NVO3
Internet-Draft
Intended status: Standards Track
Expires: September 7, 2015

P. Jain
K. Singh
D. Garcia del Rio
Nuage Networks
W. Henderickx
Alcatel-Lucent
V. Bannai
PayPal
R. Shekhar
A. Lohiya
Juniper Networks

March 06, 2015

Generic Overlay OAM and Datapath Failure Detection
draft-jain-nvo3-overlay-oam-03

Abstract

This proposal describes a mechanism that can be used to detect Data Path Failures of various overlay technologies as VXLAN, NVGRE, MPLSoGRE and MPLSoUDP and verifying/sanity of their Control and Data Plane for given Overlay Segment. This document defines the following for each of the above Overlay Technologies:

- Encapsulation of OAM Packet, such that it has same Outer and Overlay Header as any End-System's data going over the same Overlay Segment.

- The mechanism to trace the Underlay that is exercised by any Overlay Segment.

- Procedure to verify presence of any given Tenant VM or End-System within a given Overlay Segment at Overlay End-Point.

Even though the present proposal addresses Overlay OAM for VXLAN, NVGRE, MPLSoGRE and MPLSoUDP, but the procedures described are generic enough to accommodate OAM for any other Overlay Technology.

Copyright Notice

Table of Contents

1. Introduction

VXLAN [RFC7348], NVGRE [I-D.draft-sridharan-virtualization-nvgre], MPLSoGRE [RFC4023] and MPLSoUDP [I-D.draft-ietf-mpls-in-udp] are well known technologies and are used as tunneling mechanism to Overlay either Layer 2 networks or Layer 3 networks on top of Layer 3 Underlay networks. For all above Overlay Models there are two Tunnel End Points for a given Overlay Segment. One End Point is where the Overlay Originates, and other where Overlay Terminates. In most cases the Tunnel End Point is intended to be at the edge of the network, typically connecting an access switch to an IP transport network. The access switch could be a physical or a virtual switch

located within the hypervisor on the server which is connected to End System which is a VM.

This document describes a mechanism that can be used to detect Data Plane failures and sanity of Overlay Control and Data Plane for a given Overlay Segment, and the method to trace the Underlay path that is exercised by any given Overlay Segment.

The document also defines procedures for validating the presence of any given Tenant VM/End-System/End-System or Flow representing the End-System System within a given Overlay Segment.

The proposal describes:

- The mechanism to verify Overlay Control Plane and Data Plane consistency at the Overlay End Point(s), by encapsulating the OAM Packet in exact the same way as that of any End System Traffic that is transported over the Overlay Segment.

- The mechanism to trace the Underlay that is exercised by any Overlay Segment.

- The mechanism to verify presence of any "End-System" in a given Overlay Segment.

The proposal defines the information to check correct operation of the Data Plane, as well as a mechanism to verify the Data Plane against the Control Plane for a given Overlay Segment.

It is important consideration in this proposal to carry Echo Request along same Data Path that any End System's data using the given Overlay Segment takes.

The tenants VM(s) or End System(s) are not aware of the Overlays and as such the need for the verification of the Data Path MUST solely rest with the Cloud Provider. The use cases where the Tenant VM(s) need to be aware of the Data Plane failures is beyond the scope of this document.

2. Terminology

Terminology used in this document:

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 [RFC2119].

When used in lower case, these words convey their typical use in common language, and are not to be interpreted as described in RFC2119 [RFC2119].

OAM: Operations, Administration, and Management

VXLAN: Virtual eXtensible Local Area Network.

NVGRE: Network Virtualization using GRE.

MPLSoGRE: Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)

MPLSoUDP: Encapsulating MPLS in UDP.

Originating End Point: Overlay Segment's Head End or Starting Point of Overlay Tunnel.

Terminating End Point: Overlay Segment's Tail End or Terminating Point of Overlay Tunnel.

VM: Virtual Machine.

VNI: VXLAN Network Identifier (or VXLAN Segment ID)

VSID: Virtual Subnet ID. (for NVGRE)

NVE: Network Virtualized Edge

End System: Could be Tenant VM, Host, Bridge etc. - System whose data is expected to go over Overlay Segment.

Echo Request: Throughout this document, Echo Request packet is expected to be transmitted by Originator Overlay End Point and destined to Overlay Terminating End Point.

Echo Reply: Throughout this document, Echo Reply packet is expected to be transmitted by Terminating Overlay End Point and destined to Overlay Originating End Point.

Other terminologies are as defined in [RFC7348], [I-D.draft-sridharan-virtualization-nvgre], [RFC4023] and [I-D.draft-ietf-mpls-in-udp]

3. Motivation for Overlay OAM

When any Overlay Segment fails to deliver user traffic, there is a need to provide a tool that would enable users, as Cloud Providers to detect such failures, and a mechanism to isolate faults. It may also be desirable to test the data path before mapping End System traffic to the Overlay Segment.

The basic idea is to facilitate following verifications:-

- End-System's data that are expected to go over a particular Overlay Segment actually ends up using the Data-Path represented by given Overlay Segment between the two End-Points.

- To verify the correct value of Overlay Segment Identifier is programmed at Originating and Terminating End Point(s) for a given Overlay Segment. Segment Identifier will be VNI for VXLAN, VSID for NVGRE, MPLS Label for MPLSoGRE and MPLSoUDP.

- The facilitate mechanism to trace the Underlay that is exercised by any Overlay Segment.

- The mechanism to verify presence of any "End-System" in a given Overlay Segment.

To facilitate verification of Overlay Segment or any End-System using the Overlay, this document proposes sending of a Packet (called an "Echo Request") along the same data path as other Packets belonging to this Segment. Echo Request also carries information about the Overlay Segment whose Data Path is to be verified. This Echo Request is forwarded just like any other End System Data Packet belonging to that Overlay Segment, as it contains the same Overlay Encapsulation as regular End System's data.

On receiving Echo Request at the end of the Overlay Segment, it is sent to the Control Plane of the Terminating Overlay End Point, which in-turn would respond with Echo Reply.

To facilitate tracing of the Underlay used by any given Overlay Segment, the document proposes Echo Request/Reply encapsulation in "trace mode", which would allow the user or Cloud Provider to gather information of the Underlay network.

4. Approach

The proposal aims at validating Data Plane and its view of Control Plane for a particular Overlay Segment. To achieve this aim, the draft proposes creating an Overlay OAM Packet which MUST be

encapsulated with the Overlay Header as that of any End-Point data going over the same Overlay Segment. This would guarantee the data-path for OAM Packet follows the same path as that for any End User data going over the same Overlay Segment.

The draft outlines procedures to encode Overlay Header and Inner Ethernet or IP Header based on the type of payload that Overlay is expected to carry.

5. Packet Format

Generic Overlay Echo Request/Reply is a UDP Packet identified by well known UDP Port XXXX. The payload carried by Overlay typically could be either be Layer 2 / Ethernet Frame, or it could be Layer 3 / IP Packet.

5.1. Overlay OAM Encapsulation in Layer 2 Context

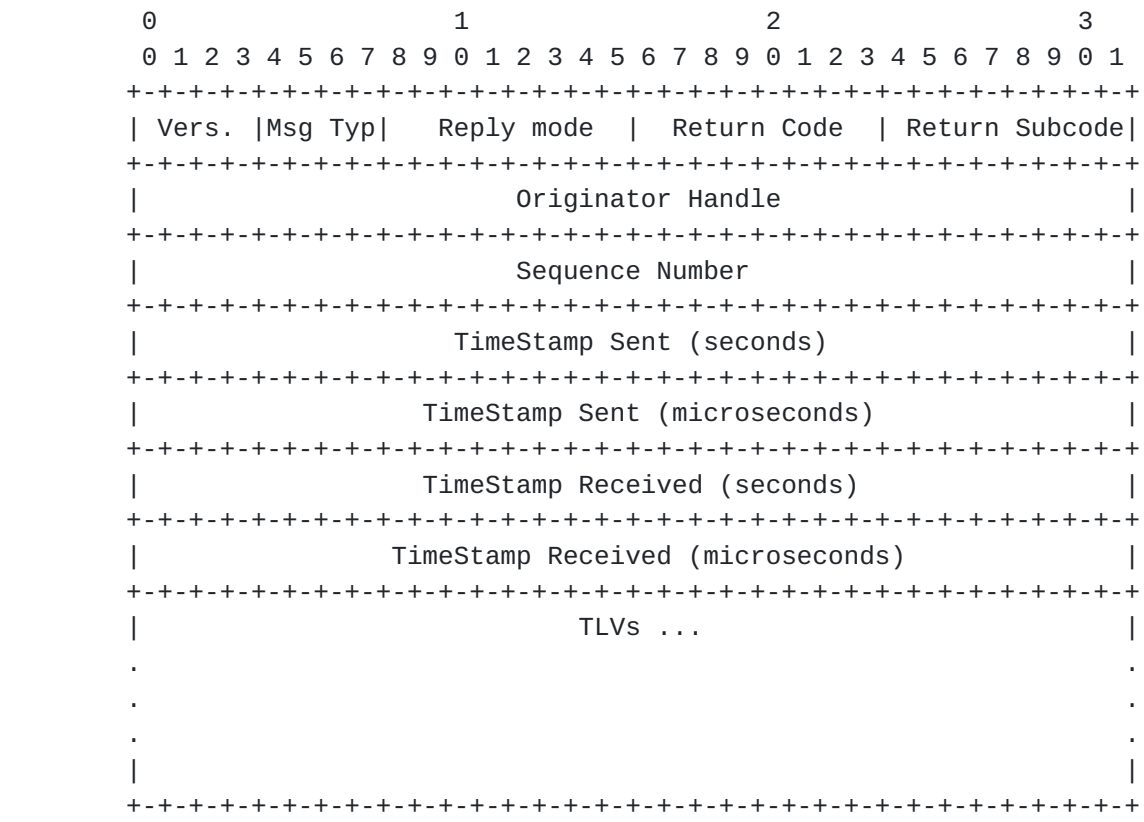
If the encapsulated payload carried by Overlay is of type Ethernet, then the OAM Echo Request packet would have inner Ethernet Header, followed by IP and UDP Header. The payload of inner UDP would be as described in below section "Generic Overlay OAM Packet Format".

5.2. Overlay OAM Encapsulation in Layer 3 Context

If the encapsulated payload carried by Overlay is of type IP, then the OAM Echo Request packet would have inner IP Header, followed by UDP Header. The payload of inner UDP would be as described in below section "Generic Overlay OAM Packet Format".

5.3. Generic Overlay OAM Packet Format

Following is the format of UDP payload of Generic Overlay OAM Packet:



Generic Overlay OAM Packet

The Vers. field represents the PDU encoding version

Value	What it means
0	Initial Version
15	Reserved value

The Message Type is one of the following:-

Value	What it means
1	Echo Request
2	Echo Reply

Reply Mode Values:-

Value	What it means
1	Do not reply
2	Reply via an IPv4/IPv6 UDP Packet
3	Reply via Overlay Segment

Echo Request with 1 (Do not reply) in the Reply Mode field may be used for one-way connectivity tests. The receiving node may log gaps in the Sequence Numbers and/or maintain delay/jitter statistics. For normal operation Echo Request would have 2 (Reply via an IPv4 UDP Packet) in the Reply Mode field.

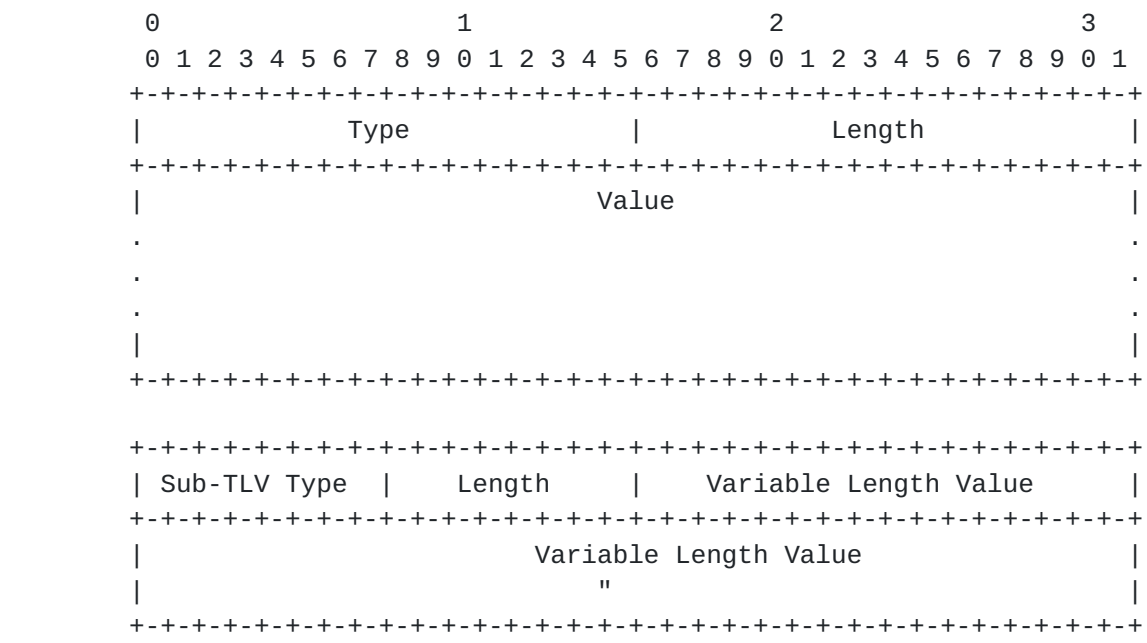
If it is desired that the reply also comes back via Overlay Segment i.e. encapsulated with the Overlay Header, then the Reply Mode filed needs to be set to 3 (Reply via Overlay Segment).

The Originator's Handle is filled in by the Originator, and returned unchanged by the receiver in the Echo Reply (if any). The value used for this field can be implementation dependent, this MAY be used by the Originator for matching up requests with replies.

The Sequence Number is assigned by the Originator of Echo Request and can be (for example) used to detect missed replies.

The TimeStamp Sent is the time-of-day (in seconds and microseconds, according to the sender's clock) in NTP format [NTP] when the VXLAN Echo Request is sent. The TimeStamp Received in an Echo Reply is the time-of-day (according to the receiver's clock) in NTP format that the corresponding Echo Request was received.

TLVs (Type-Length-Value tuples) have the following format:



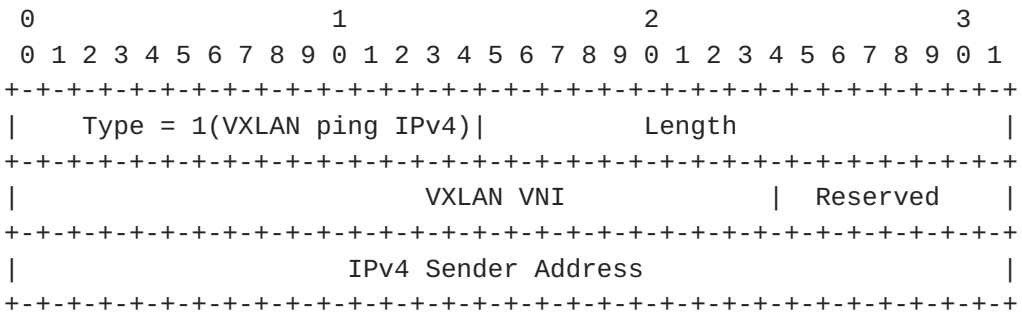
Types are defined below; Length is the length of the Value field in octets. The Value field depends on the Type; it is zero padded to align to a 4-octet boundary. There could be one or many optional Sub-TLV that could be encoded under the TLV.

5.3.1. TLV Types for various Overlay Ping Models

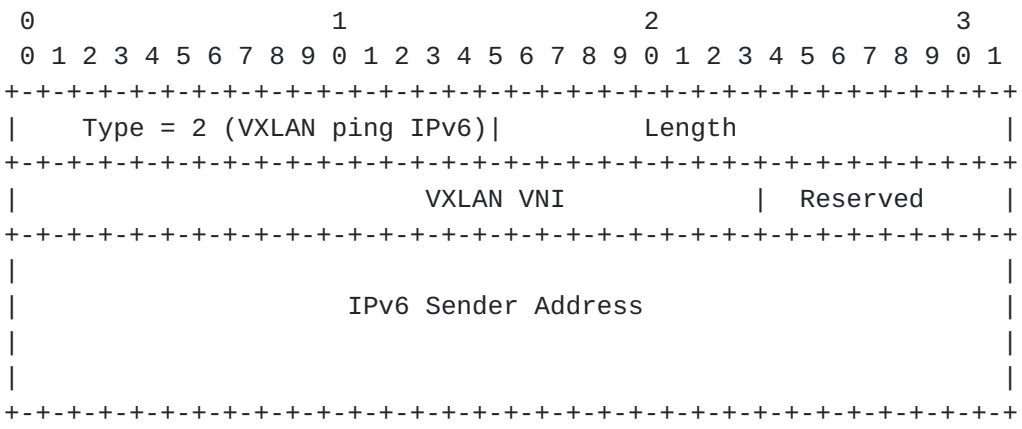
TLV Types:-

Value	What it means
1	VXLAN Segment Ping for IPv4
2	VXLAN Segment Ping for IPv6
3	NVGRE Segment Ping for IPv4
4	NVGRE Segment Ping for IPv6
5	MPLSoGRE Segment Ping for IPv4
6	MPLSoGRE Segment Ping for IPv6
7	MPLSoUDP Segment Ping for IPv4
8	MPLSoUDP Segment Ping for IPv6

5.3.1.1. TLV for VXLAN Ping

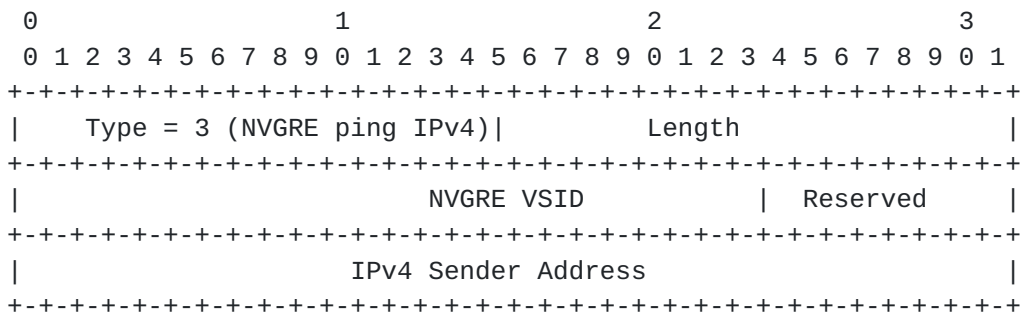


TLV if Sender Address is IPv4

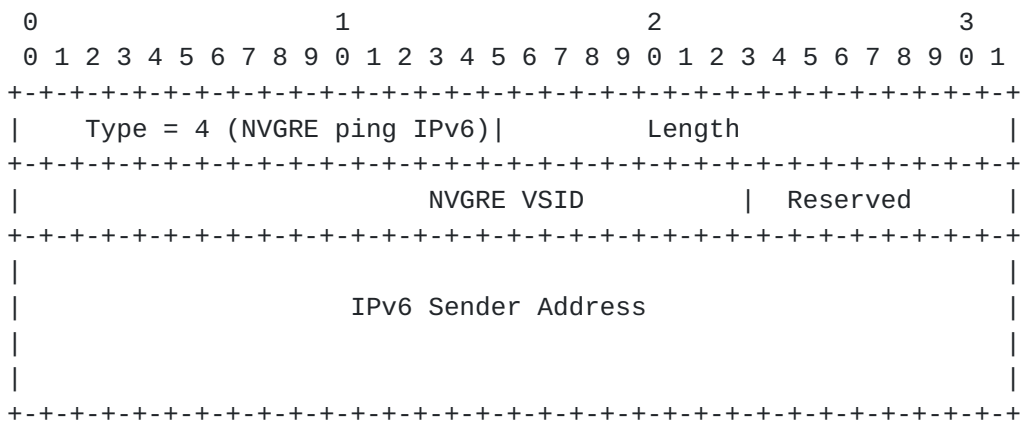


TLV if Sender Address is IPv6

5.3.1.2. TLV for NVGRE Ping

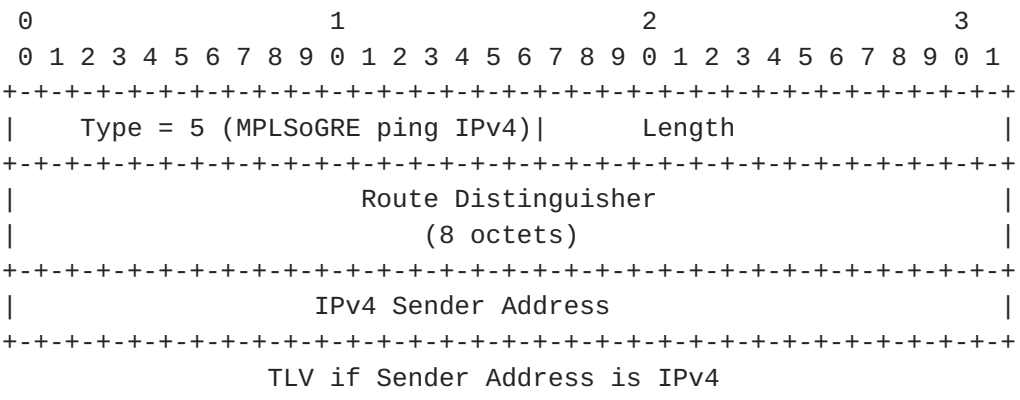


TLV if Sender Address is IPv4

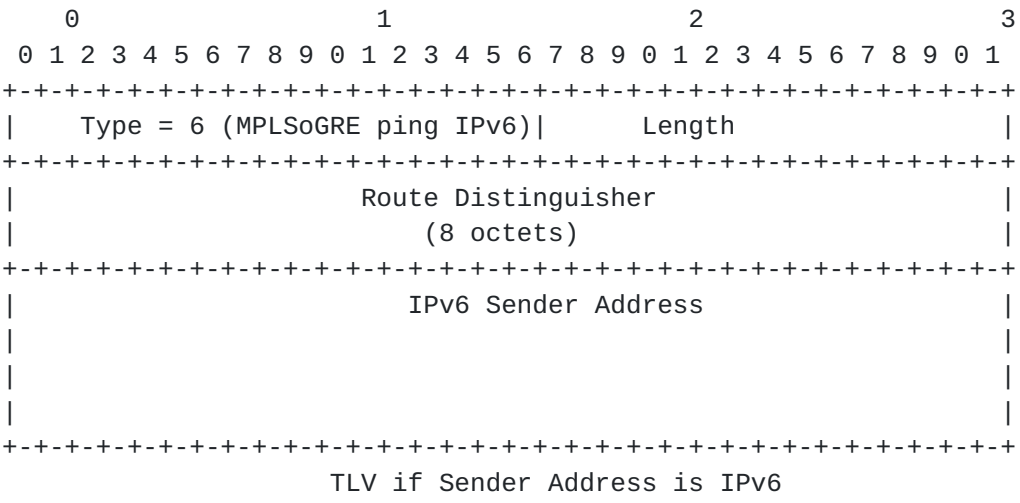


TLV if Sender Address is IPv6

5.3.1.3. TLV for MPLSoGRE Ping



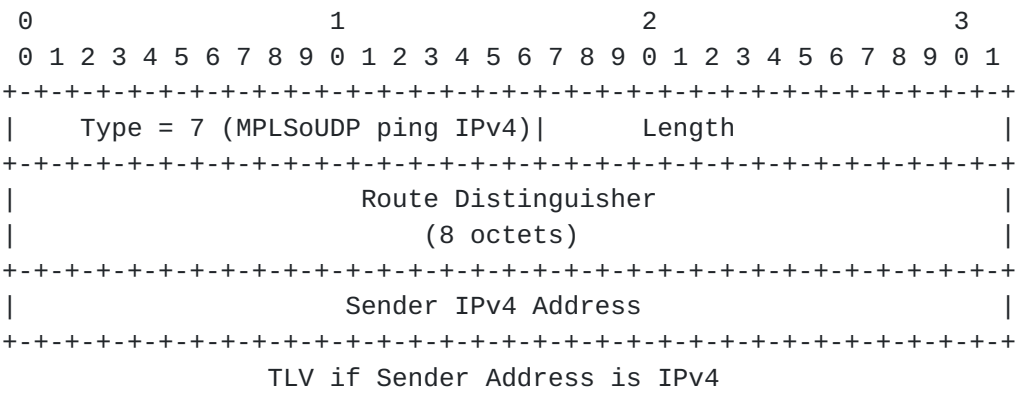
TLV if Sender Address is IPv4



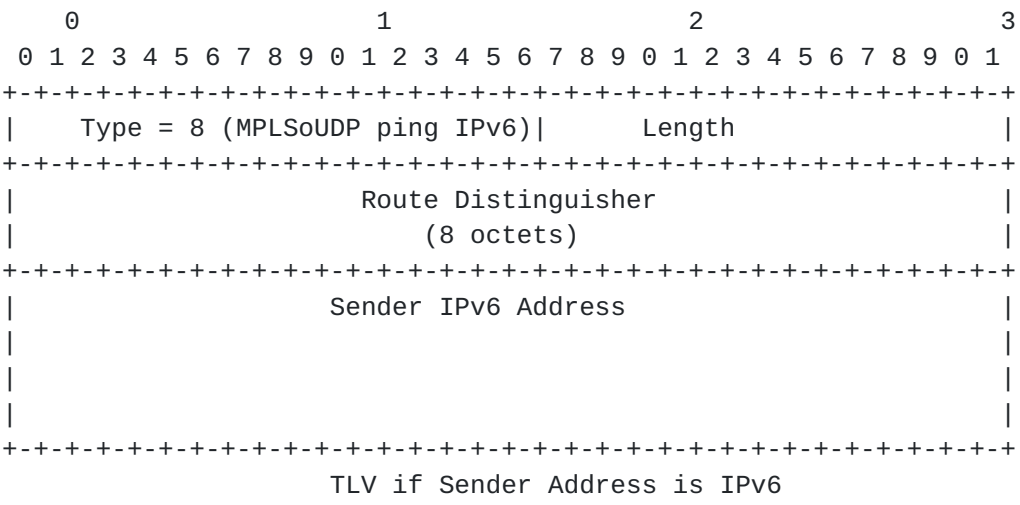
TLV if Sender Address is IPv6

Route Distinguisher is defined as part of [RFC4365]

5.3.1.4. TLV for MPLSoUDP Ping



TLV if Sender Address is IPv4



TLV if Sender Address is IPv6

Route Distinguisher is defined as part of [RFC4365]

6. Return Codes

Sender MUST always set the Return Code set to zero. The receiver can set it to one of the values listed below when replying back to Echo-Request.

Following are the Return Codes (Suggested):-

Value	What it means
0	No return code
1	Malformed Echo Request Received
2	Overlay Segment Not Present
3	Overlay Segment Not Operational
4	Return-Code-OK

7. Procedure for Overlay Segment Ping

Echo Request is used to test Data Plane and its view of Control Plane for particular Overlay Segment. The Overlay Segment to be verified is identified differently for various Overlay Technologies. For VXLAN, VNI is used to identify given Overlay Segment. For NVGRE, VSID is used. For MPLSoGRE and MPLSoUDP the MPLS Stack is used to identify a given Overlay Segment.

For the Data Plane verification, the Overlay Echo Request Packet MUST be encapsulated within the Overlay Header, which is same as that of any End-Point data going over the same Overlay Segment. This would guarantee the data-path for OAM Packet follows the same path as that for any End User data going over the same Overlay Segment.

The payload carried by Overlay typically could be either be Layer 2 or Ethernet Frame, or it could be Layer 3 or IP Packet. Based on the type of payload following is the way inner Header(s) of Echo Request would be encoded.

7.1. Encoding of Inner Header for Echo Request in Layer 2 Context

If the encapsulated payload carried by Overlay is of type Ethernet, then the OAM Echo Request packet would have inner Ethernet Header, followed by IP and UDP Header. The payload of inner UDP would be as described in below section "Generic Overlay OAM Packet Format".

Inner Ethernet Header for the Echo Request Packet MUST have the Destination Mac set to 00-00-5E-90-XX-XX (to be assigned IANA). The Source Mac should be set to Mac Address of the Originating VTEP. However, it is desired that the Inner Source Mac SHOULD not be learnt in the MAC-Table as this represent Control Packet in context of Overlay OAM.

Inner IP header is set with the Source IP Address which is a routable Address of the sender; the Destination IP Address is a (randomly chosen) IPv4 Address from the range 127/8, IPv6 addresses are chosen from the range 0:0:0:0:0:FFFF:127/104. The IP TTL is set to 255.

The inner Destination UDP port is set to xxxx (assigned by IANA for Overlay OAM).

The "Generic Overlay OAM Packet" will now be encoded, with following information.

The sender chooses a Originator's Handle and a Sequence Number. When sending subsequent Overlay Echo Requests, the sender SHOULD increment the Sequence Number by 1.

The TimeStamp Sent is set to the time-of-day (in seconds and microseconds) that the Echo Request is sent. The TimeStamp Received is set to zero. Also, the Reply Mode must be set to the desired reply mode. The Return Code and Subcode are set to zero.

Next, the TLV is Encoded for desired Overlay Type, as per Section "Types of TLVs defined for various Overlay Ping Models"

7.2. Encoding of Inner Header for Echo Request in Layer 3 Context

If the encapsulated payload carried by Overlay is of type IP, then the Encoding of the Echo Request would be same as above Section "Encoding of Inner Header for Echo Request in Layer 2 Context", but without the presence of Inner Ethernet Header.

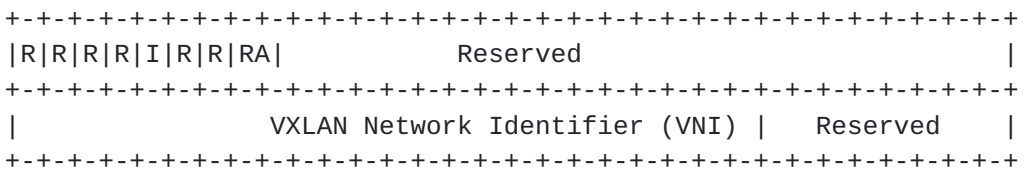
7.3. VXLAN Procedures

7.3.1. Sending VXLAN Echo Request

The Outer VxLAN header for the Echo Request packet follows the encapsulation as defined in [RFC7348]. The VNI is same as that of the VXLAN Segment that is being verified. This would make sure that OAM Packet takes the same datapath as any other End System data going over this VXLAN Segment.

The VXLAN Router Alert option [I-D.draft-singh-nvo3-vxlan-router-alert] MUST be set in the VXLAN header as shown below.

VXLAN Header:



RA: Router Alter Bit (Proposed)

Originating VTEP MAY set the I Bit to 0 in VXLAN Header when sending OAM Frame. This would cause dropping of such VXLAN frames on any Terminating VTEP that does not understand Overlay OAM framework, and prevent sending those frames to End-Systems or VMs.

It is desired to choose the Source UDP port (in the outer header), so as to exercise the same Data-Path as that of the traffic carried over the VXLAN Segment and is left to the implementation.

The Encoding of Inner Header(s) and UDP payload of Generic Overlay OAM Packet is as described in above Sub-Section i.e. "Encoding of Inner Header for Echo Request in Layer 2/Layer 3 Context".

7.3.2. Receiving VXLAN Echo Request

At the Terminating Overlay End Point or VTEP, since the Overlay OAM Packet is exactly same as that of End-System Packet(s). It is important to send OAM packet to Control Plane and prevent it from sending to the End System. The trapping and sending VXLAN Echo Request to the Control Plane is triggered by one of the following Packet processing exceptions: VXLAN Router Alert option, [I-D.draft-singh-nvo3-vxlan-router-alert] the Inner Destination MAC Address of 00-00-5E-90-XX-XX as defined in above section, and the Destination IP Address in the 127/8 Address range for IPv4 Address, or 0:0:0:0:0:FFFF:127/104 for IPv6 Address.

The Control Plane further identifies the Overlay OAM Application by UDP well know destination port xxxx.

Since the VxLAN Router Alert bit is set in VxLAN Header, which signifies the presence of Control Packet. The terminating VTEP SHOULD not learn the Mac address set in the Inner Mac Header of VxLAN Echo Request Packet.

Once the VXLAN Echo Request Packet is identified at Control Plane, it is processed as follows:-

- o General Packet sanity is verified. If the Packet is not well-formed, VTEP SHOULD send VXLAN Echo Reply with the Return Code set to "Malformed Echo Request received" and the Subcode to zero. The header fields Originator's Handle, Sequence Number, and Timestamp Sent are not examined, but are included in the VXLAN Echo Reply message

- o VNI Validation: If there is no entry for VNI, it indicates that there could be a transient or permanent disconnect between Control Plane and data Plane and VTEP needs to report an error with Return Code of "Overlay Segment Not Present" and a Return Subcode of Zero. If the mapping for VNI Exists, but the state is not Operational, VTEP needs to report an error with Return Code of "Overlay Segment Not Operational" If the mapping exists then send VXLAN Echo Reply with a Return Code of "Return-Code-OK", and a Return Subcode of Zero. The procedures for sending the Echo Reply are found in subsection below section.

7.3.3. Sending VXLAN Echo Reply

If the Reply Mode is set to "Reply via an IPv4/IPv6 UDP Packet", the Echo Reply is a UDP Packet. It MUST ONLY be sent in response to Echo Request. The Source IP Address in the Header should be Routable Address of the replier; The Destination IP Address should be IP Address of the Echo Request's Originating End Point or the requester. The destination UDP Port is set to XXXX (assigned by IANA for identifying VXLAN OAM application). The IP TTL is set to 255.

The format of the Echo Reply is the same as the Echo Request. The Originator Handle, the Sequence Number, and TimeStamp Sent are copied from the Echo Request; the TimeStamp Received is set to the time-of-day that the Echo Request is received (note that this information is most useful if the time-of-day clocks on the requester and the replier are synchronized). The replier MUST fill in the Return Code and Subcode, as determined in the previous subsection.

If the Reply Mode is set to "Reply via Overlay Segment", then the Replying Overlay End Point is expected to place Echo Reply packet in-band in the Overlay Segment destined to the Originating Overlay End Point. The detailed encapsulation for this would be covered in next revision of the draft.

7.3.4. Receiving VXLAN Echo Reply

An Originating Overlay End Point should only receive Echo Reply in response to an Echo Request that it sent. When the Reply Mode is "Reply via an IPv4/IPv6 UDP Packet", the Echo Reply would be and IP Packet/UDP Packet, and is identified by the destination UDP Port

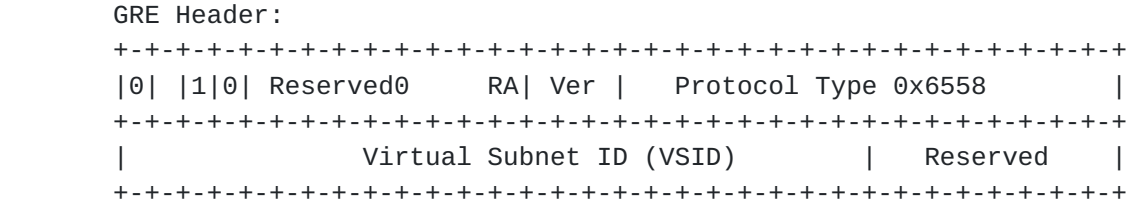
XXXX. The Originating Overlay End Point should parse the Packet to ensure that it is well-formed, then attempt to match up the Echo Reply with an Echo Request that it had previously sent, and the Originator Handle. If no match is found, then it should drop the Echo Reply Packet; otherwise, it checks the Sequence Number to see if it matches.

7.4. NVGRE Procedures

7.4.1. Sending NVGRE Echo Request

The Outer NVGRE header for the Echo Request packet follows the encapsulation as defined in [I-D.draft-sridharan-virtualization-nvgre]. The VSID is same as that of the NVGRE Segment that is being verified. This would make sure that OAM Packet takes the same datapath as any other End System data going over this NVGRE Segment.

The NVGRE Router Alert option [I-D.draft-singh-nvo3-nvgre-router-alert] MUST be set in the NVGRE header as shown below.



RA: Router Alter Bit (Proposed)

The Encoding of Inner Header(s) and UDP payload of Generic Overlay OAM Packet is as described in above Sub-Section i.e. "Encoding of Inner Header for Echo Request in Layer 2/Layer 3 Context".

7.4.2. Receiving NVGRE Echo Request

At the Terminating Overlay End Point, since the Overlay OAM Packet is exactly same as that of End-System Packet(s). It is important to send OAM packet to Control Plane and prevent it from sending to the End System. The trapping and sending NVGRE Echo Request to the Control Plane is triggered by one of the following Packet processing exceptions: NVGRE Router Alert option, [I-D.draft-singh-nvo3-nvgre-router-alert] the Inner Destination MAC Address of 00-00-5E-90-XX-XX as defined in above section, and the Destination IP Address in the 127/8 Address range for IPv4 Address, or 0:0:0:0:0:FFFF:127/104 for IPv6 Address.

The Control Plane further identifies the Overlay OAM Application by UDP well know destination port xxxx.

Since the NVGRE Router Alert bit is set in NVGRE Header, which signifies the presence of Control Packet. The Terminating Overlay End Point SHOULD not learn the Mac address set in the Inner Mac Header of NVGRE Echo Request Packet.

Once the NVGRE Echo Request Packet is identified at Control Plane, it is processed as follows:-

- General Packet sanity is verified. If the Packet is not well-formed, NVGRE End Point SHOULD send NVGRE Echo Reply with the Return Code set to "Malformed Echo Request received" and the Subcode to zero. The header fields Originator's Handle, Sequence Number, and Timestamp Sent are not examined, but are included in the NVGRE Echo Reply message

- VSID Validation: If there is no entry for VSID, it indicates that there could be a transient or permanent disconnect between Control Plane and data Plane and NVGRE End Point needs to report an error with Return Code of "Overlay Segment Not Present" and a Return Subcode of Zero. If the mapping for VSID Exists, but the state is not Operational, NVGRE End Point needs to report an error with Return Code of "Overlay Segment Not Operational" If the mapping exists then send NVGRE Echo Reply with a Return Code of "Return-Code-OK", and a Return Subcode of Zero. The procedures for sending the Echo Reply are found in subsection below section.

7.4.3. Sending NVGRE Echo Reply

The procedure for sending NVGRE Echo Reply are exactly same as defined in above section "Sending VXLAN Echo Reply".

7.4.4. Receiving NVGRE Echo Reply

The procedure for Receiving NVGRE Echo Reply are exactly same as defined in above section "Receiving VXLAN Echo Reply".

7.5. MPLSoGRE Procedures

7.5.1. Sending MPLSoGRE Echo Request

The Outer header of MPLSoGRE for the Echo Request packet follows the encapsulation as defined in [RFC4023]. The MPLS Stack is same as that of the MPLSoGRE Segment that is being verified. This would make sure that OAM Packet takes the same datapath as any other End System data going over this MPLSoGRE Segment.

However, the bottommost Label in MPLS Stack MUST be MPLS Router Alert Label [RFC3032]. This would indicate the Overlay Terminating End Point that the payload is a Control Packet and needs to be delivered to Control Plane.

The Encoding of Inner Header(s) and UDP payload of Generic Overlay OAM Packet is as described in above Sub-Section i.e. "Encoding of Inner Header for Echo Request in Layer 2/Layer 3 Context".

7.5.2. Receiving MPLSoGRE Echo Request

At the Terminating Overlay End Point, since the Overlay OAM Packet is exactly same as that of End-System Packet(s). It is important to send OAM packet to Control Plane and prevent it from sending to the End System. The trapping and sending MPLSoGRE Echo Request to the Control Plane is triggered by one of the following Packet processing exceptions: MPLS Router Alert Label, and the Destination IP Address in the 127/8 Address range for IPv4 Address, or 0:0:0:0:0:FFFF:127/104 for IPv6 Address.

The Control Plane further identifies the Overlay OAM Application by UDP well know destination port xxxx.

Once the MPLSoGRE Echo Request Packet is identified at Control Plane, it is processed as follows:-

- General Packet sanity is verified. If the Packet is not well-formed, MPLSoGRE End Point SHOULD send MPLSoGRE Echo Reply with the Return Code set to "Malformed Echo Request received" and the Subcode to zero. The header fields Originator's Handle, Sequence Number, and Timestamp Sent are not examined, but are included in the MPLSoGRE Echo Reply message

- Segment Validation: If there is no entry for service represented by given Route Distinguisher for the MPLSoGRE Segment, it indicates that there could be a transient or permanent disconnect between Control Plane and Data Plane and MPLSoGRE End Point needs to report an error with Return Code of "Overlay Segment Not Present" and a Return Subcode of Zero. If the entry for service represented by given Route Distinguisher for the MPLSoGRE Segment is present, but is Operationally Down. The End Point needs to report an error with Return Code of "Overlay Segment Not Operational" If the mapping of service represented by given Route Distinguisher for the MPLSoGRE Segment is present and Active, then send MPLSoGRE Echo Reply with a Return Code of "Return-Code-OK".

7.5.3. Sending MPLSoGRE Echo Reply

The procedure for sending MPLSoGRE Echo Reply are exactly same as defined in above section "Sending VXLAN Echo Reply".

7.5.4. Receiving MPLSoGRE Echo Reply

The procedure for Receiving MPLSoGRE Echo Reply are exactly same as defined in above section "Receiving VXLAN Echo Reply".

7.6. MPLSoUDP Procedures

7.6.1. Sending MPLSoUDP Echo Request

The Outer header of MPLSoUDP for the Echo Request packet follows the encapsulation as defined in [I-D.draft-ietf-mpls-in-udp]. The MPLS Stack is same as that of the MPLSoUDP Segment that is being verified. This would make sure that OAM Packet takes the same datapath as any other End System data going over this MPLSoUDP Segment.

However, the bottommost Label in MPLS Stack MUST be MPLS Router Alert Label [RFC3032]. This would indicate the Overlay Terminating End Point that the payload is a Control Packet and needs to be delivered to Control Plane.

It is desired to choose the Source UDP port (in the outer header), so as to exercise the same Data-Path as that of the traffic carried over the MPLSoUDP Segment and is left to the implementation.

The Encoding of Inner Header(s) and UDP payload of Generic Overlay OAM Packet is as described in above Sub-Section i.e. "Encoding of Inner Header for Echo Request in Layer 2/Layer 3 Context".

7.6.2. Receiving MPLSoUDP Echo Request

At the Terminating Overlay End Point, since the Overlay OAM Packet is exactly same as that of End-System Packet(s). It is important to send OAM packet to Control Plane and prevent it from sending to the End System. The trapping and sending MPLSoGRE Echo Request to the Control Plane is triggered by one of the following Packet processing exceptions: MPLS Router Alert Label, and the Destination IP Address in the 127/8 Address range for IPv4 Address, or 0:0:0:0:0:FFFF:127/104 for IPv6 Address.

The Control Plane further identifies the Overlay OAM Application by UDP well know destination port xxxx.

Once the MPLSoUDP Echo Request Packet is identified at Control Plane, it is processed as follows:-

- General Packet sanity is verified. If the Packet is not well-formed, MPLSoUDP End Point SHOULD send MPLSoUDP Echo Reply with the Return Code set to "Malformed Echo Request received" and the Subcode to zero. The header fields Originator's Handle, Sequence Number, and Timestamp Sent are not examined, but are included in the MPLSoUDP Echo Reply message

- Segment Validation: If there is no entry for service represented by given Route Distinguisher for the MPLSoUDP Segment, it indicates that there could be a transient or permanent disconnect between Control Plane and data Plane and MPLSoUDP End Point needs to report an error with Return Code of "Overlay Segment Not Present" and a Return Subcode of Zero. If the entry for service represented by given Route Distinguisher for the MPLSoUDP Segment is present, but is Operationally Down. The End Point needs to report an error with Return Code of "Overlay Segment Not Operational" If the mapping of service represented by given Route Distinguisher for the MPLSoUDP Segment is present and Active, then send MPLSoUDP Echo Reply with a Return Code of "Return-Code-OK".

7.6.3. Sending MPLSoUDP Echo Reply

The procedure for sending MPLSoGRE Echo Reply are exactly same as defined in above section "Sending VXLAN Echo Reply".

7.6.4. Receiving MPLSoUDP Echo Reply

The procedure for Receiving MPLSoGRE Echo Reply are exactly same as defined in above section "Receiving VXLAN Echo Reply".

8. Procedure for Trace

In order to be able to trace the Path that a particular flow in the Overlay takes through the Underlay Network, following mechanism can be used - An overlay Echo Request packet is built and sent using the mechanisms described in the Section "Procedure for Overlay Segment Ping" so that the overlay traceroute follows the same path as the data packet for the overlay segment being traced.

The Echo Request packet in the traceroute mode is sent with the initial TTL set to 1 in the Outer IP header and thereafter incremented by 1 in each successive request. At each transit hop where the TTL expires, an exception is created. Because of this exception, the packet gets delivered to the Control Plane. Control plane can further deliver the packet to the OAM application based on

the TTL exception and the specific UDP port XXXX in the incoming overlay echo request packet. If the transit node has the IP reachability to the destination IP address in the outer IP header, it sends back an overlay echo reply response otherwise the Overlay Echo Request is discarded by the Overlay OAM module on the transit nodes. If the transit node does not support overlay OAM functionality, it will simply generate a regular ICMP TTL exceeded response. This could result into "false negatives". The originating Overlay node that generated the OAM echo request SHOULD try sending the echo request with TTL=n+1, n+2, ... to probe the nodes further down the path to the terminating overlay End-point.

At the originating node, when the Echo Reply from the transit node corresponding to the traceroute query is received, it can correlate the incoming Echo Reply with the traceroute query by matching on the sequence numbers in the Overlay Echo Request/Reply packets. Even if the intermedite node is not capable of generatin an OAM-aware reply, the ICMP TTL exceeded response SHOULD [RFC1812] include enough information of the original packet that allows the sender to identify the request that originated the received response.

Current revision of this draft limits overlay traceroute capability to fault isolation only. A subsequent version of the draft will include mechanisms to trace all possible paths in the underlay that can be used to carry overlay tunnel traffic. Implementations can use a mechanism of randomising/incrementing the source UDP port of the outer IP header as well as incrementing the TTL in order to attempt to cover multiple underlay paths followed by the encapsulated traffic. A system could increment the source UDP port 8 or 16 times, for example, before incrementing the TTL field by one, then repeating the UDP port sweet and continuing.

9. Procedure for End-System Ping

In typical Overlay deployment scenarios there is a desired to check the presence of any given Tenant VM/End-System or Flow representing the End-System System within a given Overlay Segment. This draft proposes the way to achieve it via End-System Ping.

The End-System can be identified at Overlay End Point by either its IP Address, Ethernet MAC Address or combination of IP/MAC Address, as well as an arbitrary packet.

In that case, it would be important to verify the End-System connectivity by procedure which goes over the Overlay Segment from Originating Overlay End-Point and verifies the presence of the End-System at the Terminating Overlay End-Point.

The scope of End-System Ping is solely with the Cloud Provider which owns control of the Overlay End Point(s). It is expected that the Overlay End Point traps this request and checks the Presence of the End-System via its MAC Address, Route or Flow information and replies back. There SHOULD not be a case where the End-System Ping is delivered to the actual End-Point.

9.1. Sub-TLV for End-System Ping

This section defines new set of Sub-TLVs, that needs to be added to be carried in Echo Request/Reply packets to verify presence of one of more End-System(s) which are present in Overlay Segment.

Sub-TLV Types:-

Value	What it means
1	End-System MAC Sub-TLV
2	End-System IPv4 Sub-TLV
3	End-System IPv6 Sub-TLV
4	End-System MAC/IPv4 Sub-TLV
5	End-System MAC/IPv6 Sub-TLV
6	End-System Arbitrary Packet Sub-TLV

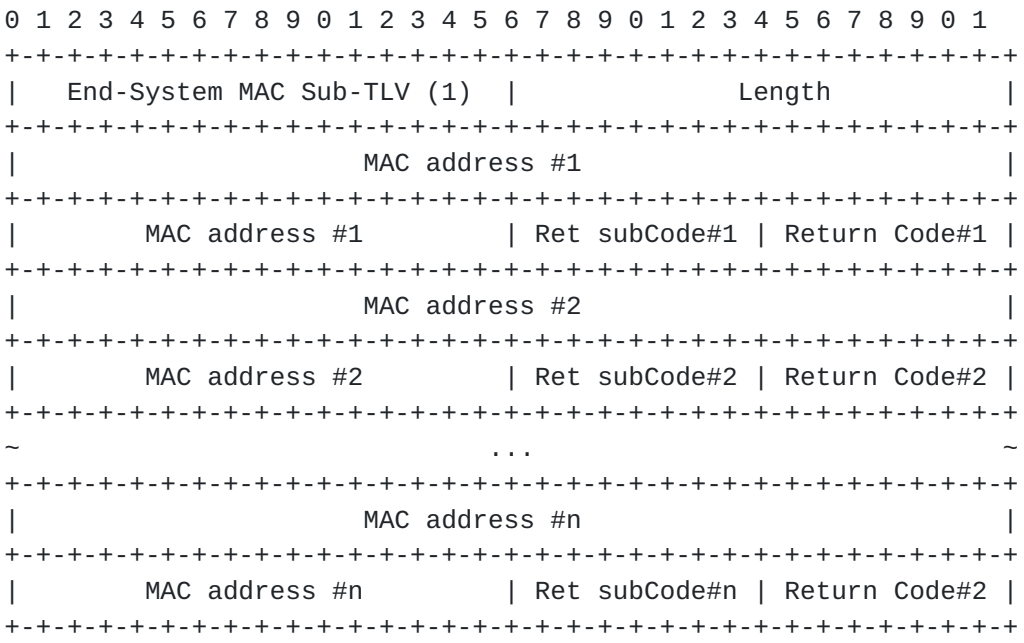
End-System Return Code:-

Value	What it means
1	End-System Present
2	End-System Not Present

End-System Return sub-Code:-

Value	What it means
0	Cannot determine action
1	End system action forward
2	End system action flood
3	End-System action dropped by rules
4	End-System action dropped by other

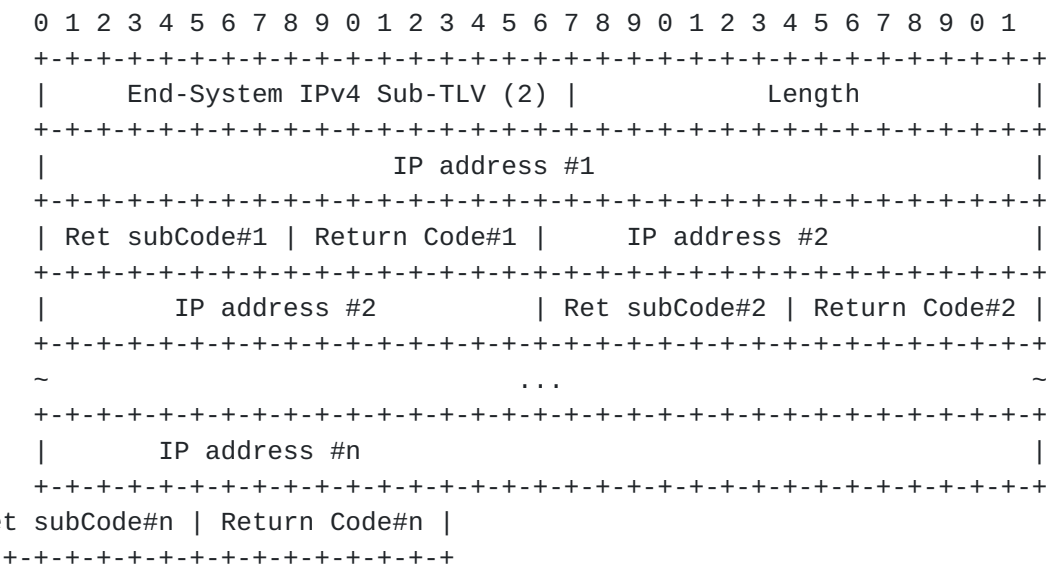
9.1.1. Sub-TLV for Validating End-System MAC Address

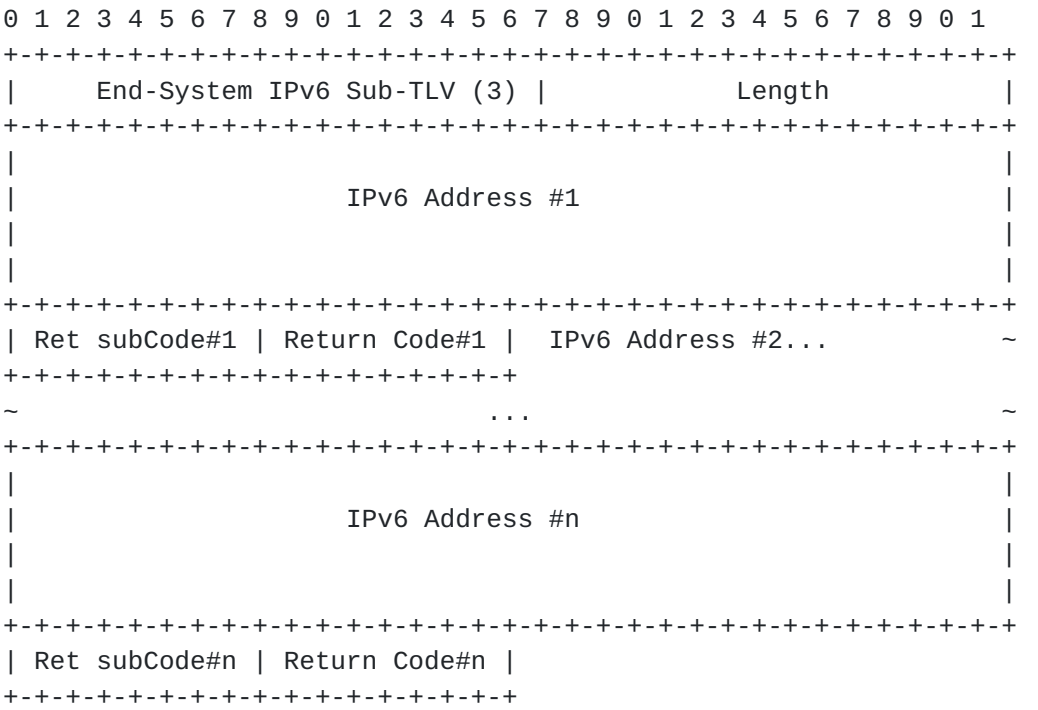


MAC Address: MAC Address of the End-System, that user is interested to validate.

Return Code: Return Code specifying status of End-System at Overlay End Point

9.1.2. Sub-TLV for Validating End-System IP Address

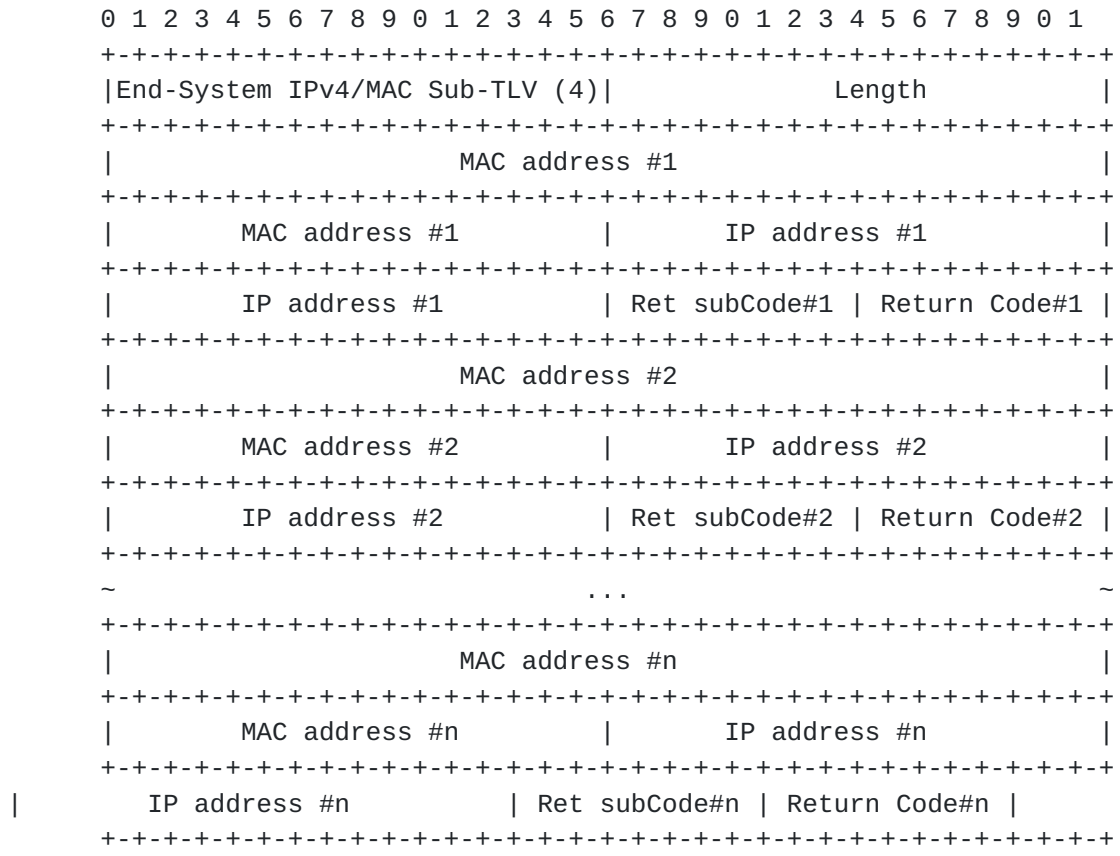


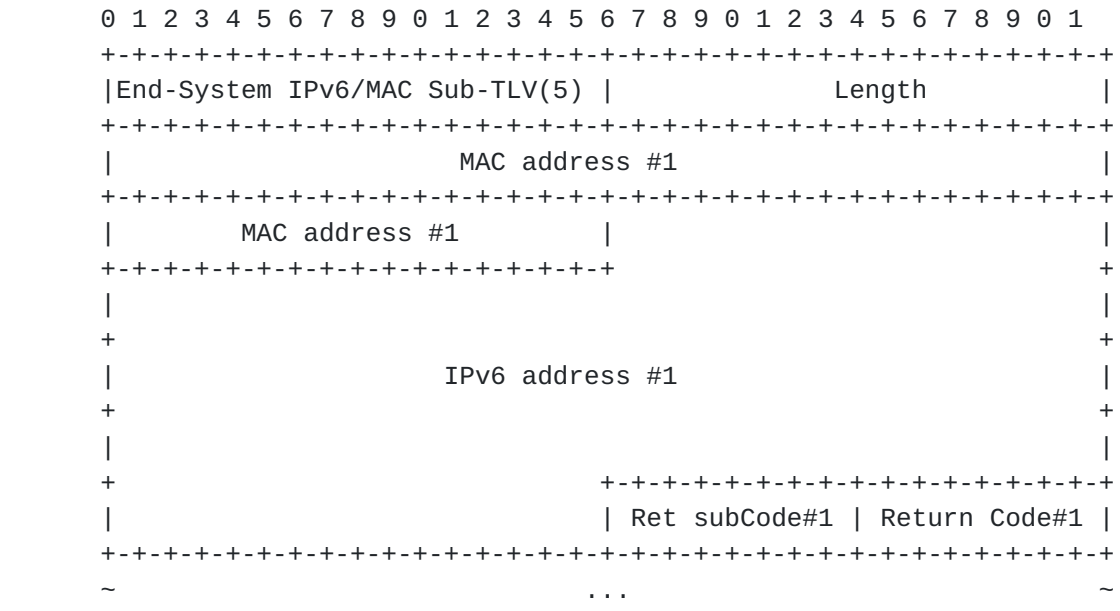


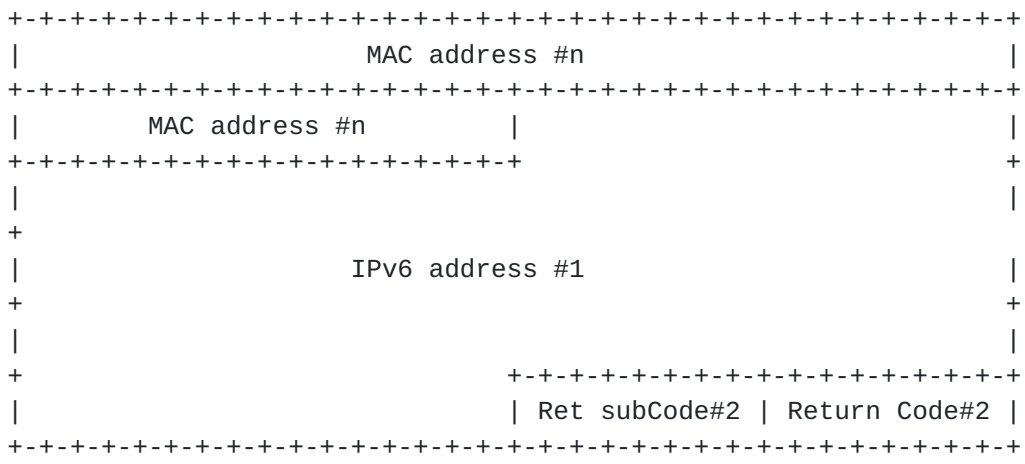
IP Address : IP Address of the End-System, that user is interested to validate.

Return Code: Return Code specifying status of End-System at Overlay End Point

9.1.3. Sub-TLV for Validating End-System MAC and IP Address

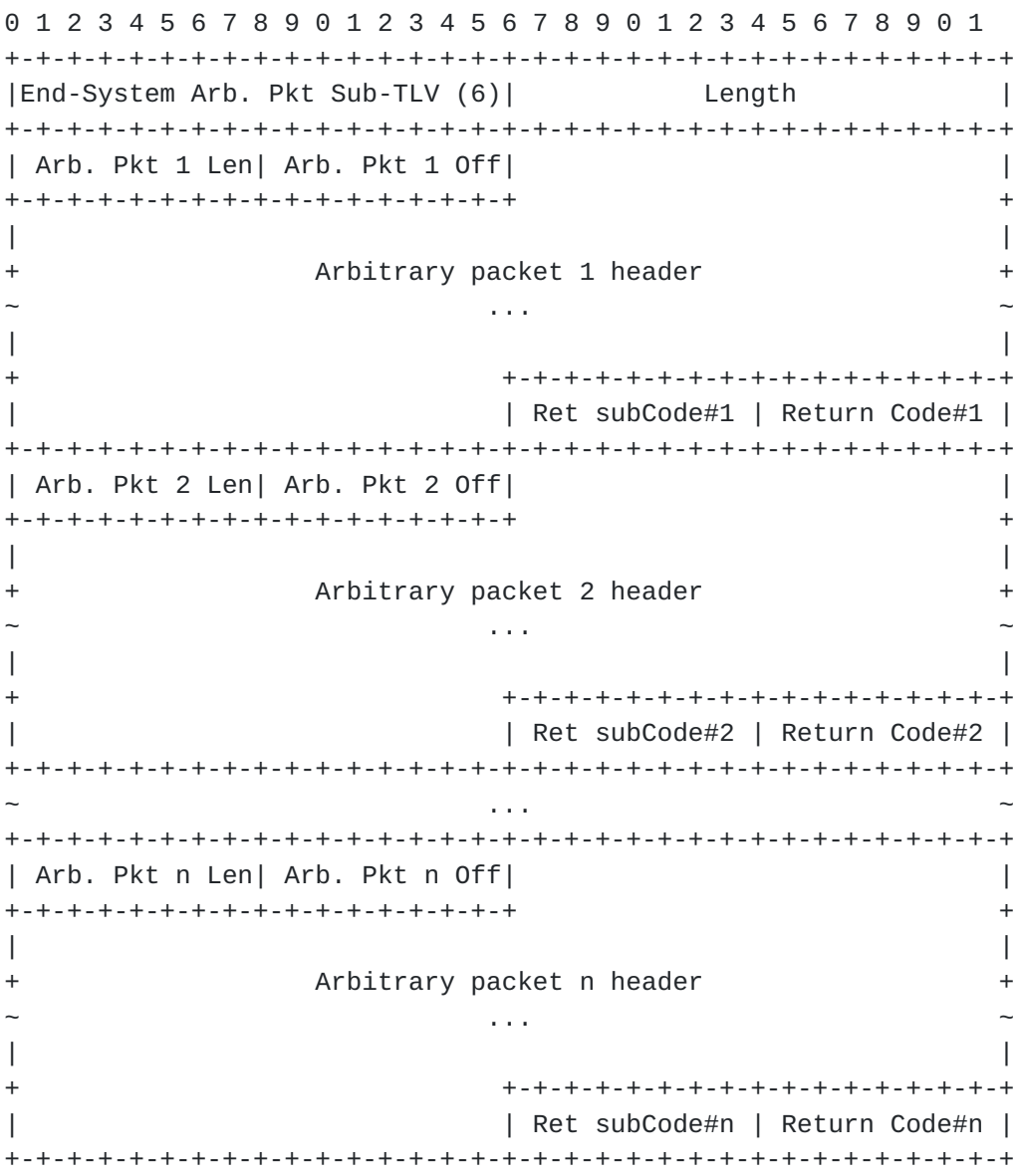






IP Address : IP Address of the End-System, that user is interested to validate.
MAC Address: MAC Address of the End-System, that user is interested to validate.
Return Code: Return Code specifying status of End-System at Overlay End Point

9.1.4. Sub-TLV for Validating End-System Arbitrary packet


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
End-System Arb. Pkt Sub-TLV (6)
Length
Arb. Pkt 1 Len
Arb. Pkt 1 Off
Arbitrary packet 1 header
...
Ret subCode#1
Return Code#1
Arb. Pkt 2 Len
Arb. Pkt 2 Off
Arbitrary packet 2 header
...
Ret subCode#2
Return Code#2
...
Arb. Pkt n Len
Arb. Pkt n Off
Arbitrary packet n header
...
Ret subCode#n
Return Code#n

Field Name explanation

Field Name	Explanation
Arb. Pkt Len	Length in bytes of the arbitrary packet header that follows. (not including the Arbitrary packet offset field)
Arb. Pkt Off	Offset from the start of a regular ethernet frame that the arbitrary data represents. This offset does not include the preamble or start-of-frame delimiter. A value of 0 represents that the data that follows is the fir Arb. Pkt Len bytes of an Ethernet frame starting by the first octect of its DA. A value of 12 means the first 2 octects of the Arbitrary Packet represent the ethertype of the test payload.
Arbitrary Paket	Arbitrary Paket: Arbitrary packet to verify on the remote end. This is a raw bitstream starting by its Destination MAC address -if the Offset is 0- and includes ethertypes, vlan-tags, DSCP values and any other part of the packet that could be used to match against an ACL, flow table or other traffic classification/filtering/forwarding element. This arbitraty packet must be of length Arb Plt Len and represents the ethernet packet present at Arbitrary Packet Offset bytes from the first byte of the Destination MAC address.
Ret subCode	return sub-code specifying the forwarding actions or drops at the Overlay End Point
Return Code	Return Code specifying status of End-System at Overlay End Point

9.2. Sending End-System Ping Request

When it is desired to check presence of a given End-System, the Echo Request Message is prepared as described in above Section "Procedure for Overlay Segment Ping". This packet should compose of Outer Header, Overlay Header, Inner Header, Generic Overlay Header with TLV representing desired Overlay Type (VXLAN, NVGRE, MPLSoGRE or MPLSoUDP). Apart form this the packet should also have one of the Sub-TLV's as defined in above section "Sub-TLV for End-System Ping" to identify the type of End-System Ping that user is interested in.

Because of the above mentioned encapsulation, it would be guaranteed that the packet follows the same Data Path as that of any End-User data going over the given Overlay Segment.

User need to fill in MAC, IP, MAC/IP combination or the Arbitrary packet for the End-System(s) that needs to be validated at the Overlay End Point in the respective Sub-TLV for End-System Ping.

9.3. Receiving End-System Ping Request

On receiving the End-System Ping Request the processing to trap this Packet, and sent it to Control Plane is done by Overlay Terminating End-System as define in above Section "Procedure for Overlay Segment Ping". Once the OAM Packet reaches OAM Application, it is identified as End-System Ping Request by virtue of presence any of the Sub-TLV's as defined in Section "Sub-TLV for End-System Ping".

If the Sub-TLV is of Type "End-System MAC Sub-TLV", the Overlay End Point should iterate through the list of MAC Addresses and verify the presence of individual MAC Address in its Flow Table or MAC Table for the given Overlay Segment.

If the MAC Address is present, it should set the respective End-System's Return Code field in the Sub-TLV to 1 "End-System-Present".

If the MAC Address is not present, it should set respective the End-System's Return Code filed in the Sub-TLV to 2 "End-System-Not-Present".

If the Sub-TLV is of Type "End-System IP Sub-TLV", the Overlay End Point should iterate through the list of IP Addresses and verify the presence of individual IP Address in its Flow Table or Route Table for the given Overlay Segment.

If the IP Address is present, it should set the respective End-System's Return Code field in the Sub-TLV to 1 "End-System-Present".

If the IP Address is not present, it should set respective the End-System's Return Code filed in the Sub-TLV to 2 "End-System-Not-Present".

If the Sub-TLV is of Type "End-System MAC and IP Sub-TLV", the Overlay End Point should iterate through the list of MAC/IP Addresses and verify the presence of individual MAC/IP Combination in its Flow Table or MAC and IP Table for the given Overlay Segment.

If the IP and MAC Address is present, it should set the respective End-System's Return Code field in the Sub-TLV to 1 "End-System-Present".

If the IP and MAC Address is not present, it should set respective the End-System's Return Code filed in the Sub-TLV to 2 "End-System-Not-Present".

If the Sub-TLV is of Type "Arbitrary packet Sub-TLV", the Overlay End Point should iterate through the list of arbitrary packets and verify the presence of individual MAC/Ethertype/VLAN/IP/DSCP/etc Combination in its Flow Table or forwarding tables for the given Overlay Segment. Unused bytes (from a non-zero offset field or short arbitrary packet) should be filled in with 0x00 for whatever fields/bits are needed in order for the system to perform a flow or forwarding table lookup.

If the arbitrary packet is present, it should set the respective End-System's Return Code field in the Sub-TLV to 1 "End-System-Present".

If the arbitrary packet is deemed not present, it should set respective the End-System's Return Code filed in the Sub-TLV to 2 "End-System-Not-Present".

In general, for the TEPs supporting more advanced diagnostics and/or packet match simulation capabilities, the return sub-code SHOULD be set based on the expected fate of the packet according to the following guidelines.

If the provided information (be it MAC, IPv4, IPv6, a combination of MAC/IPv4, MAC/IPv6 or an arbitrary packet) is enough to determine the fate of a hypothetical packet with those addresses and other arbitrary fields, then the expected action SHOULD be reported back to the originator.

If the fate of the packet can not be properly determined, then the respective End-System's sub-Return code should be set to 0, "Cannot determine action"

If the provided information is enough to determine that the packet would be forwarded to the End-System, then the corresponding sub-Return code should be set to 1, "End system action forward"

If the provided information can determine that the packet would be floded (for example, due to a MAC address not present in the forwarding tables and requiring flooding to all ports), then the corresponding sub-Return code should be set to 2, "End system action flood"

If the information provided can determine that the packet would be dropped by ACL rules configured in the system, then the corresponding sub-Return code should be set to 3, "End system action dropped by rules"

Finally, if the information provided can determine that the packet would be dropped by other rules (for example, a configuration setting to disable the flooding of unkwnon packets or such as an anti-spoof filter) then the correspoinding sub-Return code should be set to 4, "End system action dropped by others"

9.4. Sending End-System Ping Reply

The procedure for sending End-System Echo Reply is same as defined in above section "Sending VXLAN Echo Reply". The replier MUST fill Sub-TLV with proper Return Code and sub-code for each element in the End-System Sub-TLV.

9.5. Receiving End-System Ping Reply

An Originating Overlay End Point should only receive Echo Reply for End-System Ping, in response to an Echo Request that it sent. By virtue of presence of End-System Sub-TLV it would identify the status of respective End-System, and report it to the user. The other part of the handling is similar to section "Receiving VXLAN Echo Reply"

10. Security Considerations

TBD

11. Management Considerations

None

12. Acknowledgements

This document is the outcome of many discussions among many people, including Saurabh Shrivastava, Krishna Ram Kuttuva Jeyaram and Suresh Boddapati of Nuage Networks, Jorge Rabadan of Alcatel-Lucent, Inc and Rahul Kasralikar of Juniper Networks, Inc.

13. IANA Considerations

Action-1: This specification reserves a IANA UDP Port Number to be used when sending the Overlay OAM Packet

Action-2: This specification reserves a IANA Ethernet unicast Address for VXLAN/NVGRE Exception handling. This Address needs to be reserved from the block. "IANA Ethernet Address block - Unicast Use"

14. References

14.1. Normative References

[I-D.draft-ietf-mpls-in-udp]
Xu, , Sheth, , Yong, , Pignataro, , Yongbing , , and Li, "Encapsulating MPLS in UDP", May 2013.

[I-D.draft-lasserre-nvo3-framework]
Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", September 2011.

[I-D.draft-singh-nvo3-nvgre-router-alert]
Singh, K., Jain, P., Balus, F., and W. Henderickx, "NVGRE Router Alert Option", May 2013.

[I-D.draft-singh-nvo3-vxlan-router-alert]
Singh, K., Jain, P., Balus, F., and W. Henderickx, "VxLAN Router Alert Option", May 2013.

[I-D.draft-sridharan-virtualization-nvgre]
Sridharan, M., Duda, K., Ganga, I., Greenberg, A., Lin, G., Pearson, M., Thaler, P., Tumuluri, C., Venkataramiah, N., and Y. Wang, "NVGRE: Network Virtualization using Generic Routing Encapsulation", February 2013.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, March 2005.

[RFC4365] Rosen, E., "Applicability Statement for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4365, February 2006.

[RFC4379] Kompella, K. and G. Swallow, "Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures", RFC 4379, February 2006.

[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.

14.2. Informative References

[RFC1812] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

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

[RFC4330] Mills, D., "Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI", RFC 4330, January 2006.

Authors' Addresses

Pradeep Jain
Nuage Networks
755 Ravendale Drive
Mountain View, CA 94043
USA

Email: pradeep@nuagenetworks.net

Kanwar Singh
Nuage Networks
755 Ravendale Drive
Mountain View, CA 94043
USA

Email: kanwar@nuagenetworks.net

Diego Garcia del Rio
Nuage Networks
755 Ravendale Drive
Mountain View, CA 94043
USA

Email: diego@nuagenetworks.net

Wim Henderickx
Alcatel-Lucent
Copernicuslaan 50
Antwerp 2018
Belgium

Email: wim.henderickx@alcatel-lucent.be

Vinay Bannai
PayPal
2211 N. First St,
San Jose 95131
USA

Email: vbannai@paypal.com

Ravi Shekhar
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: rshekhar@juniper.net

Anil Lohiya
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: alohiya@juniper.net