUST provide means to encrypt a single or a set of zero, one or multiple Geneve Options while leave other Geneve Options in clear. Reversely, a Geneve security mechanism MUST be able to leave a Geneve option in clear, while encrypting the others.
- o SEC-GEN-4: Geneve security mechanism MUST provide means to encrypt the information of Geneve Header (excluding Geneve Options). Reversely, a Geneve security mechanism MUST be able to leave in clear Geneve Header information (Geneve Options excluded) while encrypting the other.
- o SEC-GEN-5: Geneve security mechanisms MUST provide the ability to provide confidentiality protection between multiple nodes, i.e. multiple transit devices and a NVE.
- o SEC-GEN-6: Geneve security mechanism MUST provide the ability to pad a Geneve packet.
- o SEC-GEN-7: 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.

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. Tenant's System may provide integrity protection of the inner payload by protect their communications using for example IPsec/ESP, IPsec/AH [RFC4302], TLS or DTLS. Such protection protects at various layers the Tenants from receiving spoofed packets, as any injected packet is expected to be discarded by the destination Tenant's System. Note IPsec/ESP with NULL encryption may be used to authenticate-only the layers above IP in which case the IP header remains unprotected. However IPsec/AH enables the protection of the entire IP packet, including the IP header. As a result, when Geneve encapsulates IP packets the Tenant has the ability to integrity protect the IP packet on its own, without relying on the Geneve overlay network. On the other hand, L2 layers remains unprotected. As encryption is using authenticated encryption, authentication may also be provided via encryption. At the time of writing the document DTLS 1.3 [I-D.ietf-tls-dtls13] is still a draft document and TLS 1.3 does not yet provide the ability for authenticate only the traffic. As such it is likely that the use of DTLS1.3 may not involve authentication-only cipher suites. Similarly to confidentiality protection, integrity protection may be handled either entirely by the Geneve Overlay Provider, or partially by the Tenant or handled by both the Tenant and the Geneve Overlay Provider.

In addition to confidentiality protection of the inner payload, integrity protection also prevents the Tenant System from receiving illegitimate packets that may disrupt the Tenant's System performance. The Geneve overlay network need to prevent the overlay to be used as a vector to spoof packets being steered to the Tenant's system. As a result, the Overlay Network Provider needs to ensure that inner packets steered to the Tenant's network are only originating from one Tenant System and not from an outsider using the Geneve Overlay to inject packets to one virtual network. As such, the destination NVE MUST be able to authenticate the incoming Geneve packets from the source NVE. This may be performed by the NVE authenticating the full Geneve Packet. When the Geneve Overlay wants to take advantage of the authentication performed by the Tenant System, the NVE should be able to perform some checks between the Geneve Header and the inner payload. Suppose two Geneve packets are composed of a Geneve Header (H1, and H2) and a inner payload (P1 and P2). Suppose H1, H2, P1 and P2 are authenticated. The replacement of P2 by P1 by an attacker will be detected by the NVE only if there is a binding between H2 and P2. Such integrity protection is handled by the Geneve Overlay Provider.

While traffic injection may target the Tenant's virtual network or a specific Tenant System, traffic injection may also target the Geneve Overlay Network by injecting Geneve Options that will affect the processing of the Geneve Packet. 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. As such, the Geneve Overlay should provide integrity protection of the Geneve Information present in the Geneve Header to guarantee Geneve processing is not altered.

The Geneve architecture considers transit devices that may process some Geneve Options. More specifically, a Geneve packet may have A subset of Geneve Information of the Geneve Header (excluding Geneve Options) as well as a set of zero, one or multiple of Geneve Options accessed by one or more transit devices. This information needs to be authenticated by a transit device while other options may be authenticated by other transit devices or the tunnel endpoint. The integrity protection is handled by the Geneve Overlay Provider and authentication MUST be performed prior any processing.

#### 5.2.1. Operational Security Requirements

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

- o SEC-OP-4: A secure deployment of a Geneve overlay MUST provide the capability authenticate the inner payload when encryption is not provided. A Geneve overlay provider MAY disable this capability for example when this is performed by the Tenant System and that level of integrity is believed to be sufficient. In order to provide additional protection to traffic already protected by the Tenant the Geneve network operator MAY partially protect the unprotected part of the inner payload.
- o SEC-OP-5: A secure deployment of a Geneve overlay MUST evaluate the risk associated to a change of the Geneve Outer Header, Geneve Header (excluding Geneve Options) and Geneve Option. When a risk analysis concludes that the risk is too high, this piece of information MUST be authenticated.
- o SEC-OP-6: 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. Instead, it is RECOMMENDED that mechanisms binds the inner payload to the Geneve Header. To prevent injection between virtualized network, it is strongly

RECOMMENDED that at least the Geneve Header without Geneve Options is authenticated.

- o SEC-OP-7: 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.

#### 5.2.2. Geneve Security Requirements

A Geneve security mechanism must fulfill the requirements below:

- o SEC-GEN-8: Geneve security mechanism MUST provide the capability to authenticate the inner payload.
- o SEC-GEN-9: Geneve security mechanism SHOULD provide the capability to partially authenticate the inner payload header.
- o SEC-GEN-10: Geneve security mechanism MUST provide the capability to authenticate a single or a set of options while leave other Geneve Option unauthenticated. Reversely, a Geneve security mechanism MUST be able to leave a Geneve option unauthenticated, while encrypting the others.
- o SEC-GEN-11: Geneve security mechanism MUST provide means to authenticate the information of Geneve Header (Geneve Option excluded). Reversely, a Geneve security mechanism MUST be able to leave unauthenticated Geneve header information (Geneve Options excluded) while authenticating the other.
- o SEC-GEN-12: Geneve Security mechanism MUST provide means for a tunnel endpoint (NVE) to authenticate data prior it is being processed.
- o SEC-GEN-13: Geneve Security mechanism MUST provide means for a transit device to authenticate data prior it is being processed.

#### 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 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 needs to provide confidentiality protection for data traffic 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 Tenant System is not protected against anti-replay. A packet even authenticated can be replayed. DTLS and IPsec provides anti replay mechanisms, so it is unlikely that authenticated Tenant's traffic is subject to replay attacks.

Similarly to integrity protection, the Geneve Overlay Provider should prevent the overlay to be used to replay packet to the Tenant's System. In addition, similarly to integrity protection, the Geneve Overlay network may also be a target of a replay attack, and NVE as well as transit devices should benefit from the same protection.

Given the proximity between authentication and anti-replay mechanisms and that most authentication mechanisms integrates anti-replay attacks, we RECOMMEND that authentication contains an anti-replay mechanisms.

##### 5.4.1. Geneve Security Requirements

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

- o SEC-OP-8: 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 authenticated with anti replay protection.

#### 5.4.2. Geneve Security Requirements

A Geneve security mechanism must fulfill the requirements below:

- o SEC-GEN-14: Geneve Security mechanism MUST provide authentication with anti-replay protection.

#### 5.5. Security Management

##### 5.5.1. Operational Security Requirements

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

- o SEC-OP-9: A secure deployment of a Geneve overlay MUST define the security policies that associates the encryption, and authentication associated to each flow between NVEs.
- o SEC-OP-10: 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.

##### 5.5.2. Geneve Security Requirements

A Geneve security mechanism must fulfill the requirements below:

- o SEC-GEN-15: 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).
- o SEC-GEN-16: A Geneve security mechanism MUST be able to assign different cryptographic keys to protect the unicast tunnels between NVEs respectively.
- o SEC-GEN-17: 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. DTLS

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

The use of DTLS MAY fill the security requirements for a secure Geneve deployment. However DTLS cannot be considered as the Geneve security mechanism enabling all Geneve deployments. To ease the reading of the Requirements met by DTLS or IPsec, the requirements list indicates with Y (Yes) when the requirement and N (No) when the requirement is not met. In addition, an explanation is provided on the reasoning. This section is not normative and its purpose is limited to illustrative purpose.

#### 8.1.1. Operational Security Requirements

This section shows how DTLS may secure some Geneve deployments. Some Geneve deployments may not be secured by DTLS, but that does not exclude DTLS from being used.

- o SEC-OP-1 (Y): A deployment using DTLS between NVEs with an non NULL encryption cipher suite will provide confidentiality to the full Geneve Packet which contains the inner payload. As such the use of DTLS meets SEC-OP-1. Note that DTLS does not provide partial encryption and as such the Geneve Overlay Provider may not benefit from the encryption performed by the Tenant if performed, which may result in some portion of the payload being encrypted twice.
- o SEC-OP-2 (Y): A deployment using DTLS between NVEs with an non NULL encryption cipher suite encrypt the Geneve Packet which includes the Geneve Header and associated metadata. Only the UDP port is leaked which could be acceptable. As such, the use of DTLS meets SEC-OP-2.
- o SEC-OP-3 (Y/N): A deployment using DTLS between NVEs will not be able to send dummy packets or pad Geneve Packet unless this is managed by the Geneve packet itself. DTLS does not provide the ability to send dummy traffic, nor to pad. As a result DTLS itself does not meet this requirement. This requirement may be met if handled by the Geneve protocol. As such SEC-OP-3 may not be met for some the deployment. However, it is not a mandatory requirement and as such it is likely that the use of DTLS SEC-OP-3 is met.
- o SEC-OP-4 (Y): Similarly to SEC-OP-1, A deployment using DTLS between NVEs provides integrity protection to the full Geneve Packet which includes the inner payload. As such the use of DTLS meets SEC-OP-4. Note that DTLS 1.2 provides integrity-only cipher suites while DTLS 1.3 does not yet. As a result, the use of DTLS 1.3 may provide integrity protection using authenticated encryption.
- o SEC-OP-5 (Y): Similarly to SEC-OP-2, A deployment using DTLS between NVE authenticates the full Geneve Packet which includes the Geneve Header. Only the UDP port is left unauthenticated. As such, the use of DTLS meets SEC-OP-5.
- o SEC-OP-6 (Y): A deployment using DTLS between NVE authenticates NVE-to-NVE communications and the use of DTLS meets SEC-OP-6.
- o SEC-OP-7 (Y/N): A deployment using DTLS between NVEs is not compatible with a Geneve architecture that includes transit devices. When the DTLS session uses a non NULL encryption cipher suite, the transit device will not be able to access it. When the NULL encryption cipher suite is used, the transit device may be able to access the data, but will not be able to authenticate it prior to processing the packet. As such the use of DTLS only



meets SEC-OP-7 for deployment that do not include any transit devices.

- o SEC-OP-8 (Y): A deployment using DTLS between NVEs provides anti-replay protection and so, the use of DTLS meets SEC-OP-8.
- o SEC-OP-9 (Y/N): DTLS does not define any policies. Instead DTLS process is bound to an UDP socket. As such handling of flow policies is handled outside the scope of DTLS. As such SEC-OP-9 is met outside the scope of DTLS.
- o SEC-OP-10 (N): DTLS session may be established with specific material, as such it is possible to assign different material for each flow. However, the binding between flow and session is performed outside the scope of DTLS. In addition, DTLS does not support multicast. As such, the use of DTLS may only meets SEC-OP-10 in the case of unicast communications.

#### 8.1.2. Geneve Security Requirements

This section shows that DTLS cannot be used as a generic Geneve security mechanism to secure Geneve deployments. A Geneve security mechanism would need to meet all SEC-GEN requirements.

- o SEC-GEN-1 (Y): A deployment using DTLS between NVEs with an non NULL encryption cipher suite will provide confidentiality to the full Geneve Packet which contains the inner payload. As such the use of DTLS meets SEC-GEN-1.
- o SEC-GEN-2 (Y): A deployment using DTLS between NVEs with an non NULL encryption cipher suite will not be able to partially encrypt the inner payload header. However such requirement is not set a mandatory so the use of DTLS meets SEC-GEN-2
- o SEC-GEN-3 (N): A deployment using DTLS between NVEs with an non NULL encryption cipher suite encrypt the Geneve Packet which includes the Geneve Header and all Geneve Options. However DTLS does not provides any means to selectively encrypt or leave in clear text a subset of Geneve Options. As a result the use of DTLS does not meet SEC-GEN-3.
- o SEC-GEN-4 (N): A deployment using DTLS between NVEs with an non NULL encryption cipher suite encrypt the Geneve Packet which includes the Geneve Header and all Geneve Options. However, DTLS does not provides means to selectively encrypt some information of the Geneve Header. As such the use of DTLS does not meet SEC-GEN-5.

- o SEC-GEN-5 (N): A deployment using DTLS between NVEs with an non NULL encryption cipher suite provides end-to-end security between the NVEs and as such does not permit the interaction of one or multiple on-path transit devices. As such the use of DTLS does not meet SEC-GEN-5.
- o SEC-GEN-6 (N): A deployment using DTLS between NVEs with an non NULL encryption cipher suite does not provide padding facilities. This requirements is not met by DTLS itself and needs to be handled by Geneve and specific options. As a result, the use of DTLS does not meet SEC-GEN-6
- o SEC-GEN-7 (N): A deployment using DTLS between NVEs with an non NULL encryption cipher suite does not provide the ability to send dummy packets. This requirements is not met by DTLS itself and needs to be handled by Geneve and specific options. As a result, the use of DTLS does not meet SEC-GEN-7.
- o SEC-GEN-8 (Y): A deployment using DTLS between NVEs with an non NULL encryption cipher suite or a NULL encryption cipher suite provide authentication of the inner payload. As such the use of DTLS meets SEC-GEN-8.
- o SEC-GEN-9 (Y): A deployment using DTLS between NVEs does not provide the ability to partially authenticate the inner payload header. However such requirement is not set a mandatory so the use of DTLS meets SEC-GEN-9
- o SEC-GEN-10 (N): A deployment using DTLS between NVEs authenticates the Geneve Packet which includes the Geneve Header and all Geneve Options. However, DTLS does not provides means to selectively encrypt some information of the Geneve Header. As such the use of DTLS meets SEC-GEN-10.
- o SEC-GEN-11 (N): A deployment using DTLS between NVEs authenticates the Geneve Packet which includes the Geneve Header and all Geneve Options. However, DTLS does not provides means to selectively authenticate some information of the Geneve Header. As such the use of DTLS does not meet SEC-GEN-11.
- o SEC-GEN-12 (Y): A deployment using DTLS between NVEs authenticates the data prior the data is processed by the NVE. As such, the use of DTLS meets SEC-GEN-12.
- o SEC-GEN-13 (N): A deployment using DTLS between NVEs authenticates the data when the tunnel reaches the NVE. As a result the transit device is not able to authenticate the data prior accessing it and the use of DTLS does not meet SEC-GEN-13.

- o SEC-GEN-14 (Y): DTLS provides anti-replay mechanism as such, the use of DTLS meets SEC-GEN-14.
- o SEC-GEN-15 (N): DTLS itself does not have a policy base mechanism. As a result, the classification of the flows needs to be handled by a module outside DTLS. In order to meet SEC-GEN-15 further integration is needed and DTLS in itself cannot be considered as meeting SEC-GEN-15.
- o SEC-GEN-16 (Y): DTLS is able to assign various material to each flows, as such the use of DTLS meets SEC-GEN-16.
- o SEC-GEN-17 (N): DTLS does not handle multicast communications. As such the use of DTLS does not meet SEC-GEN-17.

## 8.2. IPsec

This section compares how NVE communications using IPsec/ESP or IPsec/AH meet the security requirements for a secure Geneve overlay deployment. In this example secures the Geneve IP packet including Outer IP header, the Geneve Outer Header, the Geneve Header including Geneve Options and the inner payload.

The use of IPsec/ESP or IPsec/AH share most of the analysis performed for DTLS. 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.

As a result, the use of IPsec MAY fill the security requirements for a secure Geneve deployment. However IPsec cannot be considered as the Geneve security mechanism enabling all Geneve deployments.

### 8.2.1. Operational Security Requirements

This section shows how IPsec may secure some Geneve deployments. Some Geneve deployments may not be secured by IPsec, but that does not exclude IPsec from being used.

- o SEC-OP-1 (Y): A deployment using IPsec/ESP between NVEs with an non NULL encryption will provide confidentiality to the full Outer IP payload of the Geneve Packet which contains the inner payload. As a result, such deployments meet SEC-OP-1. Note that IPsec/ESP does not provide partial encryption and as such the Geneve Overlay Provider may not benefit from the encryption performed by the Tenant if performed, which may result in some portion of the payload being encrypted twice.

- o SEC-OP-2 (Y): A deployment using IPsec/ESP between NVEs with a non NULL encryption encrypts the Outer IP payload Geneve IP Packet which includes the Geneve Header and associated information. As such SEC-OP-2 is met.
- o SEC-OP-3 (Y): A deployment using IPsec/ESP between NVEs will be able to send dummy packets or pad Geneve Packet. As such OP-SEC-3 is met.
- o SEC-OP-4 (Y): Similarly to SEC-OP-1, A deployment using IPsec/ESP or IPsec/AH between NVEs provides integrity protection to the full Geneve Packet which includes the inner payload. As such SEC-OP-4 is met.
- o SEC-OP-5 (Y): Similarly to SEC-OP-2, A deployment using IPsec/ESP or IPsec/AH between NVE authenticates the full Geneve Packet which includes the Geneve Header. As such SEC-OP-5 is met as well.
- o SEC-OP-6 (Y): A deployment using IPsec/ESP or IPsec/AH between NVE authenticates NVE-to-NVE communications and SEC-OP-6 is met.
- o SEC-OP-7 (Y/N): A deployment using IPsec between NVEs is not compatible with a Geneve architecture that includes transit devices. When IPsec/ESP with a non NULL encryption is used, the transit device will not be able to access it. When IPsec/AH or IPsec/ESP with the NULL encryption is used, the transit device may be able to access the data, but will not be able to authenticate it prior to processing the packet. As SEC-OP-7 is only met for deployment that do not include any transit devices.
- o SEC-OP-8 (Y): A deployment using IPsec between NVEs provides anti-replay protection and so meets SEC-OP-8.
- o SEC-OP-9 (Y/N): IPsec enables the definition of security policies. As such IPsec is likely to handle a per flow security. However the traffic selector required for Geneve flows may not be provided natively by IPsec. As such Sec-OP-9 is only partially met.
- o SEC-OP-10 (Y): IPsec session may be established with specific material, as such it is possible to assign different material for each flow. In addition IPsec supports multicats communications. As such SEC-OP-10 is met.

#### 8.2.2. Geneve Security Requirements

This section shows that IPsec cannot be used as a generic Geneve security mechanism to secure Geneve deployments. A Geneve security mechanism would need to meet all SEC-GEN requirements.

- o SEC-GEN-1 (Y): A deployment using IPsec/ESP between NVEs with an non NULL encryption provide confidentiality to the full Geneve Packet which contains the inner payload. As such IPsec/ESP meets SEC-GEN-1.
- o SEC-GEN-2 (Y): A deployment using IPsec/ESP between NVEs with an non NULL encryption will not be able to partially encrypt the inner payload header. However such requirement is not set a mandatory so IPsec/ESP meets SEC-GEN-2
- o SEC-GEN-3 (N): A deployment using IPsec between NVEs with an non NULL encryption encrypts the Outer IP payload of the Geneve Packet which includes the Geneve Header and all Geneve Options. However IPsec/ESP does not provides any means to selectively encrypt or leave in clear text a subset of Geneve Options. As a result SEC-GEN-3 is not met.
- o SEC-GEN-4 (N): A deployment using IPsec/ESP between NVEs with an non NULL encryption encrypts the Geneve Packet which includes the Geneve Header and all Geneve Options. However, IPsec/ESP does not provides means to selectively encrypt some information of the Geneve Header. As such SEC-GEN-5 is not met.
- o SEC-GEN-5 (N): A deployment using IPsec between NVEs with an non NULL encryption provides end-to-end security between the NVEs and as such does not permit the interaction of one or multiple on-path transit devices. As such IPsec/ESP does not meet SEC-GEN-5.
- o SEC-GEN-6 (Y): A deployment using IPsec/ESP between NVEs with an non NULL encryption provides padding facilities and as such IPsec/ESP meets SEC-GEN-6.
- o SEC-GEN-7 (Y): A deployment using IPsec between NVEs with an non NULL encryption cipher provides the ability to send dummy packets. As such IPsec/ESP meets SEC-GEN-7.
- o SEC-GEN-8 (Y): A deployment using IPsec/ESP or IPsec/AH authenticates the inner payload. As such SEC-GEN-8 is met.
- o SEC-GEN-9 (Y): A deployment using IPsec/AH or IPsec/ESP between NVEs does not provide the ability to partially authenticate the inner payload header. However such requirement is not set a mandatory so IPsec meets SEC-GEN-9
- o SEC-GEN-10 (N): A deployment using IPsec/ESP or IPsec/AH between NVEs authenticates the Geneve Packet which includes the Geneve Header and all Geneve Options. However, IPsec does not provides

means to selectively encrypt some information of the Geneve Header. As such SEC-GEN-10 is not met.

- o SEC-GEN-11 (N): A deployment using IPsec/ESP or IPsec/AH between NVEs authenticates the Geneve Packet which includes the Geneve Header and all Geneve Options. However, IPsec does not provides means to selectively authenticate some information of the Geneve Header. As such SEC-GEN-11 is not met.
- o SEC-GEN-12 (Y): A deployment using IPsec/ESP or IPsec/AH between NVEs authenticates the data prior the data is processed by the NVE. As such SEC-GEN-12 is met.
- o SEC-GEN-13 (N): A deployment using IPsec/ESP or IPsec/AH between NVEs authenticates the data when the tunnel reaches the NVE. As a result the transit device is not able to authenticate the data prior accessing it and SEC-GEN-13 is not met.
- o SEC-GEN-14 (Y): IPsec/ESP and IPsec/AH provides anti-replay mechanism as such SEC-GEN-14 is met.
- o SEC-GEN-15 (N): IPsec is a policy base architecture. As a result, the classification of the flows needs to be handled by IPsec. However, the traffic selector available are probably not those required by Geneve and further integration is needed. As such SEC-GEN-15 is not met.
- o SEC-GEN-16 (Y): IPsec is able to assign various material to each flows, as such SEC-GEN-16 is met.
- o SEC-GEN-17 (Y): IPsec handles mutlicast communications. As such SEC-GEN-17 is met.

## 9. Acknowledgments

We would like to thank Ilango S Ganaga, Magnus Nystroem for their 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>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [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>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [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>>.
- [RFC8247] Nir, Y., Kivinen, T., Wouters, P., and D. Migault, "Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 8247, DOI 10.17487/RFC8247, September 2017, <<https://www.rfc-editor.org/info/rfc8247>>.
- [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-09 (work in progress), February 2019.
- [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.
- [I-D.ietf-tls-dtls13]  
Rescorla, E., Tschofenig, H., and N. Modadugu, "The Datagram Transport Layer Security (DTLS) Protocol Version 1.3", draft-ietf-tls-dtls13-30 (work in progress), November 2018.

## Authors' Addresses

Daniel Migault  
Ericsson  
8275 Trans Canada Route  
Saint Laurent, QC 4S 0B6  
Canada  
  
EMail: [daniel.migault@ericsson.com](mailto:daniel.migault@ericsson.com)



Sami Boutros  
VMware, Inc.

EMail: [boutros@vmware.com](mailto:boutros@vmware.com)<

Dan Wings  
VMware, Inc.

EMail: [dwing@vmware.com](mailto:dwing@vmware.com)

Suresh Krishnan  
Kaloom

EMail: [suresh@kaloom.com](mailto:suresh@kaloom.com)

RTGWG Working Group  
Internet-Draft  
Intended status: Informational  
Expires: December 30, 2018

G. Mirsky  
ZTE Corp.  
June 28, 2018

Identification of Overlay Operations, Administration, and Maintenance  
(OAM)  
draft-mirsky-rtgwg-oam-identify-00

Abstract

This document analyzes how the presence of Operations, Administration, and Maintenance (OAM) control command and/or special data is identified in some overlay networks, and an impact on the choice of identification may have on OAM functionality.

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 December 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions used in this document . . . . .	2
2.1. Terminology . . . . .	2
2.2. Keywords . . . . .	3
3. Overlay Network Encapsulations . . . . .	3
3.1. Encapsulations with Meta-data . . . . .	3
3.2. Fixed-size Encapsulations . . . . .	5
3.3. Source Information Availability . . . . .	5
3.4. On-path OAM . . . . .	6
4. Conclusions . . . . .	6
5. IANA Considerations . . . . .	6
6. Security Considerations . . . . .	7
7. Acknowledgment . . . . .	7
8. References . . . . .	7
8.1. Normative References . . . . .	7
8.2. Informational References . . . . .	7
Author's Address . . . . .	9

## 1. Introduction

Operations, Administration, and Maintenance (OAM) protocols are used to detect, localize defects in the network, and monitor network performance. Some OAM functions, e.g., failure detection, work in the network proactively, while others, e.g., defect localization, usually performed on-demand. These tasks achieved by a combination of active, passive, and hybrid OAM methods, as defined in [RFC7799].

This document analyzes how the presence of Operations, Administration, and Maintenance (OAM) control command and/or special data, i.e., OAM packet, is identified in some overlay networks, and an impact the choice of identification may have on OAM functionality of active and hybrid OAM methods for the respective overlay network encapsulation.

## 2. Conventions used in this document

## 2.1. Terminology

AMM Alternate Marking method

BIER Bit Indexed Explicit Replication

DetNet Deterministic Networks

GUE Generic UDP Encapsulation

HTS Hybrid Two-step

NSH Network Service Header

NVO3 Network Virtualization Overlays

OAM Operations, Administration and Maintenance

SFC Service Function Chaining

TLV Type-Length-Value

VXLAN-GPE Generic Protocol Extension for VXLAN

Underlay Network or Underlay Layer: The network that provides connectivity between the DetNet nodes. MPLS network providing LSP connectivity between DetNet nodes is an example of underlay layer.

## 2.2. Keywords

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. Overlay Network Encapsulations

New overlay network encapsulations analyzed in two groups:

- o encapsulations that support optional meta-data;
- o fixed-size encapsulations.

### 3.1. Encapsulations with Meta-data

Number of the new encapsulation protocols (e.g., Geneve [I-D.ietf-nvo3-geneve], GUE [I-D.ietf-intarea-gue], and SFC NSH [RFC8300]) support use of Type-Length-Value (TLV) encoding to include optional information into the header. The identification of OAM in these protocols is as the following:

Geneve:

O (1 bit): OAM packet. This packet contains a control message instead of a data payload. 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.

GUE:

Undefined.

SFC NSH:

0 bit: Setting this bit indicates an OAM packet.

Common between Geneve and NSH is the use of the dedicated flag to identify the OAM packet and, at the same time, the presence of the field that identifies the protocol of the payload that immediately follows after the encapsulation header. [RFC8393] points that if the value of that field interpreted as none, i.e., no payload follows the header, then OAM may be included in TLVs, thus creating an active OAM packet. The problem with this mechanism to support active OAM methods may be a limitation of the size of data that can be included in a TLV. For example, the maximum size of data in an NSH Meta-data Type 2, as defined in section 2.5.1 [RFC8300], is 512 octets. The maximum length of data in Geneve Option, per section 3.5 [I-D.ietf-nvo3-geneve], is 128 octets. Thus, using one TLV as active OAM packet, would not allow creating test packets of larger size, which is useful when measuring packet loss and latency with synthetic traffic as part of service activation procedure.

[I-D.ietf-sfc-oam-framework] suggests that the 0 bit used to identify OAM packet and the Next Protocol field identifies the OAM function:

While the presence of OAM marker in the overlay header (e.g., 0 bit in the NSH header) indicates it as OAM packet, it is not sufficient to indicate what OAM function the packet is intended for.

At the same time, some of in-situ OAM proposals, e.g., [I-D.ietf-sfc-ioam-nsh], suggest using TLV to communicate hybrid OAM commands and data. The proposed resolution of using the combination of 0 bit and the Next Protocol field:

... the 0 bit MUST NOT be set for regular customer traffic which also carries IOAM data and the 0 bit MUST be set for OAM packets which carry only IOAM data without any regular data payload.

implies that the O bit only identifies the active OAM packet and not set when hybrid OAM methods used.

### 3.2. Fixed-size Encapsulations

Number of the new encapsulation protocols (e.g., VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe], BIER [RFC8296]) use fixed-size header. The identification of OAM in these protocols is as the following:

#### VXLAN-GPE:

OAM Flag Bit (O bit): The O bit is set to indicate that the packet is an OAM packet.

#### BIER:

OAM packet identified by the value of the Next Protocol field. IANA BIER Next Protocol Identifiers registry includes the identifier for OAM (5).

VXLAN-GPE use of a combination of OAM Flag Bit and the Next Protocol field requires clarification of the header interpretation when the OAM Flag Bit is set and the value of the Next Protocol field is one of defined in section 3.2 of [I-D.ietf-nvo3-vxlan-gpe].

BIER encapsulation, defined in [RFC8296], identifies OAM message immediately following the BIER header by the value of the Next Protocol field.

### 3.3. Source Information Availability

Availability of the packet originator's source information is required for active two-way OAM, e.g., echo request/reply. In cases when the underlay network is IPv4/IPv6 the source information will be provided by the encapsulation of the underlay. But when using MPLS underlay network encapsulation of an active OAM packet have to follow certain rules:

- o if available, use Sender ID in the overlay domain (example BFIR ID in BIER [RFC8296]);
- o use IP/UDP encapsulation of an OAM packet in overlay (similar to Section 4.3 [RFC8029]).

### 3.4. On-path OAM

In addition to active methods, OAM toolset may include methods that don't use specially constructed and injected in the network test packets. [RFC7799] defines OAM methods that are neither entirely active nor passive but are combine both as hybrid methods.

One of the examples of the hybrid OAM method, in-situ OAM, mentioned in Section 3.1. Another example, Alternate Marking method (AMM) [RFC8321], enables on-path OAM functions, e.g., delay and loss measurements, using the data traffic. Because AMM impact on the network can be minimized, measured metrics can be correlated to the network conditions experienced by the specific service. Of all listed in Section 3, BIER allocated the field that may be used for AMM, as discussed in [I-D.ietf-bier-pmmm-oam]. Applicability of AMM to other overlay protocols, i.e. SFC NSH discussed in [I-D.mirsky-sfc-pmamm] and Geneve [I-D.fmm-nvo3-pm-alt-mark], been actively discussed.

Hybrid Two-step (HTS), defined in [I-D.mirsky-ippm-hybrid-two-step], is provides on-path collection and transport of the telemetry information. HTS enables accurate and consistent measurements by separating the measurement action from the transport while ensuring that the follow-up packet that carries the telemetry information does follow the data packet that had triggered the measurement.

## 4. Conclusions

OAM control commands and data may be present as part of the overlay encapsulation header or as a payload that follows the overlay network header. The recommendations:

- o OAM in the overlay header, if supported by the overlay network, identified by the dedicated flag. Use of this method as active OAM is possible but functionality is limited.
- o OAM that follows the overlay header identified as payload type, e.g. by the value of the Next Protocol field.

## 5. IANA Considerations

This document does not propose any IANA consideration. This section may be removed.

## 6. Security Considerations

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

## 7. Acknowledgment

TBD

## 8. References

### 8.1. 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>>.
- [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>>.

### 8.2. Informational References

- [I-D.fmm-nvo3-pm-alt-mark]  
Fioccola, G., Mirsky, G., and T. Mizrahi, "Performance Measurement (PM) with Alternate Marking in Network Virtualization Overlays (NVO3)", draft-fmm-nvo3-pm-alt-mark-02 (work in progress), June 2018.
- [I-D.ietf-bier-pmmm-oam]  
Mirsky, G., Zheng, L., Chen, M., and G. Fioccola, "Performance Measurement (PM) with Marking Method in Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-pmmm-oam-04 (work in progress), June 2018.
- [I-D.ietf-intarea-gue]  
Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", draft-ietf-intarea-gue-05 (work in progress), December 2017.
- [I-D.ietf-nvo3-geneve]  
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-06 (work in progress), March 2018.



- [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.
- [I-D.ietf-sfc-ioam-nsh]  
Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "NSH Encapsulation for In-situ OAM Data", draft-ietf-sfc-ioam-nsh-00 (work in progress), May 2018.
- [I-D.ietf-sfc-oam-framework]  
Aldrin, S., Pignataro, C., Kumar, N., Akiya, N., Krishnan, R., and A. Ghanwani, "Service Function Chaining (SFC) Operation, Administration and Maintenance (OAM) Framework", draft-ietf-sfc-oam-framework-04 (work in progress), March 2018.
- [I-D.mirsky-ippm-hybrid-two-step]  
Mirsky, G., Lingqiang, W., and G. Zhui, "Hybrid Two-Step Performance Measurement Method", draft-mirsky-ippm-hybrid-two-step-00 (work in progress), February 2018.
- [I-D.mirsky-sfc-pmamm]  
Mirsky, G., Fioccola, G., and T. Mizrahi, "Performance Measurement (PM) with Alternate Marking Method in Service Function Chaining (SFC) Domain", draft-mirsky-sfc-pmamm-03 (work in progress), June 2018.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed.,  
"Network Service Header (NSH)", RFC 8300,  
DOI 10.17487/RFC8300, January 2018,  
<<https://www.rfc-editor.org/info/rfc8300>>.
- [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>>.
- [RFC8393] Farrel, A. and J. Drake, "Operating the Network Service  
Header (NSH) with Next Protocol "None"", RFC 8393,  
DOI 10.17487/RFC8393, May 2018,  
<<https://www.rfc-editor.org/info/rfc8393>>.

## Author's Address

Greg Mirsky  
ZTE Corp.

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)

RTGWG Working Group  
Internet-Draft  
Intended status: Informational  
Expires: August 29, 2020

G. Mirsky  
ZTE Corp.  
February 26, 2020

Identification of Overlay Operations, Administration, and Maintenance  
(OAM)  
draft-mirsky-rtgwg-oam-identify-04

Abstract

This document analyzes how the presence of Operations, Administration, and Maintenance (OAM) control command and/or special data is identified in some overlay networks and an impact on the choice of identification may have on OAM functionality.

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 August 29, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions used in this document . . . . .	2
2.1. Terminology . . . . .	2
2.2. Keywords . . . . .	3
3. A Control Channel in an Overlay Network . . . . .	3
4. Overlay Network Encapsulations . . . . .	4
4.1. Encapsulations with Meta-data . . . . .	4
4.1.1. Available Solutions . . . . .	6
4.2. Fixed-size Encapsulations . . . . .	6
4.3. Source Information Availability . . . . .	7
4.4. On-path OAM . . . . .	7
5. Conclusions . . . . .	8
6. IANA Considerations . . . . .	8
7. Security Considerations . . . . .	8
8. Acknowledgment . . . . .	8
9. References . . . . .	8
9.1. Normative References . . . . .	9
9.2. Informational References . . . . .	9
Author's Address . . . . .	11

## 1. Introduction

Operations, Administration, and Maintenance (OAM) protocols are used to detect, localize defects in the network, and monitor network performance. Some OAM functions, e.g., failure detection, work in the network proactively, while others, e.g., defect localization, usually performed on-demand. These tasks achieved by a combination of active, passive, and hybrid OAM methods, as defined in [RFC7799].

This document analyzes how the presence of Operations, Administration, and Maintenance (OAM) control command and/or special data, i.e., OAM packet, is identified in some overlay networks, and an impact the choice of identification may have on OAM functionality of active and hybrid OAM methods for the respective overlay network encapsulation.

## 2. Conventions used in this document

## 2.1. Terminology

AMM Alternate Marking method

BIER Bit Indexed Explicit Replication

DetNet Deterministic Networks

GUE Generic UDP Encapsulation

HTS Hybrid Two-step

NSH Network Service Header

NVO3 Network Virtualization Overlays

OAM Operations, Administration and Maintenance

SFC Service Function Chaining

TLV Type-Length-Value

VXLAN-GPE Generic Protocol Extension for VXLAN

ACH Associated Channed Header

Underlay Network or Underlay Layer: The network that provides connectivity between the DetNet nodes. MPLS network that provides LSP connectivity between DetNet nodes is an example of an underlay layer.

## 2.2. Keywords

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. A Control Channel in an Overlay Network

There's a need for a general control channel between the endpoints of an overlay network for OAM protocols that can be used for fault detection, diagnostics, maintenance, and other functions. Such a control tunnel is dedicated to carrying only control and management data between tunnel endpoints. In other words, the control channel of an overlay network SHOULD NOT carry the client's data. And the endpoint node SHOULD NOT forward a packet received over the control channel. The identification of the control channel might be using different methods. For example, Virtual Network Identifier might be used to identify the control channel in VXLAN and Geneve.

#### 4. Overlay Network Encapsulations

New overlay network encapsulations analyzed in two groups:

- o encapsulations that support optional meta-data;
- o fixed-size encapsulations.

##### 4.1. Encapsulations with Meta-data

Number of the new encapsulation protocols (e.g., Geneve [I-D.ietf-nvo3-geneve], GUE [I-D.ietf-intarea-gue], and SFC NSH [RFC8300]) support use of Type-Length-Value (TLV) encoding to include optional information into the header. The identification of OAM in these protocols is as the following:

Geneve:

O (1 bit): after the WGLC discussion, the interpretation of the O field has changed. The O field now identifies a control packet. This packet contains a control message. Control messages are sent between tunnel endpoints. Tunnel Endpoints MUST NOT forward the payload and transit devices MUST NOT attempt to interpret it. Since these are infrequent control messages, it is RECOMMENDED that tunnel 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.

[I-D.mmabb-nvo3-geneve-oam] defines the Geneve encapsulation for active OAM. Initially, four options have been presented:

- + with IP/UDP header demultiplexing active OAM protocols, e.g., Fault Management and Performance Monitoring, can be done using the destination UDP port number.
- + demultiplex active OAM protocols by the value of the Protocol Type field in the Geneve header.
- + with using MPLS Generic Associated Channel Label [RFC5586] and Associated Channel Header (ACH) [RFC4385]. Active OAM protocols are demultiplexed using the value of the Channel Type field.

- + using the new EtherType to identify Geneve OAM and the ACH. Active OAM protocols will be demultiplexed based on the Channel Type field's value.

#### GUE:

C-bit provides the separate namespace to carry formatted data that are implicitly addressed to the decapsulator to monitor or control the state or behavior of a tunnel. The payload is interpreted as a control message with the type specified in the proto/ctype field. The format and contents of the control message are indicated by the type and can be variable length.

#### SFC NSH:

0 bit: Setting this bit indicates an OAM packet.

Common between Geneve and NSH is the use of the dedicated flag to identify the OAM packet and, at the same time, the presence of the field that identifies the protocol of the payload that immediately follows after the encapsulation header. [RFC8393] points out that if the value of that field interpreted as none, i.e., no payload follows the header, then OAM may be included in TLVs, thus creating an active OAM packet. The problem with this mechanism to support active OAM methods may be a limitation of the size of data that can be included in a TLV. For example, the maximum size of data in an NSH Meta-data Type 2, as defined in section 2.5.1 [RFC8300], is 512 octets. The maximum length of data in Geneve Option, per section 3.5 [I-D.ietf-nvo3-geneve], is 128 octets. Thus, using one TLV as active OAM packet, would not allow creating test packets of larger size, which is useful when measuring packet loss and latency with synthetic traffic as part of the service activation procedure.

[I-D.ietf-sfc-oam-framework] suggests that the 0 bit used to identify OAM packet and the Next Protocol field identifies the OAM function:

While the presence of OAM marker in the overlay header (e.g., 0 bit in the NSH header) indicates it as OAM packet, it is not sufficient to signal for which OAM function the packet is intended.

At the same time, some of in-situ OAM proposals, e.g., [I-D.ietf-sfc-ioam-nsh], suggest using TLV to communicate hybrid OAM commands and data. The proposed resolution of using the combination of 0 bit and the Next Protocol field:

... the O bit MUST NOT be set for regular customer traffic which also carries IOAM data and the O bit MUST be set for OAM packets which carry only IOAM data without any regular data payload.

implies that the O bit only identifies the active OAM packet and not set when hybrid OAM methods used.

#### 4.1.1. Available Solutions

One of the possible solutions for encapsulations with meta-data has been specified in [I-D.ietf-sfc-multi-layer-oam]:

To identify the active OAM message the value on the Next Protocol field MUST be set to Active SFC OAM. The rules of interpreting the values of O bit and the Next Protocol field are as follows:

- o O bit set and the Next Protocol value is not one of identifying active or hybrid OAM protocol (per [RFC7799] definitions), e.g., defined in this specification Active SFC OAM - a Fixed-Length Context Header or Variable-Length Context Header(s) contain OAM command or data and the type of payload determined by the Next Protocol field;
- o O bit set and the Next Protocol value is one of identifying active or hybrid OAM protocol - the payload that immediately follows SFC NSH contains OAM command or data;
- o O bit is clear - no OAM in a Fixed-Length Context Header or Variable-Length Context Header(s) and the payload determined by the value of the Next Protocol field;
- o O bit is clear, and the Next Protocol value is one of identifying active or hybrid OAM protocol MUST be identified and reported as the erroneous combination. An implementation MAY have control to enable processing of the OAM payload.

From the above-listed rules follows the recommendation to avoid the combination of OAM in a Fixed-Length Context Header or Variable-Length Context Header(s) and in the payload immediately following the SFC NSH because there is no unambiguous way to identify such combination using the O bit and the Next Protocol field.

#### 4.2. Fixed-size Encapsulations

Number of the new encapsulation protocols (e.g., VXLAN-GPE [I-D.ietf-nvo3-vxlan-gpe], BIER [RFC8296]) use fixed-size header. The identification of OAM in these protocols is as the following:



**VXLAN-GPE:**

OAM Flag Bit (O bit): The O bit is set to indicate that the packet is an OAM packet.

**BIER:**

OAM packet identified by the value of the Next Protocol field. IANA BIER Next Protocol Identifiers registry includes the identifier for OAM (5).

The use of a combination of OAM Flag Bit and the Next Protocol field in VXLAN-GPE requires clarification of the header interpretation when the OAM Flag Bit is set, and the value of the Next Protocol field is one of defined in section 3.2 of [I-D.ietf-nvo3-vxlan-gpe].

BIER encapsulation, defined in [RFC8296], identifies OAM message immediately following the BIER header by the value of the Next Protocol field.

**4.3. Source Information Availability**

Availability of the packet originator's source information is required for active two-way OAM, e.g., echo request/reply. In cases when the underlay network is IPv4/IPv6 the source information will be derived from the underlay. But when using MPLS underlay network encapsulation of an active OAM packet have to follow specific rules:

- o if available, use Sender ID in the overlay domain (example BFIR ID in BIER [RFC8296];
- o use IP/UDP encapsulation of an OAM packet in the overlay (similar to Section 4.3 [RFC8029]).

**4.4. On-path OAM**

In addition to active methods, OAM toolset may include methods that don't use specially constructed and injected in the network test packets. [RFC7799] defines OAM methods that are neither entirely active nor passive but are a combination of both as hybrid methods.

One of the examples of the hybrid OAM methods, in-situ OAM, mentioned in Section 4.1. Another example, Alternate Marking method (AMM) [RFC8321], enables on-path OAM functions, e.g., delay and loss measurements, using the data traffic. Because AMM impact on the network can be minimized, measured metrics can be correlated to the network conditions experienced by the specific service. Of all listed in Section 4, BIER allocated the field that may be used for

AMM, as discussed in [I-D.ietf-bier-pmmm-oam]. Applicability of AMM to other overlay protocols, i.e., SFC NSH discussed in [I-D.mirsky-sfc-pmamm], Geneve [I-D.fmm-nvo3-pm-alt-mark], and in IPv6 networks [I-D.fioccola-v6ops-ipv6-alt-mark], been actively discussed.

Hybrid Two-step (HTS), defined in [I-D.mirsky-ippm-hybrid-two-step], provides on-path collection and transport of the telemetry information. HTS enables accurate and consistent measurements by separating the measurement action from the transporting data while ensuring that the follow-up packet that carries the telemetry information does follow the data packet that had triggered the measurement.

## 5. Conclusions

OAM control commands and data may be present as part of the overlay encapsulation header or as a payload that follows the overlay network header. The recommendations:

- o OAM in the overlay header, if supported by the overlay network, identified by the dedicated flag. Use of this method as active OAM is possible, but functionality is limited.
- o OAM that follows the overlay header identified as payload type, e.g., by the value of the Next Protocol field.

## 6. IANA Considerations

This document does not propose any IANA consideration. This section may be removed.

## 7. Security Considerations

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

## 8. Acknowledgment

TBD

## 9. References

### 9.1. 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>>.
- [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. Informational References

- [I-D.fioccola-v6ops-ipv6-alt-mark]  
Fioccola, G., Velde, G., Cociglio, M., and P. Muley, "IPv6 Performance Measurement with Alternate Marking Method", draft-fioccola-v6ops-ipv6-alt-mark-01 (work in progress), June 2018.
- [I-D.fmm-nvo3-pm-alt-mark]  
Fioccola, G., Mirsky, G., and T. Mizrahi, "Performance Measurement (PM) with Alternate Marking in Network Virtualization Overlays (NVO3)", draft-fmm-nvo3-pm-alt-mark-03 (work in progress), October 2018.
- [I-D.ietf-bier-pmmm-oam]  
Mirsky, G., Zheng, L., Chen, M., and G. Fioccola, "Performance Measurement (PM) with Marking Method in Bit Index Explicit Replication (BIER) Layer", draft-ietf-bier-pmmm-oam-07 (work in progress), January 2020.
- [I-D.ietf-intarea-gue]  
Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", draft-ietf-intarea-gue-09 (work in progress), October 2019.
- [I-D.ietf-nvo3-geneve]  
Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-14 (work in progress), September 2019.
- [I-D.ietf-nvo3-vxlan-gpe]  
Maino, F., Kreeger, L., and U. Elzur, "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-09 (work in progress), December 2019.

- [I-D.ietf-sfc-ioam-nsh]  
Brockners, F. and S. Bhandari, "Network Service Header (NSH) Encapsulation for In-situ OAM (IOAM) Data", draft-ietf-sfc-ioam-nsh-02 (work in progress), September 2019.
- [I-D.ietf-sfc-multi-layer-oam]  
Mirsky, G., Meng, W., Khasnabish, B., and C. Wang, "Active OAM for Service Function Chains in Networks", draft-ietf-sfc-multi-layer-oam-04 (work in progress), November 2019.
- [I-D.ietf-sfc-oam-framework]  
Aldrin, S., Pignataro, C., Kumar, N., Krishnan, R., and A. Ghanwani, "Service Function Chaining (SFC) Operations, Administration and Maintenance (OAM) Framework", draft-ietf-sfc-oam-framework-11 (work in progress), September 2019.
- [I-D.mirsky-ippm-hybrid-two-step]  
Mirsky, G., Lingqiang, W., and G. Zhui, "Hybrid Two-Step Performance Measurement Method", draft-mirsky-ippm-hybrid-two-step-04 (work in progress), October 2019.
- [I-D.mirsky-sfc-pmamm]  
Mirsky, G., Fioccola, G., and T. Mizrahi, "Performance Measurement (PM) with Alternate Marking Method in Service Function Chaining (SFC) Domain", draft-mirsky-sfc-pmamm-09 (work in progress), December 2019.
- [I-D.mmbb-nvo3-geneve-oam]  
Mirsky, G., Xiao, M., Boutros, S., and D. Black, "OAM for use in GENEVE", draft-mmbb-nvo3-geneve-oam-01 (work in progress), January 2020.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<https://www.rfc-editor.org/info/rfc5586>>.
- [RFC7799] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, <<https://www.rfc-editor.org/info/rfc7799>>.

- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.
- [RFC8300] Quinn, P., Ed., Elzur, U., Ed., and C. Pignataro, Ed., "Network Service Header (NSH)", RFC 8300, DOI 10.17487/RFC8300, January 2018, <<https://www.rfc-editor.org/info/rfc8300>>.
- [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>>.
- [RFC8393] Farrel, A. and J. Drake, "Operating the Network Service Header (NSH) with Next Protocol "None"", RFC 8393, DOI 10.17487/RFC8393, May 2018, <<https://www.rfc-editor.org/info/rfc8393>>.

## Author's Address

Greg Mirsky  
ZTE Corp.

Email: [gregimirsky@gmail.com](mailto:gregimirsky@gmail.com)