Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 16, 2020

F. Maino, Ed.
Cisco Systems
L. Kreeger, Ed.
Arrcus
U. Elzur, Ed.
Intel

October 14, 2019

Generic Protocol Extension for VXLAN
draft-ietf-nvo3-vxlan-gpe-08

Abstract

This draft describes extending Virtual eXtensible Local Area Network (VXLAN), via changes to the VXLAN header, with three new capabilities: support for multi-protocol encapsulation, operations, administration and management (OAM) signaling and explicit versioning.

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].

Status of This Memo

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

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

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

This Internet-Draft will expire on April 16, 2020.

Copyright Notice

Table of Contents

1. Introduction

Virtual eXtensible Local Area Network VXLAN [RFC7348] defines an encapsulation format that encapsulates Ethernet frames in an outer UDP/IP transport. As data centers evolve, the need to carry other protocols encapsulated in an IP packet is required, as well as the need to provide increased visibility and diagnostic capabilities

within the overlay. The VXLAN header does not specify the protocol being encapsulated and therefore is currently limited to encapsulating only Ethernet frame payload, nor does it provide the ability to define OAM protocols. In addition, [RFC6335] requires that new transports not use transport layer port numbers to identify tunnel payload, rather it encourages encapsulations to use their own identifiers for this purpose. VXLAN GPE is intended to extend the existing VXLAN protocol to provide protocol typing, OAM, and versioning capabilities.

The Version and OAM bits are introduced in Section 3, and the choice of location for these fields is driven by minimizing the impact on existing deployed hardware.

In order to facilitate deployments of VXLAN GPE with hardware currently deployed to support VXLAN, changes from legacy VXLAN have been kept to a minimum. Section 5 provides a detailed discussion about how VXLAN GPE addresses the requirement for backward compatibility with VXLAN.

2. VXLAN Without Protocol Extension

VXLAN provides a method of creating multi-tenant overlay networks by encapsulating packets in IP/UDP along with a header containing a network identifier which is used to isolate tenant traffic in each overlay network from each other. This allows the overlay networks to run over an existing IP network.

Through this encapsulation, VXLAN creates stateless tunnels between VXLAN Tunnel End Points (VTEPs) which are responsible for adding/ removing the IP/UDP/VXLAN headers and providing tenant traffic isolation based on the VXLAN Network Identifier (VNI). Tenant systems are unaware that their networking service is being provided by an overlay.

When encapsulating packets, a VTEP must know the IP address of the proper remote VTEP at the far end of the tunnel that can deliver the inner packet to the Tenant System corresponding to the inner destination address. In the case of tenant multicast or broadcast, the outer IP address may be an IP multicast group address, or the VTEP may replicate the packet and send it to all known VTEPs. If multicast is used in the underlay network to send encapsulated packets to remote VTEPs, Any Source Multicast is used and each VTEP serving a particular VNI must perform a (*, G) join to the same group IP address.

Inner to outer address mapping can be determined in two ways. One is source based learning in the data plane, and the other is distribution via a control plane.

Source based learning requires a receiving VTEP to create an inner to outer address mapping by gleaning the information from the received packets by correlating the inner source address to the outer source IP address. When a mapping does not exist, a VTEP forwards the packets to all remote VTEPs participating in the VNI by using IP multicast in the IP underlay network. Each VTEP must be configured with the IP multicast address to use for each VNI. How this occurs is out of scope.

The control plane used to distribute inner to outer mappings is also out of scope. It could use a centralized authority or be distributed, or use a hybrid.

The VXLAN Network Identifier (VNI) provides scoping for the addresses in the header of the encapsulated PDU. If the encapsulated packet is an Ethernet frame, this means the Ethernet MAC addresses are only unique within a given VNI and may overlap with MAC addresses within a different VNI. If the encapsulated packet is an IP packet, this means the IP addresses are only unique within that VNI.

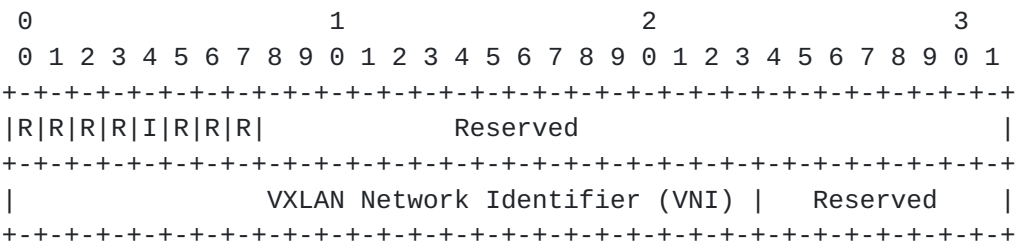


Figure 1: VXLAN Header

3. Generic Protocol Extension for VXLAN (VXLAN GPE)

3.1. VXLAN GPE Header

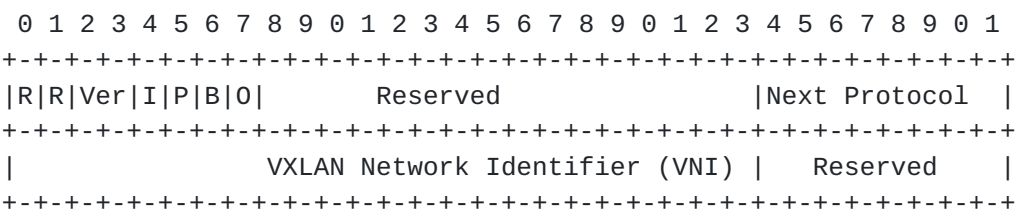


Figure 2: VXLAN GPE Header

Flags (8 bits): The first 8 bits of the header are the flag field. The bits designated "R" above are reserved flags. These MUST be set to zero on transmission and ignored on receipt.

Version (Ver): Indicates VXLAN GPE protocol version. The initial version is 0. If a receiver does not support the version indicated it MUST drop the packet.

Instance Bit (I bit): The I bit MUST be set to indicate a valid VNI.

Next Protocol Bit (P bit): The P bit is set to indicate that the Next Protocol field is present.

BUM Traffic Bit (B bit): The B bit is set to indicate that this is ingress-replicated BUM Traffic (ie, Broadcast, Unknown unicast, or Multicast).

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

Next Protocol: This 8 bit field indicates the protocol header immediately following the VXLAN GPE header.

VNI: This 24 bit field identifies the VXLAN overlay network the inner packet belongs to. Inner packets belonging to different VNIs cannot communicate with each other (unless explicitly allowed by policy).

Reserved: Reserved fields MUST be set to zero on transmission and ignored on receipt.

3.2. Multi Protocol Support

This draft defines the following two changes to the VXLAN header in order to support multi-protocol encapsulation:

P Bit: Flag bit 5 is defined as the Next Protocol bit. The P bit MUST be set to 1 to indicate the presence of the 8 bit next protocol field.

When UDP dest port=4790, P = 0 the "Next Protocol" field must be set to zero and the payload MUST be ETHERNET(L2) as defined by [RFC7348].

Flag bit 5 was chosen as the P bit because this flag bit is currently reserved in VXLAN.

Next Protocol Field: The lower 8 bits of the first word are used to carry a next protocol. This next protocol field contains the protocol of the encapsulated payload packet. A new protocol registry will be requested from IANA, see section 10.2.

This draft defines the following Next Protocol values:

0x1 : IPv4

0x2 : IPv6

0x3 : Ethernet

0x4 : Network Service Header [RFC8300]

0x5 : Multiprotocol Label Switching [RFC3031]. See [I-D.ietf-idr-tunnel-encaps] for more details.

0x6: Unassigned.

0x7: virtual Broadband Network Gateway (vBNG) [I-D.huang-nvo3-vxlan-gpe-extension-for-vbng].

0x8 to 0x7F: Unassigned.

0x80: Group-Based Policy (GBP) [I-D.lemon-vxlan-lisp-gpe-gbp]

0x81: In-situ OAM Data (iOAM)[I-D.brockners-ippm-ioam-vxlan-gpe]

0x82 to 0xFF: Unassigned.

Next protocol values from Ox80 to 0xFF are assigned to protocols encoded as generic "shim" headers. These protocols, when present, MUST be encapsulated before protocols identified by next protocol values from 0x0 to 0x7F.

Implementations that are not aware of a given shim header MUST ignore the header and proceed to parse the next protocol. Shim protocols MUST have the first 32 bits defined as:

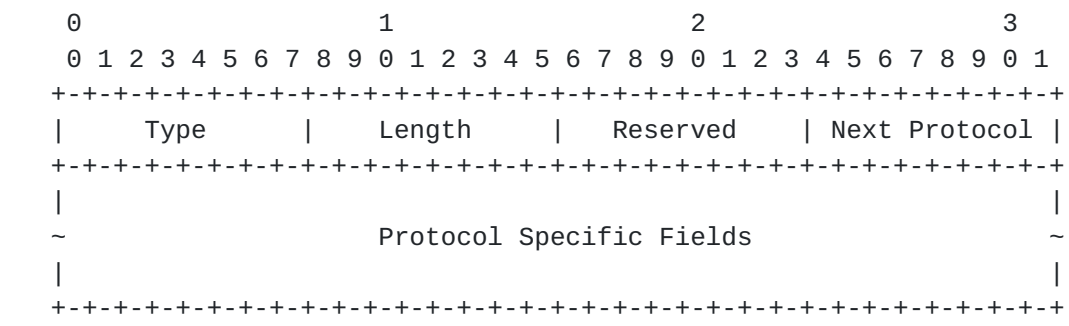


Figure 3: Shim Header

Where:

Type: This field MAY be used to identify different messages of this protocol.

Length: The length, in in 4-octet units, of this protocol message not including the first 4 octects.

Reserved: The use of this field is reserved to the protocol defined in this message.

Next Protocol Field: This next protocol field contains the protocol of the encapsulated payload. The protocol registry will be requested from IANA as per section 10.2.

3.3. Replicated BUM Traffic

Flag bit 6 is defined as the B bit. When the B bit is set to 1, the packet is marked as an an ingress-replicated BUM Traffic (i.e. Broadcast, Unknown unicast, or Multicast) to help egress VTEP to differentiate between known and unknown unicast. The details of using the B bit are out of scope for this document, but please see [RFC8365] for an example in the EVPN context. As with the P-bit, bit 6 is currently a reserved flag in VXLAN.

3.4. OAM Support

Flag bit 7 is defined as the O bit. When the O bit is set to 1, the packet is an OAM packet and OAM processing MUST occur. Other header fields including Next Protocol MUST adhere to the definitions in Section 3. The OAM protocol details are out of scope for this document. As with the P-bit, bit 7 is currently a reserved flag in VXLAN.

3.5. Version Bits

VXLAN GPE bits 2 and 3 are defined as version bits. These bits are reserved in VXLAN. The version field is used to ensure backward compatibility going forward with future VXLAN GPE updates.

The initial version for VXLAN GPE is 0.

4. Outer Encapsulations

In addition to the VXLAN GPE header, the packet is further encapsulated in UDP and IP. Data centers based on Ethernet, will then send this IP packet over Ethernet.

Outer UDP Header:

Destination UDP Port: IANA has assigned the value 4790 for the VXLAN GPE UDP port. This well-known destination port is used when sending VXLAN GPE encapsulated packets.

Source UDP Port: The source UDP port is used as entropy for devices forwarding encapsulated packets across the underlay (ECMP for IP routers, or load splitting for link aggregation by bridges). Tenant traffic flows should all use the same source UDP port to lower the chances of packet reordering by the underlay for a given flow. It is recommended for VTEPs to generate this port number using a hash of the inner packet headers. Implementations MAY use the entire 16 bit source UDP port for entropy.

UDP Checksum: Source VTEPs MAY either calculate a valid checksum, or if this is not possible, set the checksum to zero. When calculating a checksum, it MUST be calculated across the entire packet (outer IP header, UDP header, VXLAN GPE header and payload packet). All receiving VTEPs must accept a checksum value of zero. If the receiving VTEP is capable of validating the checksum, it MAY validate a non-zero checksum and MUST discard the packet if the checksum is determined to be invalid.

Outer IP Header:

This is the header used by the underlay network to deliver packets between VTEPs. The destination IP address can be a unicast or a multicast IP address. The source IP address must be the source VTEP IP address which can be used to return tenant packets to the tenant system source address within the inner packet header.

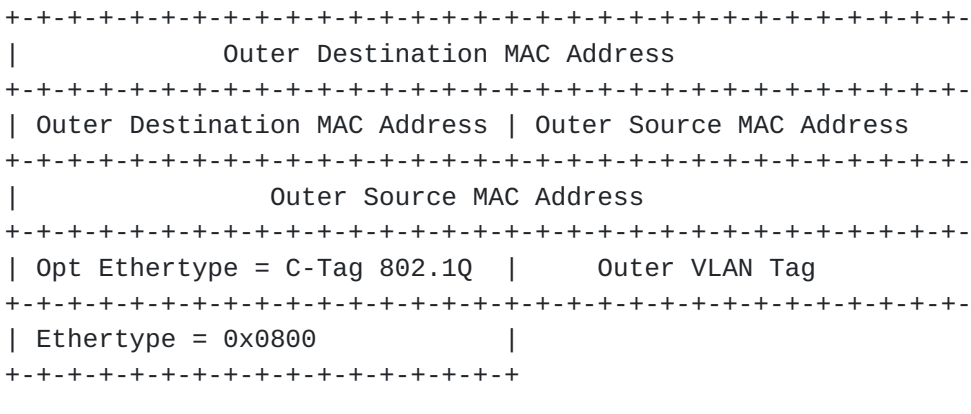
When the outer IP header is IPv4, VTEPs MUST set the DF bit.

Outer Ethernet Header:

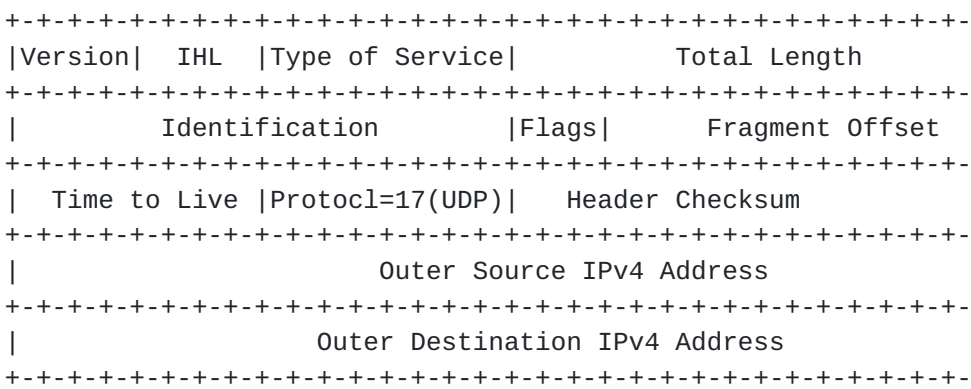
Most data centers networks are built on Ethernet. Assuming the outer IP packet is being sent across Ethernet, there will be an Ethernet header used to deliver the IP packet to the next hop, which could be the destination VTEP or be a router used to forward the IP packet towards the destination VTEP. If VLANs are in use within the data center, then this Ethernet header would also contain a VLAN tag.

The following figures show the entire stack of protocol headers that would be seen on an Ethernet link carrying encapsulated packets from a VTEP across the underlay network for both IPv4 and IPv6 based underlay networks.

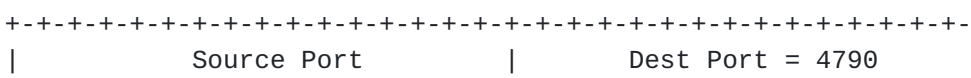
Outer Ethernet Header:



Outer IPv4 Header:



Outer UDP Header:



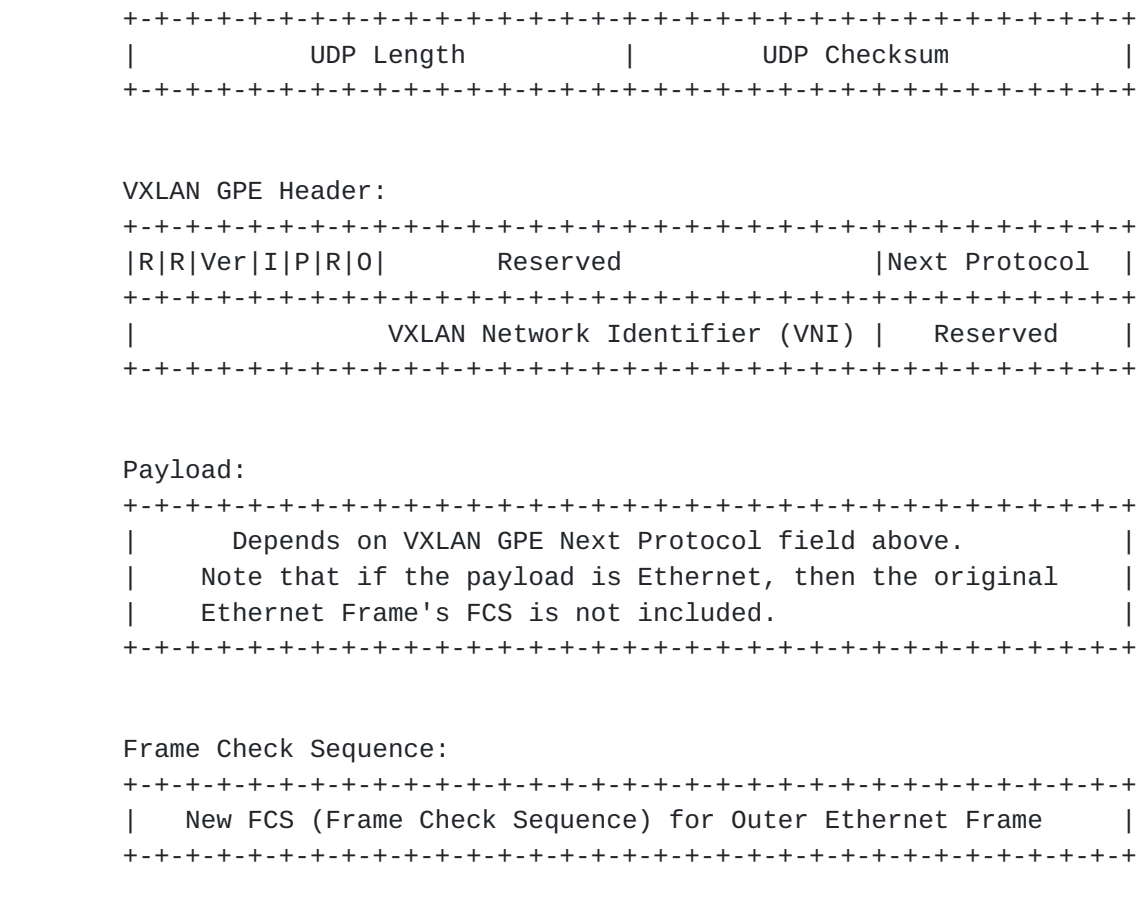
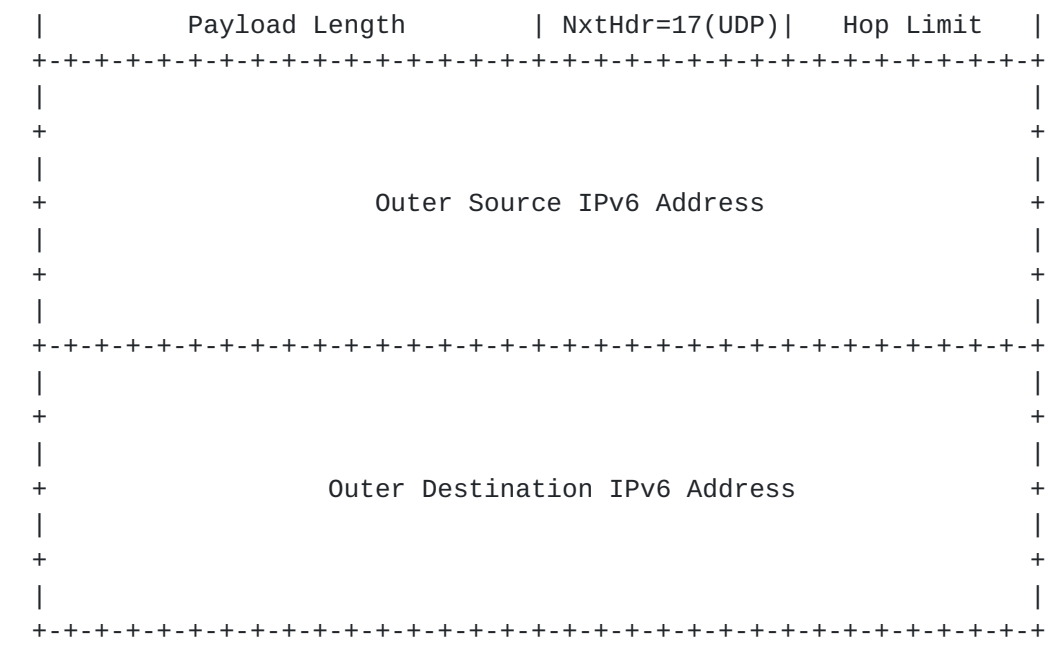
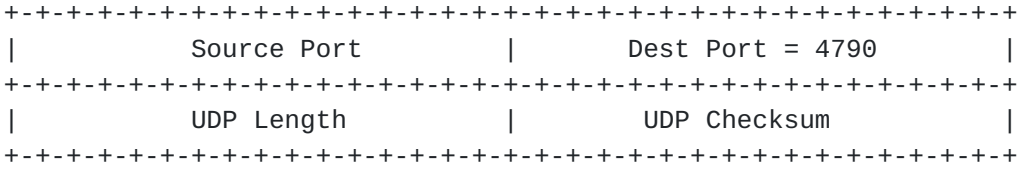


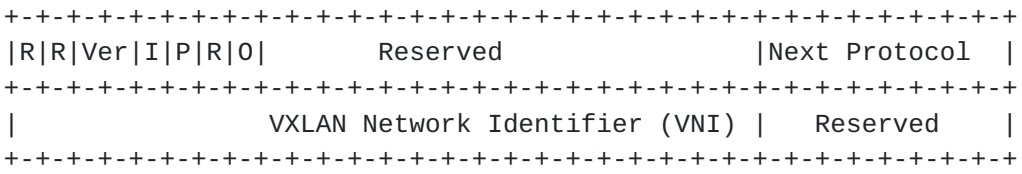
Figure 4: Outer Headers for VXLAN GPE over IPv4



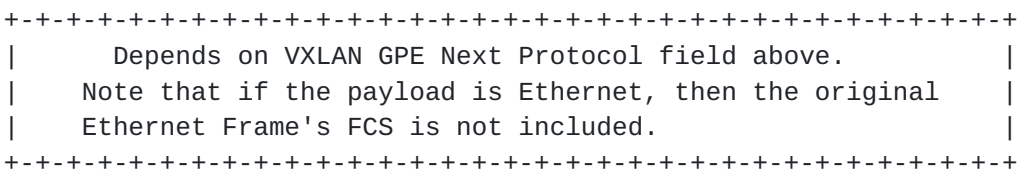
Outer UDP Header:



VXLAN GPE Header:



Payload:



Frame Check Sequence:

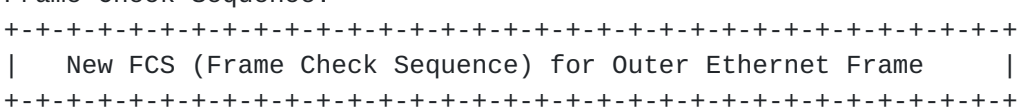


Figure 5: Outer Headers for VXLAN GPE over IPv6

4.1. Inner VLAN Tag Handling

If the inner packet (as indicated by the VXLAN GPE Next Protocol field) is an Ethernet frame, it is recommended that it does not contain a VLAN tag. In the most common scenarios, the tenant VLAN tag is translated into a VXLAN Network Identifier. In these scenarios, VTEPs should never send an inner Ethernet frame with a VLAN tag, and a VTEP performing decapsulation should discard any inner frames received with a VLAN tag. However, if the VTEPs are specifically configured to support it for a specific VXLAN Network Identifier, a VTEP may support transparent transport of the inner VLAN tag between all tenant systems on that VNI. The VTEP never looks at the value of the inner VLAN tag, but simply passes it across the underlay.

4.2. Fragmentation Considerations

VTEPs MUST never fragment an encapsulated VXLAN GPE packet, and when the outer IP header is IPv4, VTEPs MUST set the DF bit in the outer IPv4 header. It is recommended that the underlay network be configured to carry an MTU at least large enough to accommodate the added encapsulation headers. It is recommended that VTEPs perform Path MTU discovery [RFC1191] [RFC1981] to determine if the underlay network can carry the encapsulated payload packet.

5. Backward Compatibility

5.1. VXLAN VTEP to VXLAN GPE VTEP

A VXLAN VTEP conforms to VXLAN frame format and uses UDP destination port 4789 when sending traffic to VXLAN GPE VTEP. As per VXLAN, reserved bits 5 and 7, VXLAN GPE P and O-bits respectively must be set to zero. The remaining reserved bits must be zero, including the VXLAN GPE version field, bits 2 and 3. The encapsulated payload MUST be Ethernet.

5.2. VXLAN GPE VTEP to VXLAN VTEP

A VXLAN GPE VTEP MUST NOT encapsulate non-Ethernet frames to a VXLAN VTEP. When encapsulating Ethernet frames to a VXLAN VTEP, the VXLAN GPE VTEP MUST conform to VXLAN frame format and hence will set the P bit to 0, the Next Protocol to 0 and use UDP destination port 4789. A VXLAN GPE VTEP MUST also set O = 0 and Ver = 0 when encapsulating Ethernet frames to VXLAN VTEP. The receiving VXLAN VTEP will treat this packet as a VXLAN packet.

A method for determining the capabilities of a VXLAN VTEP (GPE or non-GPE) is out of the scope of this draft.

5.3. VXLAN GPE UDP Ports

VXLAN GPE uses a IANA assigned UDP destination port, 4790, when sending traffic to VXLAN GPE VTEPs.

5.4. VXLAN GPE and Encapsulated IP Header Fields

When encapsulating and decapsulating IPv4 and IPv6 packets, certain fields, such as IPv4 Time to Live (TTL) from the inner IP header need to be considered. VXLAN GPE IP encapsulation and decapsulation utilizes the techniques described in [RFC6830], section 5.3.

6. VXLAN GPE Examples

This section provides three examples of protocols encapsulated using the Generic Protocol Extension for VXLAN described in this document.

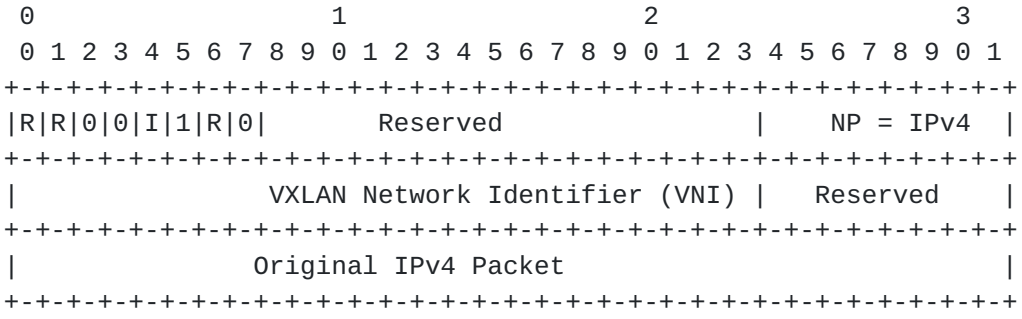


Figure 6: IPv4 and VXLAN GPE

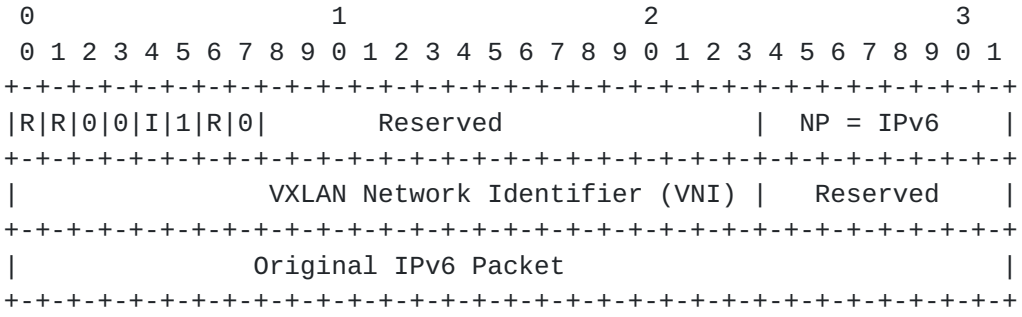


Figure 7: IPv6 and VXLAN GPE

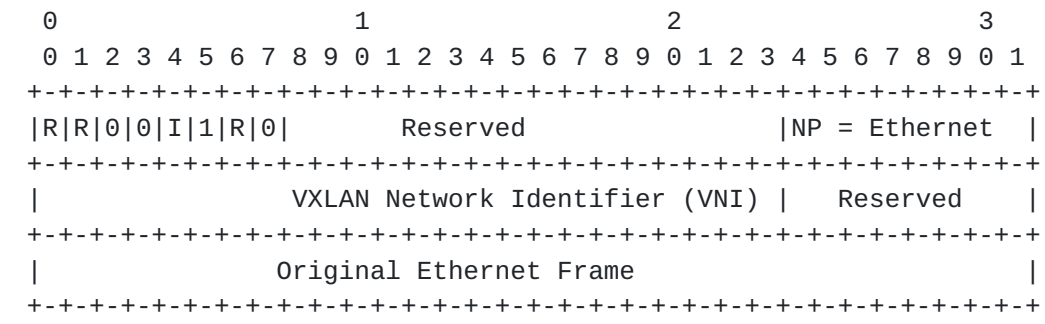


Figure 8: Ethernet and VXLAN GPE

7. Security Considerations

VXLAN's security is focused on issues around L2 encapsulation into L3. With VXLAN GPE, issues such as spoofing, flooding, and traffic redirection are dependent on the particular protocol payload encapsulated.

8. Contributors

Paul Quinn
Cisco Systems
paulq@cisco.com

Rajeev Manur
Broadcom
rmanur@broadcom.com

Michael Smith
Cisco Systems
michsmit@cisco.com

Darrel Lewis
Cisco Systems
darlewis@cisco.com

Puneet Agarwal
Innovium, Inc
puneet@acm.org

Lucy Yong
Huawei USA
lucy.yong@huawei.com

Xiaohu Xu
Huawei Technologies
xuxiaohu@huawei.com

Pankaj Garg
Microsoft
pankajg@microsoft.com

David Melman
Marvell
davidme@marvell.com

Jennifer Lemon
Broadcom Limited
jennifer.lemon@broadcom.com

9. Acknowledgments

A special thank you goes to Dino Farinacci for his guidance and detailed review.

10. IANA Considerations

10.1. UDP Port

UDP 4790 port has been assigned by IANA for VXLAN GPE.

10.2. VXLAN GPE Next Protocol

IANA is requested to set up a registry of "Next Protocol". These are 8-bit values. Next Protocol values in the table below are defined in this draft. New values are assigned via Standards Action [RFC5226].

Next Protocol	Description	Reference
0x0	Reserved	This Document
0x1	IPv4	This Document
0x2	IPv6	This Document
0x3	Ethernet	This Document
0x4	NSH	This Document
0x5	MPLS	This Document
0x6	Unassigned	
0x7	vBNG	This Document
0x8..0x7F	Unassigned	
0x80	GBP	This Document
0x81	iOAM	This Document
0x82..0xFF	Unassigned	

10.3. VXLAN GPE Flag and Reserved Bits

There are ten flag bits at the beginning of the VXLAN GPE header, followed by 16 reserved bits and an 8-bit reserved field at the end of the header. New bits are assigned via Standards Action [RFC5226].

Bits 0-1 - Reserve6

Bits 2-3 - Version

Bit 4 - Instance ID (I bit)

Bit 5 - Next Protocol (P bit)

Bit 6 - Reserved

Bit 7 - OAM (O bit)

Bit 8-23 - Reserved

Bits 24-31 in the 2nd Word -- Reserved

Reserved bits/fields MUST be set to 0 by the sender and ignored by the receiver.

11. References

11.1. Normative References

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[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>.

[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>.

[RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <https://www.rfc-editor.org/info/rfc8365>.

11.2. Informative References

[I-D.brockners-ippm-ioam-vxlan-gpe]
Brockners, F., Bhandari, S., Govindan, V., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Kfir, A., Gafni, B., Lapukhov, P., and M. Spiegel, "VXLAN-GPE Encapsulation for In-situ OAM Data", draft-brockners-ippm-ioam-vxlan-gpe-02 (work in progress), July 2019.

[I-D.huang-nvo3-vxlan-gpe-extension-for-vbng]
Huang, L., Hu, S., Wang, Z., and T. Ao, "VXLAN GPE Extension for Packets Exchange Between Control and User Plane of vBNG", draft-huang-nvo3-vxlan-gpe-extension-for-vbng-01 (work in progress), October 2017.

[I-D.ietf-idr-tunnel-encaps]
Patel, K., Velde, G., and S. Ramachandra, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-14 (work in progress), September 2019.

[I-D.lemon-vxlan-lisp-gpe-gbp]
Lemon, J., Maino, F., Smith, M., and A. Isaac, "Group Policy Encoding with VXLAN-GPE and LISP-GPE", draft-lemon-vxlan-lisp-gpe-gbp-02 (work in progress), April 2019.

Authors' Addresses

Fabio Maino (Editor)
Cisco Systems

Email: fmaino@cisco.com

Larry Kreeger (editor)
Arrcus

Email: lkreeger@gmail.com

Uri Elzur (editor)
Intel

Email: uri.elzur@intel.com