Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 29, 2015

P. Quinn
Cisco Systems, Inc.
R. Manur
Broadcom
L. Kreeger
D. Lewis
F. Maino
M. Smith
Cisco Systems, Inc.
P. Agarwal

L. Yong
Huawei USA
X. Xu
Huawei Technologies
U. Elzur
Intel
P. Garg
Microsoft
D. Melman
Marvell

February 25, 2015

Generic Protocol Extension for VXLAN
draft-quinn-vxlan-gpe-04.txt

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.

Status of this Memo

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.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |R|R|R|R|I|R|R|R|            Reserved                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                VXLAN Network Identifier (VNI) |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: VXLAN Header

3. Generic Protocol Extension for VXLAN (VXLAN GPE)

3.1. VXLAN GPE Header

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |R|R|Ver|I|P|R|O|       Reserved                |Next Protocol  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                VXLAN Network Identifier (VNI) |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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.

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 P=1, the destination UDP port MUST be 4790.

P = 0 indicates that the payload MUST conform to VXLAN as defined in [RFC7348], including destination UDP port.

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 9.2.

This draft defines the following Next Protocol values:

0x1 : IPv4
0x2 : IPv6
0x3 : Ethernet
0x4 : Network Service Header [NSH]

3.3. 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.

3.4. 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.

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.

```
       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                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Opt Ethertype = C-Tag 802.1Q  |     Outer VLAN Tag            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Ethertype = 0x0800            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      Outer IPv4 Header:
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |Version|  IHL  |Type of Service|          Total Length         |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |         Identification        |Flags|      Fragment Offset    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  Time to Live |Protocl=17(UDP)|   Header Checksum             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                       Outer Source IPv4 Address               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                   Outer Destination IPv4 Address              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      Outer UDP Header:
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Source Port         |       Dest Port = 4790        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           UDP Length          |        UDP Checksum           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      VXLAN GPE Header:
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |R|R|Ver|I|P|R|O|       Reserved                |Next Protocol  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                VXLAN Network Identifier (VNI) |   Reserved    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      Payload:
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |      Depends on VXLAN GPE Next Protocol field above.          |
      |    Note that if the payload is Ethernet, then the original    |
      |    Ethernet Frame’s FCS is not included.                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

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

Figure 3: Outer Headers for VXLAN GPE over IPv4

```
       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                       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | Opt Ethertype = C-Tag 802.1Q  |     Outer VLAN Tag            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      | 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         |       Dest Port = 4790        |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           UDP Length          |        UDP Checksum           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      VXLAN GPE Header:
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |R|R|Ver|I|P|R|O|       Reserved                |Next Protocol  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                VXLAN Network Identifier (VNI) |   Reserved    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

      Payload:
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |      Depends on VXLAN GPE Next Protocol field above.          |
      |    Note that if the payload is Ethernet, then the original    |
      |    Ethernet Frame’s FCS is not included.                      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

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

Figure X: Outer Headers for VXLAN GPE over IPv6

Figure 4: 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.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |R|R|0|0|I|1|R|0|       Reserved                |    NP = IPv4  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                VXLAN Network Identifier (VNI) |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Original IPv4 Packet                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: IPv4 and VXLAN GPE

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |R|R|0|0|I|1|R|0|       Reserved                |  NP = IPv6    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                VXLAN Network Identifier (VNI) |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Original IPv6 Packet                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: IPv6 and VXLAN GPE

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |R|R|0|0|I|1|R|0|       Reserved                |NP = Ethernet  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                VXLAN Network Identifier (VNI) |   Reserved    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |               Original Ethernet Frame                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: 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. Acknowledgments

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

9. IANA Considerations

9.1. UDP Port

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

9.2. VXLAN GPE Next Protocol

IANA is requested to set up a registry of "Next Protocol". These are 8-bit values. Next Protocol values 0, 1, 2, 3 and 4 are defined in this draft. New values are assigned via Standards Action [RFC5226].

Next Protocol	Description	Reference
0	Reserved	This document
1	IPv4	This document
2	IPv6	This document
3	Ethernet	This document
4	NSH	This document
5..253	Unassigned	

Table 1

9.3. 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 - Reserved
Bits 2-3 - Version
Bit 4 - Instance ID (I bit)
Bit 5 - Next Protocol (P bit)
Bit 6 - Reserved
Bit 7 - OAM (O bit)
Bits 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.

10. References

10.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

10.2. Informative References

[NSH] Quinn, P. and et al. , "Network Service Header", 2014.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, October 1994.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[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, August 2011.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

[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, August 2014.

Authors' Addresses

Paul Quinn
Cisco Systems, Inc.

Email: paulq@cisco.com

Rajeev Manur
Broadcom

Email: rmanur@broadcom.com

Larry Kreeger
Cisco Systems, Inc.

Email: kreeger@cisco.com

Darrel Lewis
Cisco Systems, Inc.

Email: darlewis@cisco.com

Fabio Maino
Cisco Systems, Inc.

Email: fmaino@cisco.com

Michael Smith
Cisco Systems, Inc.

Email: michsmit@cisco.com

Puneet Agarwal

Email: puneet@acm.org

Lucy Yong
Huawei USA

Email: lucy.yong@huawei.com

Xiaohu Xu
Huawei Technologies

Email: xuxiaohu@huawei.com

Uri Elzur
Intel

Email: uri.elzur@intel.com

Pankaj Garg
Microsoft

Email: Garg.Pankaj@microsoft.com

David Melman
Marvell

Email: davidme@marvell.com