

NV03 working group
Internet Draft
Category: Informational

L. Dunbar
D. Eastlake
Huawei

Expires: April 4 2014

September 20, 2013

NV03 NVA Gap Analysis

draft-dunbar-nvo3-nva-gap-analysis-01

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

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

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

The intent of the draft is to identify the gaps of existing solutions against NVO3's NVE <-> NVA control plane requirement. Through the gap analysis, the document provides a basis for future works that develop solutions for NVE<->NVA control plane.

Table of Contents

1. Introduction	3
2. Terminology	3
3. Overall Requirement for NVE<->NVA Control Plane	4
4. Existing Directory Components	5
4.1. Types of NVA:	5
4.2. Key components of the information kept in the NVA	6
4.3. Mapping Entries Distribution Mechanism	6
4.3.1. Push Mode	6
4.3.2. Pull Mode	8
4.3.3. Hybrid Mode.....	11
5. Redundancy	12
6. Inconsistency Processing.....	12
7. Gap Summary	12
7.1. Features necessary to NVO3 but not present in TRILL ...	12
7.2. Additional detailed requirement applicable to NVO3's NVA	13
8. Security Considerations.....	14
9. IANA Considerations	14
10. Acknowledgements	14
11. References	14
11.1. Normative References.....	14
11.2. Informative References.....	15
Authors' Addresses	15

1. Introduction

The intent of the draft is to identify the gaps of existing solutions against NVO3's requirement for Network Virtualization Authority (NVA). Through the gap analysis, the document provides a basis for future works to develop solutions for (NVA).

The existing solutions analyzed in draft include the LISP mapping database system and TRILL's directory mechanism.

Section 4.5 of [nvo3-problem-statement] describes the back-end Network Virtualization Authority (NVA) that is responsible for distributing the mapping information for entire overlay system. [nvo3-nve-nva-cp-req] defines the requirement for the control plane between NVA and NVE.

There are many similarities between LISP, TRILL [RFC6325] and NVO3, e.g. LISP using IP header to achieve overlay, TRILL using TRILL header to achieve overlay, and NVO3 using L3 headers plus VNID to achieve overlay. This draft analyzes the TRILL directory mechanisms along with some LISP mapping database system that are applicable to NVO3's NVA<->NVE and summarize the gaps.

2. Terminology

The following terms are used interchangeably in this document:

- The terms "Subnet" and "VLAN" because it is common to map one subnet to one VLAN.
- The term ''Directory'' and ''Network Virtualization Authority (NVA)''
- The term ''NVE'' and ''Edge''

Bridge: IEEE Std 802.1Q-2011 compliant device [802.1Q]. In this draft, Bridge is used interchangeably with Layer 2 switch.

DA: Destination Address

DC: Data Center

EoR: End of Row switches in data center. Also known as aggregation switches.

End Station: Guest OS running on a physical server or on a virtual machine. An end station in this document has at

least one IP address and at least one MAC address, which could be in DA or SA field of a data frame.

LISP: Locator/ID Separation Protocol

RBridge: ''Routing Bridge'', an alternative name for a TRILL switch.

NVA: Network Virtualization Authority

NVE: Network Virtualization Edge

SA: Source Address

Station: A node, or a virtual node, with IP and/or MAC addresses, which could be in the DA or SA of a data frame.

ToR: Top of Rack Switch in data center. It is also known as access switches in some data centers.

TRILL: Transparent Interconnection of Lots of Links [RFC6325]

TRILL switch: A device implementing the TRILL protocol [RFC6325]

TS: Tenant System

VM: Virtual Machines

VN: Virtual Network

VNID: Virtual Network Instance Identifier

3. Overall Requirement for NVE<->NVA Control Plane

Section 3.1 of [nvo3-cp-req] describes the basic requirement of inner address to outer address mapping for NVO3. A NVE needs to know the mapping of the Tenant System destination (inner) address to the (outer) address (IP) on the Underlying Network of the egress NVE, in the same way as a TRILL Edge node needing to know how the inner MAC/VLAN is mapped to the egress TRILL edge.

Section 3.1 of [nvo3-cp-req] states that a protocol is needed to provide this inner to outer mapping and VN Context to each NVE that requires it and keep the mapping updated in a timely manner.

Timely updates are important for maintaining connectivity between Tenant Systems.

TRILL's directory mechanism and LISP mapping database system are to achieve the same goal as NVO3's NVE-NVA control plane, i.e. distributing the mapping table that edge nodes use to tunnel traffic across the underlying network. Therefore it is worthwhile to examine the TRILL's directory mechanism and LISP mapping database system, and analyze the gaps.

4. Existing Directory Components

For the ease of description, we match the terminologies used by TRILL/LISP to NVO3. The document will use the NVO3's terminologies as much as possible throughout the document to describe TRILL's directory assistance mechanism.

NVO3	LISP	TRILL
----	-----	-----
NVE	Edge	Edge, TRILL Edge or RBridge Edge
NVA	MapServer	Directory

4.1. Types of NVA:

NVAs can be centralized or distributed with each NVA holding the mapping information for a subset of VNs. Centralized NVA could have multiple entities for redundancy purpose. A NVA could be instantiated on a server/VM attached to a NVE, very much like a TS attached to a NVE, or could be integrated with a NVE. When a NVA is a standalone server/VM attached to a NVE, it has to be reachable via the attached NVE by other NVEs. A NVA can also be instantiated on a NVE that doesn't have any TSs attached. The NVE-NVA control plane for NVA being attached to NVE will require additional functions on NVEs than NVA being instantiated on NVE.

4.2. Key components of the information kept in the NVA

The information held by the TRILL directories is inner-outer address mapping information as well as hosts' VLAN IDs. Same is true for NVO3's NVA. For each TS (or VM), TRILL directory has the following attributes:

1. Inner Address: TS (host) Address family (IPv4/IPv6, MAC, virtual network Identifier MPLS/VLAN, etc)
2. Outer Address: The list of locally attached edges (NVEs); normally one TS is attached to one edge, TS could also be attached to 2 edges for redundancy (dual homing). One TS is rarely attached to more than 2 edges, though it could be possible;
3. Timer for NVEs to keep the entry when pushed down to or pulled from NVEs.
4. Optionally the list of interested remote edges (NVEs). This information is for NVA to promptly update relevant edges (NVEs) when there is any change to this TS' attachment to edges (NVEs). However, this information doesn't have to be kept per TS. It can be kept per VN.

NVO3's NVA will need one additional attribute: VN Context (VN ID and/or VN Name).

4.3. Mapping Entries Distribution Mechanism

A directory can offer services in a Push, Pull mode, or the combination of the two.

4.3.1. Push Mode

Under this mode, Directory Server(s) push the inner-outer mapping for all the entries of the VNs that are enabled on an edge node (NVE). If the destination of a data frame arriving at the Ingress Edge (NVE) can't be found in its inner-outer mapping database that are pushed down from the Directory Server(s) (or NVA), the Ingress edge could be configured with one or more of the following policies:

- simply drop the data frame,

- Using legacy method(s) to forward the data frames to other edges, or
- start the ''pull'' process to get information from Pull Directory Server(s) (or NVA)
When the edge is waiting for reply from Pull process, the edge can either drop or queue the packet.

Again, the VN Context (VNID or VN name) needs to be added for NVO3.

One drawback of the Push Mode is that it usually will push more mapping entries to an edge (NVE) than needed. Under the normal process of edge cache aging and unknown destination address flooding, rarely used entries would have been removed. It would be difficult for Directory Servers (NVA) to predict the communication patterns among TSs within one VN. Therefore, it is likely that the Directory Servers will push down all the entries for all the VNs that are enabled on the NVE.

And with Push there really can't be any source-based policy. It's all or nothing.

4.3.1.1. Requesting Push Service

In the Push Mode, it is necessary to have a way for an edge node (NVE) to request directory server(s) (NVA) to start the pushing process, e.g. when the NVE is initialized or re-started. Or it can be like a routing protocol where it just happens automatically.

Push Directory servers (NVAs) advertise their availability to push mapping information for a particular virtual network to all edges who participate in the VN. There could be multiple directories (NVAs), with each having mapping information for a subset of VNs.

TRILL edge uses modified Virtual Network scoped instances of the IS-IS reliable link state flooding protocol, a.k.a. the ESADI protocol mechanism, to announce all the Virtual Networks in which it is participating to directories (NVAs) who have the mapping information for the VNs. An edge subscribes to push directory information.

The subscription is VN scoped, so that a directory server doesn't need to push down the entire set of mapping entries. Each Push Directory server also has a priority. For robustness, the one or two directories with the highest priority are considered as Active in pushing information for the VN to all edges who have subscribed for that VN.

4.3.1.2. Incremental Push Service

Whenever there is any change in TS' association to an edge (NVE), which can be triggered by TS being added, removed, or de-commissioned, an incremental update can be sent to the edges that are impacted by the change. Therefore, sequence numbers have to be maintained by directory servers (NVA) and edges (NVEs).

If the Push Directory server is configured to believe it has complete mapping information for VN X then, after it has actually transmitted all of its ESADI-LSPs for X it waits its CSNP time (see Section 6.1 of [ESADI]), and then updates its ESADI-Parameters APPsub-TLV to set the Complete Push (CP) bit to one. It then maintains the CP bit as one as long as it is Active.

4.3.2. Pull Mode

Under this mode, an NVE pulls the mapping entry from the directory servers (or NVA) when its cache doesn't have the entry.

The main advantage of Pull Mode is that state is stored only where it needs to be stored and only when it is required. In addition, in the Pull Mode, edge nodes (NVEs) can age out mapping entries if they haven't been used for a certain period of time. Therefore, each edge (NVE) will only keep the entries that are frequently used, so its mapping table size will be smaller than a complete table pushed down from NVA.

The drawback of Pull Mode is that it might take some time for NVEs to pull the needed mapping from NVA. Before NVE gets the response from NVA, the NVE has to buffer the subsequent data frames with destination address to the same target. The buffer could overflow before the NVE gets the response from NVA. However, this scenario should not happen very often in data center environment because most likely the TSs are end systems

which have to wait for (TCP) acknowledgement before sending subsequent data frames. Another option is forward, not flood, subsequent frames to a default location, i.e. forward to a re-encapsulating NVE.

The practice of an edge waiting and dropping packets upon receiving an unknown DA is not new. Most deployed routers today drop packets while waiting for target addresses to be resolved. It is too expensive to queue subsequent packets while resolving target address. The routers send ARP/ND requests to the target upon receiving a packet with DA not in its ARP/ND cache and wait for an ARP or ND responses. This practice minimizes flooding when targets don't exist in the subnet. When the target doesn't exist in the subnet, routers generally re-send an ARP/ND request a few more times before dropping the packets. The holding time by routers to wait for an ARP/ND response when the target doesn't exist in the subnet can be longer than the time taken by the Pull Mode to get mapping from NVA.

4.3.2.1. Pull Requests

Here are some events that can trigger the pulling process:

- o An edge node (NVE) receives an ingress data frame with a destination whose attached edge (NVE) is unknown, or
- o The edge node (NVE) receives an ingress ARP/ND request for a target whose link address (MAC) or attached edge (NVE) is unknown.

Each Pull request can have queries for multiple inner-outer mapping entries.

4.3.2.2. Pull Response

There are several possibilities of the Pull Response:

1. Valid inner-outer address mapping, coupled with the valid timer indicating how long the entry can be cached by the edge (NVE).
The timer for cache should be short in an environment where VMs move frequently. The cache timer can also be configured.

2. The target being queried is not available. The response should include the policy if requester should forward data frame in legacy way, or drop the data frame.
3. The requestor is administratively prohibited from getting an informative response.

If no response is received to a Pull Directory request within a configurable timeout, the request should be re-transmitted with the same Sequence Number up to a configurable number of times that defaults to three.

4.3.2.3. Cache Consistency

It is important that the cached information be kept consistent with the actual placement of VMs. Therefore, it is highly desirable to have a mechanism to prevent NVEs from using the staled mapping entries.

When data at a Pull Directory changes, such as entry being deleted or new entry added, and there may be unexpired stale information at a querying edge (NVE), the Pull Directory MUST send an unsolicited message to the edge (NVE).

To achieve this goal, a Pull Directory server MUST maintain one of the following, in order of increasing specificity.

1. An overall record per VN of when the last returned query data will expire at a requestor and when the last query record specific negative response will expire.
2. For each unit of data (IA APPsub-TLV Address Set) held by the server and each address about which a negative response was sent, when the last expected response with that unit or negative response will expire at a requester.

Note: It is much more important to cache negative reply, because there are many invalid address queries. Study has shown that for each valid ND query, there are 100's of invalid address queries.

3. For each unit of data held by the server and each address about which a negative response was sent, a list of Edges that were sent that unit as the response or sent a negative

response to the address, with the expected time to expiration at each of them.

4.3.2.4. Pull Request Errors

If errors occur at the query level, they MUST be reported in a response message separate from the results of any successful queries. If multiple queries in a request have different errors, they MUST be reported in separate response messages. If multiple queries in a request have the same error, this error response MAY be reported in one response message.

4.3.2.5. Redundant Pull Directories (NVAs)

There could be multiple directories (NVA) holding mapping information for a particular VN for reliability or scalability purposes. Pulling Directories (NVAs) advertise themselves by having the Pull Directory flag on in their Interested VNs sub-TLV [rfc6326bis].

A pull request can be sent to any of them that is reachable but it is RECOMMENDED that pull requests be sent to a server (NVA) that is least cost from the requesting edge (NVE).

4.3.3. Hybrid Mode

For some edge nodes that have great number of VNs enabled and combined number of hosts under all those VNs are large, managing the inner-outer address mapping for hosts under all those VNs can be a challenge. This is especially true for Data Center gateway nodes, which need to communicate with a majority of VNs if not all.

For those Edge nodes, a hybrid mode should be considered. That is the Push Mode being used for some VNs, and the Pull Mode being used for other VNs. It is the network operator's decision by configuration as to which VNs' mapping entries are pushed down from directories (NVA) and which VNs' mapping entries are pulled.

In addition, directory can inform the Edge to use legacy way to forward if it doesn't have the mapping information, or the

Edge is administratively prohibited from forwarding data frame to the requested target.

5. Redundancy

For redundancy purpose, there should be more than one directory (NVAs) that hold mapping information for each VN. At any given time, only one or a small number of push directories is considered as active for a particular VN. All NVAs should announce its capability and priority to all the edges.

6. Inconsistency Processing

If an edge (NVE) notices that a Push Directory server (NVA) is no longer reachable [RFCclear], it MUST ignore any Push Directory data from that server because it is no longer being updated and may be stale.

There may be transient conflicts between mapping information from different Push Directory servers (NVAs) or conflicts between locally learned information and information received from a Push Directory server. TRILL associates a confidence level with address table information so, in case of such conflicts, information with a higher confidence value is preferred over information with a lower confidence. In case of equal confidence, Push Directory information is preferred to locally learned information and if information from Push Directory servers conflicts, the information from the higher priority Push Directory server is preferred.

7. Gap Summary

7.1. Features necessary to NVO3 but not present in TRILL

NVO3's NVA will need one additional attribute: VN context (VN ID and/or VN Name).

For data center networks that don't have IS-IS protocol enabled, other mechanism have to be considered.

7.2. Additional detailed requirement applicable to NVO3's NVA

Here are some of the TRILL's directory detailed requirements that should be considered by NVO3 NVA as well:

- Push Mode:
 - o For redundancy purposes, for each VN there should be multiple NVA entities holding the mapping information for the TSs in the VN. At any given time, only one or a small number of the NVAs are considered as Active for a particular VN. All NVAs should announce their capability and priority to all the edges.
 - o If the destination of a data frame arriving at the Ingress Edge (NVE) can't be found in its inner-outer mapping table that are pushed down from the Directory Server(s) (NVA), the Ingress edge could be configured to:
 - simply drop the data frame,
 - flood it to all other edges that are in the same VN,
 - or
 - start the "pull" process to get information from Pull Directory Server(s) (or NVA)
 - o If an NVE lost its connection to its NVA, it MUST ignore any Push Directory data from that server because it is no longer being updated and may be stale.
 - o When transient conflict occurs: higher priority data take precedence.
- Pull Mode:
 - o The Pull Directory response could indicate that the address being queried is not available in NVA or that the requestor is administratively prohibited from getting an informative response.
 - o The timer for ingress NVE caching should be short in an environment where VMS move frequently. The cache timer could be configured or could be sent along with the Pulled reply from the NVA.
 - o Each Pull request can have multiple queries for different TSs.
 - o It is highly desirable to have a mechanism to prevent NVEs from using the stale mapping entries pulled from NVA.

- o While waiting for query response from NVA, the NVE has to buffer the subsequent data frames with destination address to the same target. The buffer could overflow before the NVE gets the response from NVA.
 - o If no response is received to a Pull Directory request within a configurable timeout, the request should be re-transmitted with the same Sequence Number up to a configurable number of times.
- Hybrid Mode:
- o NVE can be configured to get some VN's mapping entries via push mode and other VN's mapping entries via pull mode.

8. Security Considerations

Accurate mapping of inner address into outer addresses is important to the correct delivery of information. The security of specific directory assisted mechanisms will be discussed in the document or documents specifying those mechanisms.

For general TRILL security considerations, see [RFC6325].

9. IANA Considerations

This document requires no IANA actions. RFC Editor: please delete this section before publication.

10. Acknowledgements

Special thanks to Dino Farinacci for valuable suggestions and comments to this draft.

11. References

11.1. Normative References

As an Informational document, this draft has no Normative References.

[nvo3-nve-nva-cp-req] draft-ietf-nvo3-nve-nva-cp-req-00, "Network Virtualization NVE to NVA Control Protocol Requirements", Kreeger, et al. July 31, 2013.

11.2. Informative References

- [802.1Q] IEEE Std 802.1Q-2011, "IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks", May 2011.
- [802.1Qbg] IEEE Std 802.1Qbg-2012, ''Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks -- Edge Virtual Bridging'', July 2012.
- [RFC826] Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC6325] Perlman, et, al ''RBridge: Base Protocol Specification'', <https://datatracker.ietf.org/doc/rfc6325/>, July, 2011
- [RFC6439] Perlman, et, al ''RBridges: Appointed Forwarders'', <https://datatracker.ietf.org/doc/rfc6439/>, Nov 2011

Authors' Addresses

Linda Dunbar
Huawei Technologies
5430 Legacy Drive, Suite #175
Plano, TX 75024, USA
Phone: (469) 277 5840
Email: linda.dunbar@huawei.com

Donald Eastlake
Huawei Technologies
155 Beaver Street
Milford, MA 01757 USA
Phone: 1-508-333-2270
Email: d3e3e3@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

D. Fan
L. Xia
Huawei
Z. Cao
China Mobile
N. Kim
KT

October 21, 2013

L2TP-VP: Layer Two Tunneling Protocol - Virtualization Profile
draft-fan-l2tp-vp-00

Abstract

This document describes Layer Two Tunneling Protocol (L2TP)'s virtualization profile (L2TP-VP), which reuses session header of L2TP data message to securely support overlay networks for multiple tenants, and simplifies tunnel setup by disabling all kinds of L2TP control messages.

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 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	3
2. Terminology	4
3. L2TP-VP Frame Format	5
4. Control Plane Consideration	7
5. Data Plane Consideration	8
5.1. Address Learning	8
5.2. Forwarding	8
5.2.1. Unicast Traffic	8
5.2.2. Broadcast/Unknown/Multicast(BUM) Traffic	8
5.3. MTU Configuration	8
5.4. Qos Consideration	9
5.5. ECMP	9
6. Management Plane Consideration	10
7. Deployment Consideration	11
8. Security Considerations	12
9. IANA Considerations	13
10. References	14
10.1. Normative References	14
10.2. Informative References	14
Authors' Addresses	15

1. Introduction

Traditional data centre network uses global VLAN ID to distinguish different tenants. Usually, a tenant consumes several VLAN IDs, for example, one for web server, one for application server and one for database server. When the number of tenants increases, the number of available VLAN IDs becomes rare.

When services provisioned from cloud via Internet becomes popular, a tenant's local area network needs to securely and smoothly reach anywhere via Internet if it wants. For example, a tenant can access its office IT services hosted in cloud data centers consisting of many geographically dispersed physical data centers. Traditional L2VPN technologies seems to be burdened heavily.

Layer Two Tunneling Protocol - Version 3 (L2TPv3) [RFC3931] is a mature and practical protocol that provides secure remote access service and layer 2 over IP service, but L2TPv3 also uses complicated control messages to setup tunnel. At the same time, L2TPv3 uses dynamical session id that is controlled by signaling mechanism and only has local significance, i.e., L2TPv3 is complex and does not support multiple tenants though it provides basic overlay functions.

This document will describe Layer Two Tunneling Protocol (L2TP)'s virtualization profile (L2TP-VP), which reuses session header of L2TP data message to securely support overlay networks for multiple tenants, and simplifies tunnel setup by disabling all kinds of L2TP control messages. Essentially, L2TP-VP defines a subset of L2TPv3 via fine and back-compatible reuse, and then extends L2TP's usage to network virtualization. L2TP is widely deployed and used whatever for operators' network or enterprises' network, L2TP-VP brings L2TP to the entire cloud network by further covering data center network.

2. Terminology

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

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

3. L2TP-VP Frame Format

L2TPv3 message format is specified in [RFC3931]. In order to support virtualization and reduce complexity from the control messages, two key fields are added into L2TP header to carry the original payload type and TNI (Tenant Network Identifier). The example of packet format for Ethernet encapsulation in L2TP-VP is shown in Figure 1.

[illegible]

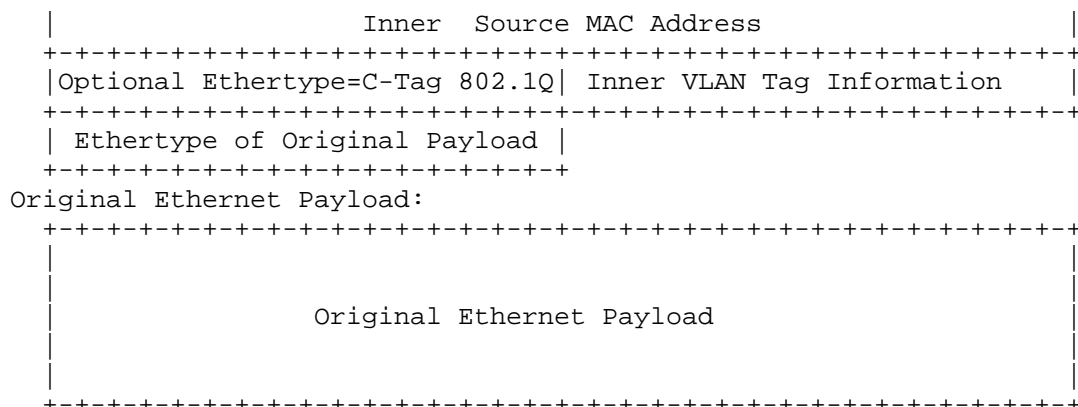


Figure 1 L2TP-VP Encapsulation Frame Format

The original Ethernet frame is encapsulated with L2TP-VP Header, Outer IP Header and Outer Ethernet Header.

L2TP-VP Header:

- o Type field : A 16-bit field is used to carry original payload type (e.g., frame type). Payload type can be Layer2 type such as ATM, FR, Ethernet, etc. It also can be Layer3 type such as IPv4 , IPv6 ,etc. In Figure 1 the type of original packet is Ethernet;
- o TNI field : A 24-bit field allows up to 16 million tenants in the same management domain. The packets with different TNI will be isolated logically;
- o Cookie field : The optional Cookie field inherits all the functions from cookie field in [RFC3931] . It is used to check the association of a received data message with TNI.Only need to change its length to be 32-bit.

Outer IP Header: Both IPv4 and IPv6 can be used as encapsulation IP header. Figure 1 shows an example of IPv4. The source IP address is filled with IP address of L2TP-VP endpoint which encapsulates the original packet with L2TP-VP frame format. The destination IP address is unicast address obtained by lookup of address table. Also it may be a multicast address representing this packet may be used for address learning.

Outer Ethernet Header: The destination MAC address in Figure 1 may be the address of next hop device. The Optional Vlan Tag may be used to limit the area of the broadcast.

4. Control Plane Consideration

In order to reduce complexity coming from control messages, there is no separate control plane in L2TP-VP. All kinds of control messages defined in [RFC3931] are disabled. All tunnel endpoints are expected to be configured by management plane(e.g., OSS).

5. Data Plane Consideration

5.1. Address Learning

For the E2E link and tunnel setup of L2TP-VP overlay network, the forwarding information including tenant systems' address, and its associated L2TP-VP endpoint address and TNI should be populated in the network. There are several options to support address learning:

- o Through the management plane, L2TP-VP endpoints will be configured part or all of the address table;
- o L2TP-VP endpoints directly acquire the forwarding information through data plane by flooding mechanism;
- o L2TP-VP endpoints join the multicast group and populate the forwarding information to the other endpoints in the same virtual network by the multicast tree.

5.2. Forwarding

5.2.1. Unicast Traffic

Ingress L2TP-VP endpoint firstly gets the destination address from the unicast traffic, then obtains IP address of the egress endpoint and the TNI by lookup of address table, at last encapsulates the original packet in L2TP-VP frame format. The source IP address in outer IP header is filled with its own IP address and the destination IP address is filled with egress endpoint's IP address.

5.2.2. Broadcast/Unknown/Multicast(BUM) Traffic

There are several proven methods to process BUM traffic.

One method needs the multicast support of underlay network. All BUM traffic originating from within a TNI is terminated by the L2TP-VP endpoint, then encapsulated and sent to the assigned multicast address. The binding relation of the TNI and the multicast address of underlay network can be configured by the management plane.

Another method is ingress replication. One BUM frame in a TNI can be replicated to multiple unicast frames which will be sent to all the egress L2TP-VP endpoints in the same TNI.

5.3. MTU Configuration

L2TP-VP overlay header can cause the MTU of the path to the egress tunnel endpoint to be exceeded. Here lists some solutions:

- o Modifying the MTU support configuration in the network devices, including L2TP-VP endpoints and other network devices which will transmit the encapsulation packets;
- o Classical ICMP-based MTU Path Discovery [RFC1191] [RFC1981] or Extended MTU Path Discovery techniques such as defined in [RFC4821].

5.4. Qos Consideration

QoS of underlay network can be provided without problem due to the fact that it's an IP network.

QoS of the overlay network may need to support the mapping of CoS marking between different network layers (e.g., Tenant Systems, Overlays, and/or Underlay) in L2TP-VP endpoints, for enabling each networking layer to independently enforce its own CoS policies.

TS's QoS fields (e.g. IP DSCP and/or Ethernet 802.1p) and policies can be defined to indicate application level CoS requirements. L2TP-VP endpoint can use the new service CoS fields in the overlay header to indicate the proper service CoS to be applied across the overlay network. This field can be mapped from the TS's QoS fields or other mechanism (e.g. DPI).

5.5. ECMP

Because the outer header is standard IP header, the L2TP-VP endpoint SHOULD provide ECMP. Basically the L2TP-VP endpoint uses a hash of various fields of the outer Ethernet header and outer IP header, furthermore it can use the fields of L2TP-VP header or even inner original packet. And the endpoint can select different fields for hash according to the requirement.

6. Management Plane Consideration

Management plane is needed to configure access type, TNI, QoS, Cookie, etc. In some scenarios, management plane should support to configure the forwarding information or policies for data plane and control plane , such as routing table, address table, etc. Management plane can be OSS or SDN controller.

7. Deployment Consideration

TBD.

8. Security Considerations

Like L2TPv3, L2TP-VP continues to adopt Cookie Field as an additional check to the received packet. A 32-bit random field is difficult to be cracked so that it can afford protection against brute-force, blind and insertion attacks.

When the network is open network and someone can sniff the whole traffic through the network , it will need other security measures. Traditional security mechanisms based on IP technique will provide authentication/encryption function ,such as IPSec.

9. IANA Considerations

TBD.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.
- [RFC3931] Lau, J., "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", March 2005.

10.2. Informative References

- [FRAMEWORK] Lasserre, M., "Framework for DC Network Virtualization", ID draft-ietf-nvo3-framework-03, July 2013.
- [REQ] Bitar, N., "NVO3 Data Plane Requirements", ID draft-ietf-nvo3-dataplane-requirements-01, July 2013.

Authors' Addresses

Duoliang Fan
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: fanduoliang@huawei.com

Liang Xia
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: frank.xialiang@huawei.com

Zhen Cao
China Mobile
Xuanwumenxi Ave. No.32 , Xicheng District
Beijing, 100053
China

Email: zehn.cao@gmail.com, caozhen@chinamobile.com

Namgon Kim
KT
463-1 Jeonmin-Dong, Yuseoung-Gu Daejeon, 305-811
Korea

Email: ng.kim@kt.com

Networking Virtualization Overlays Working Group; LISP WorkinY. Hertoghs
Internet-Draft F. Maino
Intended status: Informational V. Moreno
Expires: April 22, 2014 Cisco Systems
M. Smith
Insieme Networks
D. Farinacci
lispers.net
October 19, 2013

A Unified LISP Mapping Database for L2 and L3 Network Virtualization
Overlays
draft-hertoghs-nvo3-lisp-controlplane-unified-00

Abstract

The purpose of this draft is to document how the Locator/ID Separation Protocol (LISP) Control Plane can be used to offer a unified (offering L2 AND L3) Overlay solution that matches the framework and requirements of Network Virtualization over L3 (NVO3). This information is provided as input to the NVO3 analysis of the suitability of existing IETF protocols to the NVO3 requirements.

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 RFC 2119 [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 22, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Definition of Terms	4
3. NVO3 Framework and LISP	4
3.1. NVO3 Generic Reference Model	4
3.2. NVE Reference Model	4
3.2.1. Types of NVE's	4
3.2.2. Multihoming aspects	7
3.2.3. External connectivity aspects	8
3.2.4. Optimal Forwarding aspects	8
3.2.5. VM Mobility aspects	9
3.3. LISP dataplane options and NVO3 dataplane requirements .	12
3.3.1. Native LISP Data Plane	12
3.3.2. LISP with Generic Protocol Extension (LISP-GPE) . . .	14
3.3.3. VxLAN-GPE	15
3.3.4. L2 only solutions such as VxLAN and nvGRE	15
3.4. NVO3 control plane requirements and LISP	16
3.4.1. Inner to Outer Address Mapping	16
3.4.2. Underlying network Multi-Destination Delivery	16
3.4.3. VN connect/disconnect	17
3.4.4. VN name to VN ID Mapping	18
3.4.5. LISP Control Plane Characteristics in an NVO3 context	18
3.5. NVO3 OAM Requirements and LISP	19
3.6. NVO3 Management Plane Requirements and LISP	20
3.7. Summary	20
4. IANA Considerations	20
5. Security Considerations	20
6. Acknowledgements	20
7. References	21
7.1. Normative References	21
7.2. Informative References	21
Authors' Addresses	24

1. Introduction

The purpose of this draft is to provide a mapping between the Network Virtualization over L3 (NVO3) framework [I-D.ietf-nvo3-framework] and the Locator/ID Separation Protocol (LISP) [RFC6830], and in particular how LISP components map to the reference models defined therein. This document extends the scope of [I-D.maino-nvo3-lisp-cp] to cover the case of a unified overlay that includes L2 and L3 services.

LISP is a flexible map and encap framework that can be used for overlay network applications, including Data Center Network Virtualization. The LISP framework provides two main tools for NVO3:

1. A Data Plane that specifies how Endpoint Identifiers (EIDs) are encapsulated in Routing Locators (RLOCs), and
2. A Control Plane that specifies the interfaces to the LISP Mapping System [RFC6833]. The LISP Mapping system provides the mapping between EIDs and RLOCs.

LISP can be leveraged to offer services to both Physical and Virtual endpoints, and is architecturally EID address family agnostic. Some flows will be across an L3 overlay (using IP addresses as EIDs), and other flows will be across an L2 overlay (using MAC addresses as EIDs).

If certain requirements are met within the architecture, the choice of whether a flow is sent across the L2 overlay versus across the L3 overlay is not mapped one to one to the choice between intra subnet (bridging) and inter subnet (routing) forwarding. This allows the network administrator to offer infrastructure spanning subnets to its tenants, while not forcing them to deploy infrastructure-wide broadcast domains.

This document will focus on how to offer a unified L2 and L3 overlay, where both L2 and L3 services can be offered to the tenant's traffic simultaneously.

The draft will elaborate on achieving multi homing of Tenant Systems (TS), as well as optimal routing and forwarding, including how VM mobility is achieved and optimal traffic forwarding is achieved.

Although the LISP specification uses a specific data plane, its control plane can be decoupled fairly easily from the data plane and thus can support various data plane encapsulations. After describing the various data plane options, this document also addresses the NVO3 data plane requirements [I-D.ietf-nvo3-dataplane-requirements].

The document continues to lay out how the NVO3 control plane requirements [I-D.ietf-nvo3-nve-nva-cp-req] are addressed.

Finally this document will provide security considerations in Section 5

2. Definition of Terms

Flood-and-Learn: the use of dynamic (data plane) learning in VXLAN to discover the location of a given Ethernet/IEEE 802 MAC address in the underlay network.

For definition of NVO3 related terms, notably Tenant System (TS), Virtual Network (VN), Virtual Network Identifier (VNI), Network Virtualization Edge (NVE), Network Virtualization Authority (NVA), Data Center (DC), please consult [I-D.ietf-nvo3-framework].

For definitions of LISP related terms, notably Map-Request, Map-Reply, Ingress Tunnel Router (ITR), Egress Tunnel Router (ETR), Endstation Identifier (EID), Routing Locator (RLOC), Map-Server (MS) and Map-Resolver (MR) please consult the LISP specification [RFC6830].

3. NVO3 Framework and LISP

3.1. NVO3 Generic Reference Model

[I-D.maino-nvo3-lisp-cp] provides a mapping of the NVO3 generic reference model [I-D.ietf-nvo3-framework] onto the LISP architecture. In this document we will focus on the unified L2/L3 LISP control plane, and on how it will optimize forwarding .

3.2. NVE Reference Model

The LISP NVE Reference Model is described in [I-D.maino-nvo3-lisp-cp]. This section will look at the different types of NVEs: L2-only, L3-only, and unified L2/L3, as well as looking at VM Mobility, Multi-homing, optimal forwarding and external connectivity aspects. How TSes connect to the network is described in Section 3.4.3.

3.2.1. Types of NVE's

[RFC6830] is defined as a L3 NVE, and it can be enhanced to support L2 NVEs.

The separation of the L2 NVE and L3 NVE functions can be based on the nature of the packets: bridged packets are assigned to the L2 NVE

function, while routed packets are assigned to the L3 NVE function. Therefore these discrete functions could live on discrete networking nodes.

However, it is possible to use LISP as an unified control plane, that combines and co-locates the function of L2/L3 NVE onto a single node. The network administrator can choose which traffic will be forwarded across each service type.

3.2.1.1. L2 only NVE

[I-D.smith-lisp-layer2] describes an encapsulation method for carrying Ethernet and IEEE 802 media access control (MAC) frames within the Locator/ID Separation Protocol (LISP). As described in [I-D.maino-nvo3-lisp-cp] MAC addresses are used as EIDs in an L2 only NVE. As control plane learning is used, only broadcast and multicast traffic needs mult-destination support from the underlay.

The frame format defined in [I-D.mahalingam-dutt-dcops-vxlan], has a header compatible with the LISP data path encapsulation header, when MAC addresses are used as EIDs, as described in section 4.12.2 of [I-D.ietf-lisp-lcaf].

The LISP control plane is extensible, and can support non-LISP data path encapsulations such as NVGRE [I-D.sridharan-virtualization-nvgre], or other encapsulations that provide support for network virtualization.

3.2.1.2. L3 only NVE

LISP is defined as a virtualized IP routing and forwarding service in [RFC6830], and as such can be used to provide L3 NVE services.

3.2.1.3. Unifed L2/L3 NVE

When using a unified L2/L3 NVE, IP EIDs are registered to the LISP mapping system with the MAC Address of the Tenant System (TS) as an additional RLOC (next to the NVE RLOC), through the methods defined in [I-D.ietf-lisp-lcaf], by encoding Key/Value Pairs. MAC Address based EIDs will also be registered for TSes that are transmitting non-IP based traffic. TSes that send out both IP and non-IP traffic will therefore be registered twice. For the L2 overlay the Virtual Networking Instance (VNI)/IID denotes a network-wide bridge domain, while for the L3 overlay the VNI/IID denotes a Virtual Routing Forwarding (VRF) instance.

Implementing an NVE with a unified L2 and L3 overlay support is beneficial for multiple reasons:

Primarily it allows the network administrator to choose which traffic traverses the L2 overlay versus the L3 overlay, not only on the basis of intra-subnet (bridged) versus inter-subnet (routed) traffic flows. As a matter of fact, it is highly beneficial to choose a policy where all IP traffic is forwarded across the L3 overlay (i.e. both intra- and inter-subnet traffic). Effectively this allows the 'spread' of subnets across the Datacenter(s) without leading to network wide broadcast and associated failure domains, while allowing free mobility of the end-stations.

Secondarily, when all the TS IP and MAC addresses are registered with the NVA/LISP Mapping system, optimisations in ARP and ND [RFC4861] forwarding and handling can be achieved. ARPs and IPv6 NDs for 'unknown' destinations are by default dropped, although a policy can allow for 'unknown' ARP/ND packets to be flooded across the underlay if so desired by the operator (e.g. when there is a desire to support 'silent hosts').

Finally, as all IP traffic is forwarded across a L3 overlay, and ARP/ND operations do not need flooding services, the amount of traffic that needs to traverse the L2 overlay is limited to non-IP traffic only. This makes the registration of MAC-addresses as EIDs with the LISP control plane optional. The system in this case could use ingress replication and Flood-and-Learn to handle the non-IP traffic. Of course, the use of the LISP control plane for MAC address based EIDs is another option as well, but the choice is left to the network administrator.

However, in order to achieve the benefits of this model, there are some requirements of how TSs can connect to the unified L2/L3 NVE, and there are also requirements on how default gateway MAC/IP addresses are assigned to the NVE function, and how forwarding is done on the NVE function:

- o The NVE MUST always do an IP lookup for IP based traffic, independent of whether the destination is within the same subnet or not, or whether the destination TS is attached to the same VLAN or L2 NVI as the source TS.
- o The unified L2/L3 NVE NVI instance MUST have a uniform default gateway MAC-address and IP address across the entire NVO3 network. This is to make sure that a TS can always reach its default gateway, irrespective of location.
- o A TS can connect across a L2 switched network to a unified L2/L3 NVE, but traffic forwarded MUST follow a simple rule, where all traffic from a TS MUST always be sent upstream to the unified L2/L3 NVE, regardless of its destination MAC address, and traffic

from locally attached TS's MUST be switched through the NVE. Directly connecting a TS to a unified L2/L3 NVE automatically solves that requirement.

There are various options to provide unified L2 and L3 support for LISP in the data path.

[I-D.smith-lisp-layer2] extends LISP to support MAC addresses as EIDs, and can be used in combination with [RFC6830], using the destination UDP port in the outer LISP header for disambiguation.

Recently extensions to both LISP and VxLAN have been proposed to offer multiprotocol support across the same outer header format (i.e. using a single UDP port number), as described in [I-D.lewis-lisp-gpe], and [I-D.quinn-vxlan-gpe] respectively. It is to be noted that some functionality offered by native LISP is no longer available when using the [I-D.lewis-lisp-gpe] extension (namely nonce, echo-nonce, and map-versioning). These are optional control plane optimizations implemented in the data plane for [RFC6830], and their use is less relevant in DC applications.

The remainder of this document assumes a unified L2/L3 NVE deployment model.

3.2.2. Multihoming aspects

If the TSes are co-located with the xTR/NVE function, no support for multi-homing is needed.

If the network between the L2 device connecting the TSes and the LISP xTRs is a simple hub and spoke switched L2 topology using VLANs (this is a common assumption in DC networks), a multi-chassis Link Aggregation Group (LAG) solution can be used to offer redundancy, where both xTRs will be seen by the access device as one logical entity. xTRs connected to the same L2 switched access network are part of the same 'LISP site', and both of them can be used to send traffic to TSes behind them, as both xTRs are registering to the LISP mapping system for the EIDs behind them. Registrations performed by the individual xTR (as a result of detection of a new EID) part of the same site are performed using the RLOCs of all xTRs connected to that site. How the multi-chassis LAG solution is build is out of scope of this draft.

3.2.3. External connectivity aspects

External connectivity between a LISP enabled NV03 DC, as well as any LISP site, and the external world can be handled through a gateway device.

In case the gateway device handles connectivity to VPNs or the Internet, LISP interworking will be employed at the gateway device as per [RFC6832].

In case the gateway device is used to interconnect to another DC part of the same administrative domain (one Mapping System governs both DCs), the only function needed on this device is routing within the RLOC address space.

In case separate LISP Mapping systems are used, interworking has to be established between them, as well as providing routing between the two administrative domain in between the RLOC address spaces of both DCs respectively. Today there is no described way of interworking between DDT based Mapping systems. An external controller could also insert new RLOC locations into a DDT based Mapping system when an EID has moved to a location not governed by this particular Mapping system.

3.2.4. Optimal Forwarding aspects

Implementing a co-located and unified L2 and L3 NVE, and placing that NVE as close as possible to the TSes, leads to the most optimal forwarding.

East-to-west traffic (from NVE to NVE) will always be mapped-and-encapped towards the 'right' NVE, as the NVA function (the LISP Mapping system) has knowledge about all of the TSes IP and MAC addresses.

North to South traffic (traffic ingress into the DC) will also be delivered to the right NVE, without traffic tromboning, as traffic is switched based on the EID IP address, which will always point to the correct ETR/RLOC.

Traffic going from TSes to external destinations will also not be affected by traffic tromboning as all NVE's part of an NVI are seen as the same default gateway, independent of location.

Traffic tromboning can occur if the last hop router is not in the same location as the egress NVE, and if only a single L2 NVE is deployed. In other words, the only way for a L2-only NVE based NV03 system to avoid traffic tromboning for north-south traffic, is by

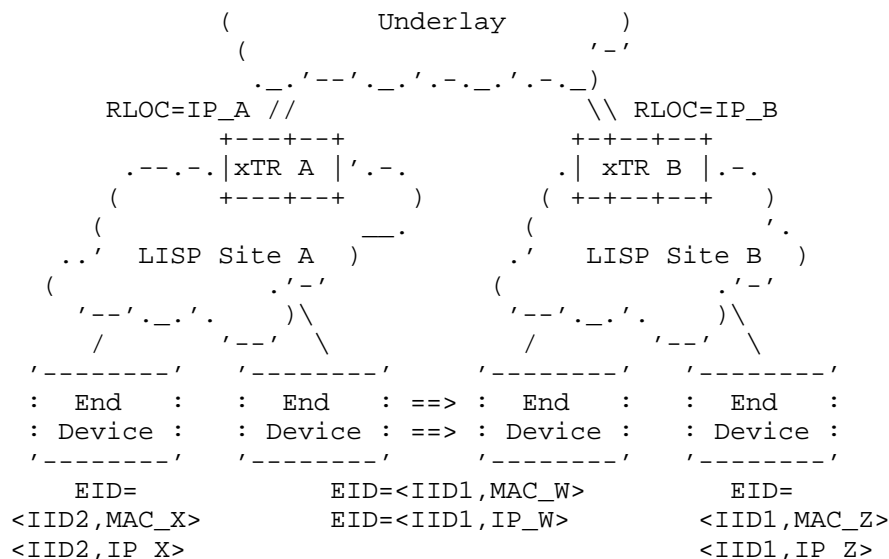


Figure 1: End System Mobility

It is assumed that the LISP xTRs have a unified L2 and L3 map-en-encap function, where IP packets, regardless of the fact they are switched intra- or inter subnet, are mapped-and-encapped across the L3 overlay. All other traffic (non-routable traffic, non-IP traffic) is mapped-and-encapped by the L2 overlay. It is also assumed that all xTRs offer the same default gateway IP and MAC address across the network, on a per VNI instance.

A unified L2/L3 overlay will lead in the xTRs registering two sets of EIDs for specific end systems, who deliver a mix of IP and non-IP traffic:

- o A tuple of EID=<IID, IP> to use for IP traffic across the L3 overlay, whereby the IID maps to a VRF instance. It will register the EID to multiple RLOCs, one being the xTR IP address, and the other one being the TS MAC Address.
- o A tuple EID= <IID,MAC> to use for non-routable, non-IP traffic, across the L2 overlay, whereby the IID maps to a network-wide Bridge Domain.

Assume the scenario described in Figure 1. Also assume that for the sake of this discussion, the VMs do not send out traffic that needs treatment by an L2 overlay.

As a result of the end system registration, the Mapping System contains the EID-to-RLOC mapping for end system W that associates EID=<IID1, IP_W> with the RLOC(s) associated with LISP site A (IP_A), as well as the RLOC associated with the MAC Address MAC_W of the end system W.

The process of migrating end system W from data center A to data center B is initiated.

ETR B receives a pre-associate message that includes EID=<IID1, IP_W>. ETR B sends a Map-Register to the mapping system registering RLOC=IP_B as an additional locator for end system W with priority set to 255. This means that the RLOC MUST NOT be used for unicast forwarding, but the mapping system is now aware of the new location.

During the migration process of end system W, ETR A receives a dissociate message, and sends a Map-Register with Record TTL=0 to signal the mapping system that end system W is no longer reachable at RLOC=IP_A. xTR A will also add an entry in its forwarding table that marks EID=<IID1, IP_W> as non-local.

When end system W has completed its migration, ETR B receives an associate message for end system W, and sends a Map-Register to the mapping system setting a non-255 priority for RLOC=IP_B. Now the mapping system is updated with the new EID-to-RLOC mapping for end system W with the desired priority.

The remote ITRs that were corresponding with end system W during the migration will keep sending packets to ETR A.

ETR A will keep forwarding locally those packets until it receives a dissociate message, and the entry in the forwarding table associated with EID=<IID1, IP_W> is marked as non-local.

Subsequent packets arriving at ETR A from a remote ITR, and destined to end system W will hit the entry in the forwarding table that will generate an exception, and will generate a Solicit-Map-Request (SMR) message that is returned to the remote ITR.

Upon receiving the SMR the remote ITR will invalidate its local map-cache entry for EID=<IID1, IP_W> and send a new Map-Request for that EID. The Map-Request will generate a Map-Reply that includes the new EID-to-RLOC mapping for end system W with RLOC=IP_B.

Similarly, unencapsulated packets arriving at ITR A from local end systems and destined to end system W will hit the entry in the forwarding table marked as non-local, and will generate an exception that by sending a Map-Request for EID=<IID1, IP_W> will populate the

map-cache of ITR A with an EID-to-RLOC mapping for end system W with RLOC=IP_B.

3.3. LISP dataplane options and NVO3 dataplane requirements

This section maps the NVO3 data plane requirements [I-D.ietf-nvo3-dataplane-requirements] to the various options available.

3.3.1. Native LISP Data Plane

Figure 2 shows the LISP header defined in the LISP specification [RFC6830]. The UDP and LISP headers are shown below for reference. For header fields description see section 5.3 of [RFC6830].

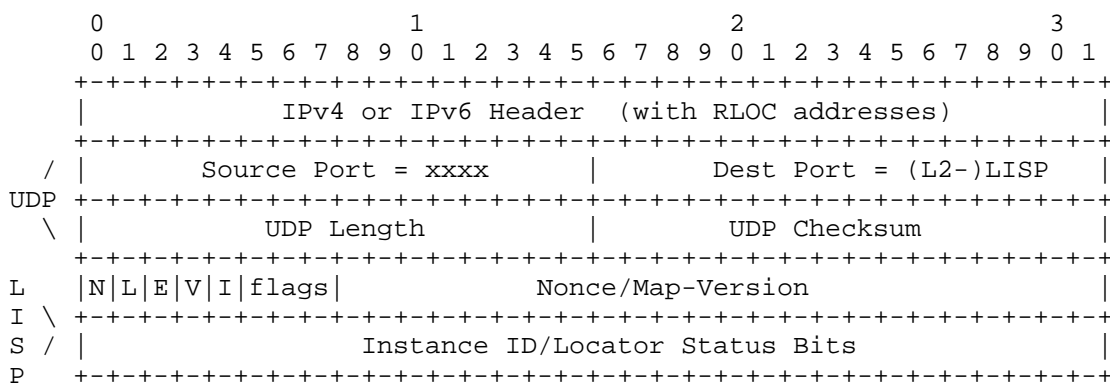


Figure 2: LISP Header

When the headers are used for encapsulating IP Packets, the UDP Destination Port is set to 4341. When the headers are used for encapsulating L2 frames, the UDP Destination Port is set to 8472 [I-D.smith-lisp-layer2].

When used in private DC and Enterprise networks, the 'I'-bit (Instance bit) is set, indicating the presence of an Instance ID (IID) inside the header. A Virtual Networking Instance (VNI) is indicated by the IID, a 24 bit field, which is used as a global identifier for the tenant in LISP. When used for L3 services, the IID can be seen as a VRF, when used for L2 services, the IID can be seen as a L2 Bridge Domain instance.

Virtual Access Point (VAP) identification is naturally supported by combining LISP and Integrated Routing and Bridging (IRB). IRB allows local ports or logical ports (ports instantiated on a local port, where frames are assigned based on some fields in the header like

VLAN IDs (VIDs)), to be added to a NVE-local bridge domain. That bridge domain instantiates the L2 Specific VNI. The bridge domain also connects to a virtual routed port, which instantiates the L3 specific VNI.

A L2 VNI provides an emulated Ethernet Multipoint service through the use of the LISP control plane, where it registers MAC addresses as EIDs.

Loop-avoidance is handled by control plane learning, and control plane enabled registration of all TS EIDs as soon as they send a first packet. Therefore unicast traffic will never result in loops, as there is no 'unknown' unicast. multi-destination traffic forwarding is performed using a multicast enabled underlay and LISP procedures laid out in [RFC6831] or through ingress replication to the list of participating NVEs in that NVI, and therefore is loop-free.

A L3 VNI behaves exactly as an IP VRF and therefore supports virtualized IP routing and forwarding, through per tenant forwarding with IP address isolation and L3 tunneling for interconnecting instances of the same VNI on NVEs.

Note that , within this document, it is assumed that a unified L2/L3 NVE is deployed and therefore all IP traffic will be forwarded using the L3 overlay, even intra-subnet traffic.

The LISP header performs the function of the NVO3 overlay header.

Through using the LISP control plane, the egress NVE can be looked up.

As the outer LISP header is an IPv4 or IPv6 header, differentiated forwarding can be supported naturally. Equally, as LISP uses IP/UDP as a transport, multipath over LAG and ECMP in the underlay are naturally supported, through the entropy introduced in the UDP header by selecting per flow source UDP port numbers. A LISP based NVO3 network can work in both uniform and pipe models [RFC2983] and fully supports ECN marking as per [RFC6040] .

As it does for L3 services, the LISP control plane replaces the use of dynamic data plane learning (Flood-and-Learn) for unicast traffic for L2 services. Packet replication in the underlay network to support L2 broadcast, unknown unicast (optional, as all MAC-address are learned by the control plane) and multicast L2 and L3 overlay services can be done by:

- o Ingress replication, which reduces the need for multicast in the NV03 underlay to zero.
- o Use of underlay multicast trees. These trees can be aggregated globally, or per tenant (rather than per VNI).

[RFC6831] and [I-D.farinacci-lisp-mr-signaling] specifies how to map a multicast flow in the EID space during distribution tree setup and packet delivery in the underlay network. LISP, being an IP based map-and-encap protocol, does not require any specific data plane functionality to make this work.

LISP interworking is described in [RFC6832] and fully supports connectivity to the internet or VPN gateways through the use of Proxy xTR's.

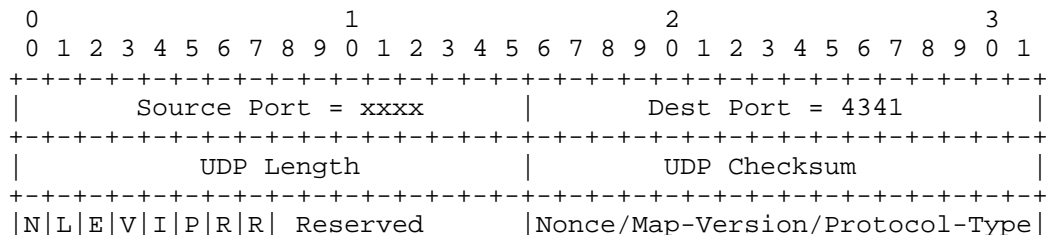
LISP, being an IP based protocol, supports ICMP-based MTU Path Discovery [RFC1191], [RFC1981] as well as extended MTU Path Discovery techniques [RFC4821]. LISP also supports a stateless and stateful way of dealing with Large Encapsulated packets, see section 5.4 of [RFC6830].

Multi-homing is handled in the control plane, by allowing the LISP mapping system to have multiple RLOC entries for every EID, allowing the ITR to load balance across both ETR's. xTRs register a 'LISP site id' to the mapping system when they come online. When the LISP mapping system receives a registration for a given EID from a certain xTRs, it will install that EID entry pointing to all the RLOCs/xTR that have the same site-id. By performing LAG across multiple xTRs, multi-homing can be provided for the access or virtual switch that connects the TSs.

OAM can be performed across LISP in the same way as OAM is performed over IP routed, or Ethernet L2 switched environments.

3.3.2. LISP with Generic Protocol Extension (LISP-GPE)

[I-D.lewis-lisp-gpe] introduces multiprotocol support for LISP, by extending the LISP header, as shown in Figure 3.



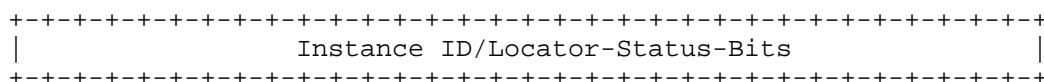


Figure 3: LISP with Generic Protocol Extension Header

A Protocol Bit (P-bit) is introduced, that when set, allows the insertion of a 16-bit Protocol Type. The UDP destination port number is 4341 for all protocol types.

Although the use of Nonce and Map-versioning are not allowed simultaneously with Protocol Type with this deployment, all of the solutions to the requirements described in Section 3.3.1 are exactly the same with this data plane encapsulation in an NV03 context.

3.3.3. VxLAN-GPE

[I-D.quinn-vxlan-gpe] extends the VXLAN encapsulation with a Protocol Type, by introducing a Protocol Bit (P-bit) and a 16-bit Protocol Type in the VXLAN header, see Figure 4. Note that this data plane encapsulation is very similar to LISP-GPE, when used in an NV03 context.

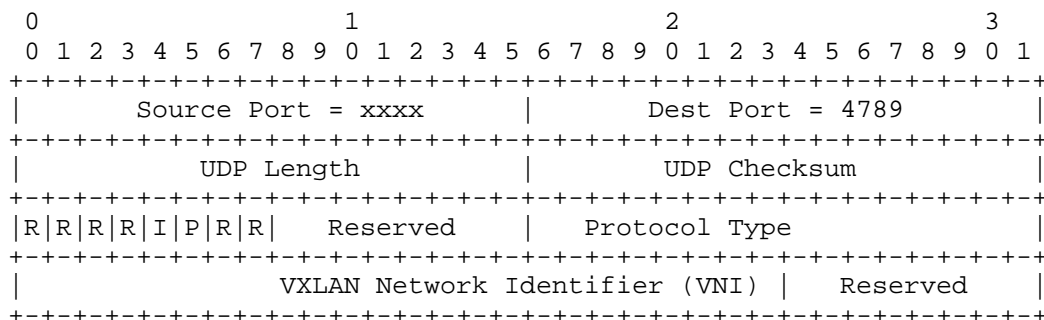


Figure 4: VXLAN with Generic Protocol Extension

All of the solutions to the requirements described in Section 3.3.1 are exactly the same with this data plane encapsulation.

3.3.4. L2 only solutions such as VxLAN and nvGRE

The LISP control plane can be leveraged to offer control plane learning for MAC Addresses for both the VXLAN [I-D.mahalingam-dutt-dcops-vxlan], as well as NVGRE [I-D.sridharan-virtualization-nvgre]. However, this solution offers sub-optimal support and hence will not be looked into further.

3.4. NVO3 control plane requirements and LISP

This section maps the NVO3 NVE to NVA control plane [I-D.ietf-nvo3-nve-nva-cp-req] requirements to the LISP control plane.

3.4.1. Inner to Outer Address Mapping

The LISP control plane, through the use of a Mapping service, provides inner to outer address mapping.

TS EIDs are registered to the LISP Mapping service by LISP ETRs within the context of a LISP instance ID, (i.e An NVO3 VNI).

A LISP based NVE will check its local cache if it needs to send a packet across the overlay. If there is a cache miss, it will request the EID to RLOC mapping from the LISP Mapping service. If there is a cache hit, it will use the local EID to RLOC mapping.

Cache entries are aged out when no traffic is being sent to those EIDs. The LISP control plane has ways of refreshing the local cache after the destination EID has moved to another RLOC. For more information, see Section 3.2.5 and [RFC6830]

3.4.2. Underlying network Multi-Destination Delivery

LISP supports delivering L2 and L3 multi-destination packets, independent of the underlying network multicast capabilities.

[RFC6831], [I-D.farinacci-lisp-mr-signaling] , more specifically section 6, describes how the LISP Control Plane is used by NVEs/ETRs to join a given EID multicast group by sending LISP Map-Requests rather than PIM Joins. Joining can be triggered by the receipt of a PIM or IGMP join in the EID space. In the case of a L2 overlay configuration on the NVE, the join is static.

In case the NVE/ETR is not multicast capable the ETR unicast RLOC will be registered, and will be added to the existing RLOC set for that given multicast EID, and the Map-Reply will contain the ITR from which the ETR wants to replicate. LISP ITRs will retrieve this list of ETRs from the Mapping System upon a Map-Request and will use this as a replication list.

In case the underlying network is multicast capable the Map-Reply will contain the multicast RLOC, which will be joined via PIM subsequently. All this state is stored on the Mapping system, or in the xTR caches per IID/VNI. In case ingress replication is deemed unscaleable, [I-D.farinacci-lisp-te] can be used, allowing a Re-encapsulating Tunnel Router (RTR) to be used as a distribution server to replicate to all the NVEs.

It is also important to point out that, in a unified L2/L3 NVE deployment, all IP traffic will get sent across the L3 overlay, and that ARP and ND packets are not handled using flooding.

In case non-IP traffic needs to be supported, L2 EIDs only need to use multicast or ingress replication for broadcast and multicast, as unicast flows are handled by the LISP control plane. This significantly reduces the multicast or ingress replication load.

3.4.3. VN connect/disconnect

We assume that a provisioning framework will be responsible for provisioning end systems (e.g. VMs) in each data center. The provisioning system configures each end system with an Ethernet/IEEE 802 MAC address and/or IP addresses and provisions the NVE with other end system specific attributes such as VLAN information, and TS/VLAN to VNI mapping information. LISP does not introduce new addressing requirements for end systems.

The provisioning infrastructure is also responsible to provide a network attach function, that notifies the NVE (the LISP site ETR) that the end system is attached to a given virtual network (identified by its VNI/IID) and that the end system is identified, within that virtual network, by a given Ethernet/IEEE 802 MAC address.

The LISP framework does not include mechanisms to provision the local NVE with the appropriate Tenant Instance for each Tenant Systems. Other protocols, such as VDP (in IEEE P802.1Qbg), should be used to implement a network attach/detach function, besides using link-up events for non-virtual end-systems. More-over it is quite common for devices to be able to 'sense' new tenant end-systems dynamically by tracking new mac addresses and IP addresses in case a VDP or link-up event cant be relied upon.

The LISP control plane can take advantage of such a network attach/detach function or the discovery of new MAC/IP addresses to trigger the registration of a Tenant System to the Mapping System. This is particularly helpful to handle mobility across the DC of the Tenant System.

Upon notification of end system network attach, the ETR sends a LISP Map-Register to the Mapping System. The Map-Register includes the EID and RLOCs of the LISP site. The EID-to-RLOC mapping is now available, via the Mapping System Infrastructure, to other LISP sites that are hosting end systems that belong to the same tenant.

For more details on end system registration see [RFC6833].

3.4.4. VN name to VN ID Mapping

The LISP Control Plane uses the Instance ID to identify the NVI. The VN Name to VNI mapping can be performed by the NVE as a result of local provisioning. Also, using LISP LCAF, it is possible to store ASCII Names in the Mapping Database, which can allow the system to resolve a VN Name to an IID/VNI.

3.4.5. LISP Control Plane Characteristics in an NVO3 context

LISP is a Control Plane solution that can scale very well to the NVO3 requirements:

1. LISP ETRs register destination EIDs into the LISP Mapping System. LISP ITRs pull destination EIDs from the LISP Mapping System and cache them for as long as traffic is being sent to those destinations. Hence a LISP Based NVE is only holding state for the active TS to TS flows, and only for the NVIs that are configured on those NVEs.
2. The LISP Control Plane is fast to acquire the needed state for a given destination through issuing a single Map-Request.
3. When an ETR (lets say ETR1) detects an EID it will also register this EID to the Mapping system. If that EID has moved from another ETR (lets say ETR2), it will update the Mapping system with a Map-Notify saying to no longer forward packets to it, and will install a 'non-local' entry in the forwarding table. If an ITR has not updated its map-cache, and therefor sends a packet to ETR2, ETR will sent a Map-Notify directly to the ITR, updating its local cache. For further information see Section 3.2.5
4. As LISP support virtualization, the NVE running the LISP Control Plane will only be maintaining state for the Tenants VNIs that are configured on it.

5. Through leveraging the LISP DDT-based Mapping system [I-D.ietf-lisp-ddt], the necessary scaling can be achieved. The LISP DDT hierarchy can be based on address family, address family prefix, and IID, and scales in a very similar way as DNS.
 6. The solution described in this document does not make use of multiple protocols, and hence is low in complexity.
 7. Through the use of the LISP LCAF [I-D.ietf-lisp-lcaf] , extensibility is achieved. It is possible to add new address families (the MAC address family is the proof point). The LCAF format also allows lookups on a generic Key. This Key can be an identifier to an ACL or policy. A combination of multiple keys can be achieved by doing recursive lookups, where EID attributes are used as keys for a subsequent lookup. LCAF allows backwards compatibility between systems that use different LCAF implementations.
 8. As the state is maintained in the LISP Mapping system acting as an NVA, adding another NVE/xTR to the network does not require any changes on existing NVEs.
 9. LISP does not rely on Multicast in the underlay, while preserving the same Control Plane characteristics regardless of underlay multicast capability.
 10. [I-D.barkai-lisp-nfv] documents one implementation of how the LISP Mapping System (NVA) can be programmed to create inner to outer address mappings. The LISP Control Plane will inform the xTR/NVE that hosts have moved, and if packets are attracted to those NVEs because of stale cache entries on other ITRs, packets will be routed to the right location, and the NVE will send a Solicited Map-Reply back to the ITR, clearing its cache, after which the ITR will request a new mapping. Obviously NVEs will be able to create inner to outer address mappings without the use of an orchestration solution.
 11. See Section 5
- 3.5. NV03 OAM Requirements and LISP

TBD

3.6. NVO3 Management Plane Requirements and LISP

TBD

3.7. Summary

The LISP Control Plane, makes a very good choice to implement NVO3 services due to the fact that it is agnostic to EID address families, and the fact that it provides an NVA in the form of the LISP Map Server with cache optimizations based on the pull-based LISP Map Cache on the LISP xTRs . The LISP control plane can be deployed across a set of different dataplane options as well. The usage of a unified L2 and L3 overlay , with the appropriate set of registrations in the LISP Mapping system, is recommended because of its optimal forwarding, scaling and IP centric characteristics, while supporting non-IP traffic as well.

4. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

[I-D.ietf-lisp-sec] defines a set of security mechanisms that provide origin authentication, integrity and anti-replay protection to LISP's EID-to-RLOC mapping data conveyed via mapping lookup process. LISP-SEC also enables verification of authorization on EID-prefix claims in Map-Reply messages.

Additional security mechanisms to protect the LISP Map-Register messages are defined in [RFC6833].

The security of the Mapping System Infrastructure depends on the particular mapping database used. The [I-D.ietf-lisp-ddt] specification, as an example, defines a public-key based mechanism that provides origin authentication and integrity protection to the LISP DDT protocol.

6. Acknowledgements

7. References

7.1. Normative References

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

7.2. Informative References

- [I-D.barkai-lisp-nfv]
sbarkai@gmail.com, s., Farinacci, D., Meyer, D., Maino, F., and V. Ermagan, "LISP Based FlowMapping for Scaling NFV", draft-barkai-lisp-nfv-02 (work in progress), July 2013.
- [I-D.farinacci-lisp-mr-signaling]
Farinacci, D. and M. Napierala, "LISP Control-Plane Multicast Signaling", draft-farinacci-lisp-mr-signaling-03 (work in progress), September 2013.
- [I-D.farinacci-lisp-te]
Farinacci, D., Lahiri, P., and M. Kowal, "LISP Traffic Engineering Use-Cases", draft-farinacci-lisp-te-03 (work in progress), July 2013.
- [I-D.ietf-lisp-ddt]
Fuller, V., Lewis, D., Ermagan, V., and A. Jain, "LISP Delegated Database Tree", draft-ietf-lisp-ddt-01 (work in progress), March 2013.
- [I-D.ietf-lisp-lcaf]
Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", draft-ietf-lisp-lcaf-03 (work in progress), September 2013.
- [I-D.ietf-lisp-sec]
Maino, F., Ermagan, V., Cabellos-Aparicio, A., Saucez, D., and O. Bonaventure, "LISP-Security (LISP-SEC)", draft-ietf-lisp-sec-04 (work in progress), October 2012.
- [I-D.ietf-nvo3-dataplane-requirements]
Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", draft-ietf-nvo3-dataplane-requirements-01 (work in progress), July 2013.
- [I-D.ietf-nvo3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-03 (work in progress), July 2013.

[I-D.ietf-nvo3-nve-nva-cp-req]

Kreeger, L., Dutt, D., Narten, T., and D. Black, "Network Virtualization NVE to NVA Control Protocol Requirements", draft-ietf-nvo3-nve-nva-cp-req-00 (work in progress), July 2013.

[I-D.ietf-nvo3-overlay-problem-statement]

Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-04 (work in progress), July 2013.

[I-D.kompella-nvo3-server2nve]

Kompella, K., Rekhter, Y., Morin, T., and D. Black, "Signaling Virtual Machine Activity to the Network Virtualization Edge", draft-kompella-nvo3-server2nve-02 (work in progress), April 2013.

[I-D.lewis-lisp-gpe]

Lewis, D., Agarwal, P., Kreeger, L., Quinn, P., Smith, M., and N. Yadav, "LISP Generic Protocol Extension", draft-lewis-lisp-gpe-01 (work in progress), October 2013.

[I-D.mahalingam-dutt-dcops-vxlan]

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-04 (work in progress), May 2013.

[I-D.maino-nvo3-lisp-cp]

Maino, F., Ermagan, V., Farinacci, D., and M. Smith, "LISP Control Plane for Network Virtualization Overlays", draft-maino-nvo3-lisp-cp-02 (work in progress), October 2012.

[I-D.quinn-vxlan-gpe]

Quinn, P., Agarwal, P., Fernando, R., Lewis, D., Kreeger, L., Smith, M., and N. Yadav, "Generic Protocol Extension for VXLAN", draft-quinn-vxlan-gpe-01 (work in progress), October 2013.

[I-D.smith-lisp-layer2]

Smith, M., Dutt, D., Farinacci, D., and F. Maino, "Layer 2 (L2) LISP Encapsulation Format", draft-smith-lisp-layer2-03 (work in progress), September 2013.

- [I-D.sridharan-virtualization-nvgre]
Sridharan, M., Greenberg, A., Wang, Y., Garg, P., Venkataramiah, N., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-03 (work in progress), August 2013.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.
- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.
- [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [RFC6831] Farinacci, D., Meyer, D., Zwiebel, J., and S. Venaas, "The Locator/ID Separation Protocol (LISP) for Multicast Environments", RFC 6831, January 2013.
- [RFC6832] Lewis, D., Meyer, D., Farinacci, D., and V. Fuller, "Interworking between Locator/ID Separation Protocol (LISP) and Non-LISP Sites", RFC 6832, January 2013.

- [RFC6833] Fuller, V. and D. Farinacci, "Locator/ID Separation Protocol (LISP) Map-Server Interface", RFC 6833, January 2013.
- [RFC6836] Fuller, V., Farinacci, D., Meyer, D., and D. Lewis, "Locator/ID Separation Protocol Alternative Logical Topology (LISP+ALT)", RFC 6836, January 2013.

Authors' Addresses

Yves Hertoghs
Cisco Systems
6a De Kleetlaan
Diegem 1831
Belgium

Phone: +32-2778-435
Fax: +32-2704-6000
Email: yves@cisco.com

Fabio Maino
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: fmaino@cisco.com

Victor Moreno
Cisco Systems
170 Tasman Drive
San Jose, California 95134
USA

Email: vimoreno@cisco.com

Michael Smith
Insieme Networks

Email: michsmith@insiemenetworks.com

Dino Farinacci
lispers.net

Email: farinacci@gmail.com

Internet Engineering Task Force
Internet Draft
Intended status: Informational
Expires: January 2014

Nabil Bitar
Verizon

Marc Lasserre
Florin Balus
Alcatel-Lucent

Thomas Morin
France Telecom Orange

Lizhong Jin

Bhumip Khasnabish
ZTE

July 1, 2013

NVO3 Data Plane Requirements
draft-ietf-nvo3-dataplane-requirements-01.txt

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 January 1, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract

Several IETF drafts relate to the use of overlay networks to support large scale virtual data centers. This draft provides a list of data plane requirements for Network Virtualization over L3 (NVO3) that have to be addressed in solutions documents.

Table of Contents

1. Introduction.....	3
1.1. Conventions used in this document.....	3
1.2. General terminology.....	3
2. Data Path Overview.....	4
3. Data Plane Requirements.....	5
3.1. Virtual Access Points (VAPs).....	5
3.2. Virtual Network Instance (VNI).....	5
3.2.1. L2 VNI.....	5
3.2.2. L3 VNI.....	6
3.3. Overlay Module.....	7
3.3.1. NVO3 overlay header.....	8
3.3.1.1. Virtual Network Context Identification.....	8
3.3.1.2. Service QoS identifier.....	8
3.3.2. Tunneling function.....	9
3.3.2.1. LAG and ECMP.....	10
3.3.2.2. DiffServ and ECN marking.....	10
3.3.2.3. Handling of BUM traffic.....	11
3.4. External NVO3 connectivity.....	11
3.4.1. GW Types.....	12
3.4.1.1. VPN and Internet GWs.....	12
3.4.1.2. Inter-DC GW.....	12
3.4.1.3. Intra-DC gateways.....	12

3.4.2. Path optimality between NVEs and Gateways.....	12
3.4.2.1. Triangular Routing Issues (Traffic Tromboning).....	13
3.5. Path MTU.....	14
3.6. Hierarchical NVE.....	15
3.7. NVE Multi-Homing Requirements.....	15
3.8. OAM.....	16
3.9. Other considerations.....	16
3.9.1. Data Plane Optimizations.....	16
3.9.2. NVE location trade-offs.....	17
4. Security Considerations.....	17
5. IANA Considerations.....	17
6. References.....	18
6.1. Normative References.....	18
6.2. Informative References.....	18
7. Acknowledgments.....	19

1. Introduction

1.1. Conventions 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 RFC-2119 [RFC2119].

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

1.2. General terminology

The terminology defined in [NVO3-framework] is used throughout this document. Terminology specific to this memo is defined here and is introduced as needed in later sections.

BUM: Broadcast, Unknown Unicast, Multicast traffic

TS: Tenant System

2. Data Path Overview

The NVO3 framework [NVO3-framework] defines the generic NVE model depicted in Figure 1:

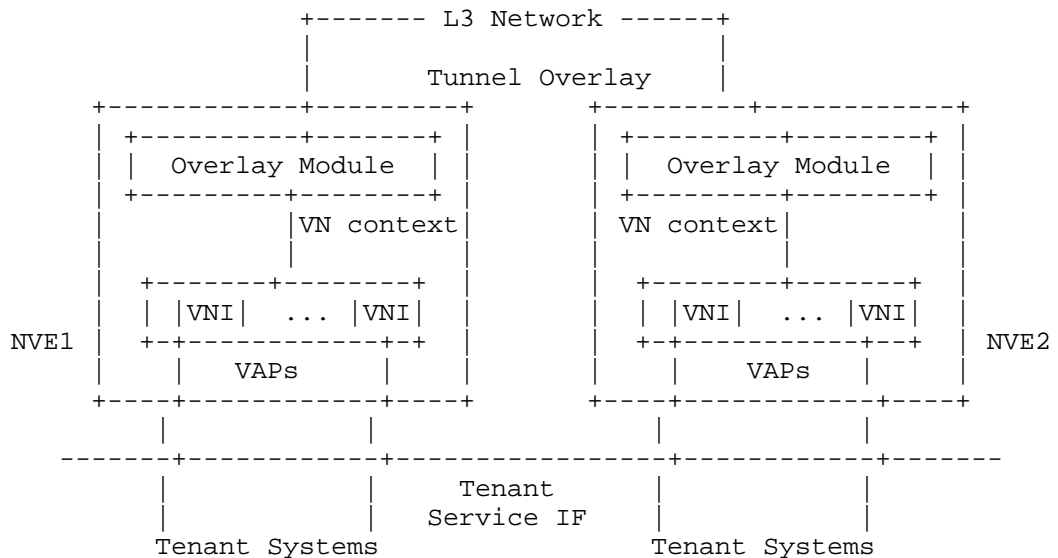


Figure 1 : Generic reference model for NV Edge

When a frame is received by an ingress NVE from a Tenant System over a local VAP, it needs to be parsed in order to identify which virtual network instance it belongs to. The parsing function can examine various fields in the data frame (e.g., VLANID) and/or associated interface/port the frame came from.

Once a corresponding VNI is identified, a lookup is performed to determine where the frame needs to be sent. This lookup can be based on any combinations of various fields in the data frame (e.g., destination MAC addresses and/or destination IP addresses). Note that additional criteria such as 802.1p and/or DSCP markings might be used to select an appropriate tunnel or local VAP destination.

Lookup tables can be populated using different techniques: data plane learning, management plane configuration, or a distributed control plane. Management and control planes are not in the scope of

this document. The data plane based solution is described in this document as it has implications on the data plane processing function.

The result of this lookup yields the corresponding information needed to build the overlay header, as described in section 3.3. This information includes the destination L3 address of the egress NVE. Note that this lookup might yield a list of tunnels such as when ingress replication is used for BUM traffic.

The overlay header MUST include a context identifier which the egress NVE will use to identify which VNI this frame belongs to.

The egress NVE checks the context identifier and removes the encapsulation header and then forwards the original frame towards the appropriate recipient, usually a local VAP.

3. Data Plane Requirements

3.1. Virtual Access Points (VAPs)

The NVE forwarding plane MUST support VAP identification through the following mechanisms:

- Using the local interface on which the frames are received, where the local interface may be an internal, virtual port in a VSwitch or a physical port on the ToR
- Using the local interface and some fields in the frame header, e.g. one or multiple VLANs or the source MAC

3.2. Virtual Network Instance (VNI)

VAPs are associated with a specific VNI at service instantiation time.

A VNI identifies a per-tenant private context, i.e. per-tenant policies and a FIB table to allow overlapping address space between tenants.

There are different VNI types differentiated by the virtual network service they provide to Tenant Systems. Network virtualization can be provided by L2 and/or L3 VNIs.

3.2.1. L2 VNI

An L2 VNI MUST provide an emulated Ethernet multipoint service as if Tenant Systems are interconnected by a bridge (but instead by using

a set of NVO3 tunnels). The emulated bridge MAY be 802.1Q enabled (allowing use of VLAN tags as a VAP). An L2 VNI provides per tenant virtual switching instance with MAC addressing isolation and L3 tunneling. Loop avoidance capability MUST be provided.

Forwarding table entries provide mapping information between tenant system MAC addresses and VAPs on directly connected VNIs and L3 tunnel destination addresses over the overlay. Such entries MAY be populated by a control or management plane, or via data plane.

In the absence of a management or control plane, data plane learning MUST be used to populate forwarding tables. As frames arrive from VAPs or from overlay tunnels, standard MAC learning procedures are used: The tenant system source MAC address is learned against the VAP or the NVO3 tunneling encapsulation source address on which the frame arrived. This implies that unknown unicast traffic be flooded i.e. broadcast.

When flooding is required, either to deliver unknown unicast, or broadcast or multicast traffic, the NVE MUST either support ingress replication or multicast. In this latter case, the NVE MUST have one or more multicast trees that can be used by local VNIs for flooding to NVEs belonging to the same VN. For each VNI, there is one flooding tree, and a multicast tree may be dedicated per VNI or shared across VNIs. In such cases, multiple VNIs MAY share the same default flooding tree. The flooding tree is equivalent with a multicast (*,G) construct where all the NVEs for which the corresponding VNI is instantiated are members. The multicast tree MAY be established automatically via routing and signaling or pre-provisioned.

When tenant multicast is supported, it SHOULD also be possible to select whether the NVE provides optimized multicast trees inside the VNI for individual tenant multicast groups or whether the default VNI flooding tree is used. If the former option is selected the VNI SHOULD be able to snoop IGMP/MLD messages in order to efficiently join/prune Tenant System from multicast trees.

3.2.2. L3 VNI

L3 VNIs MUST provide virtualized IP routing and forwarding. L3 VNIs MUST support per-tenant forwarding instance with IP addressing isolation and L3 tunneling for interconnecting instances of the same VNI on NVEs.

In the case of L3 VNI, the inner TTL field MUST be decremented by (at least) 1 as if the NVO3 egress NVE was one (or more) hop(s)

away. The TTL field in the outer IP header MUST be set to a value appropriate for delivery of the encapsulated frame to the tunnel exit point. Thus, the default behavior MUST be the TTL pipe model where the overlay network looks like one hop to the sending NVE. Configuration of a "uniform" TTL model where the outer tunnel TTL is set equal to the inner TTL on ingress NVE and the inner TTL is set to the outer TTL value on egress MAY be supported.

L2 and L3 VNIs can be deployed in isolation or in combination to optimize traffic flows per tenant across the overlay network. For example, an L2 VNI may be configured across a number of NVEs to offer L2 multi-point service connectivity while a L3 VNI can be co-located to offer local routing capabilities and gateway functionality. In addition, integrated routing and bridging per tenant MAY be supported on an NVE. An instantiation of such service may be realized by interconnecting an L2 VNI as access to an L3 VNI on the NVE.

The L3 VNI does not require support for Broadcast and Unknown Unicast traffic. The L3 VNI MAY provide support for customer multicast groups. When multicast is supported, it SHOULD be possible to select whether the NVE provides optimized multicast trees inside the VNI for individual tenant multicast groups or whether a default VNI multicasting tree, where all the NVEs of the corresponding VNI are members, is used.

3.3. Overlay Module

The overlay module performs a number of functions related to NVO3 header and tunnel processing.

The following figure shows a generic NVO3 encapsulated frame:

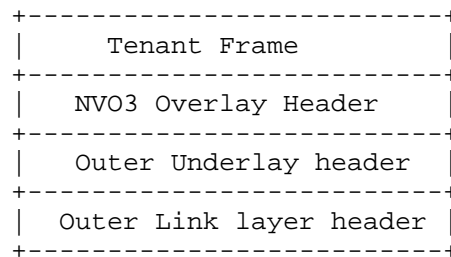


Figure 2 : NVO3 encapsulated frame

where

- . Tenant frame: Ethernet or IP based upon the VNI type
- . NVO3 overlay header: Header containing VNI context information and other optional fields that can be used for processing this packet.
- . Outer underlay header: Can be either IP or MPLS
- . Outer link layer header: Header specific to the physical transmission link used

3.3.1. NVO3 overlay header

An NVO3 overlay header **MUST** be included after the underlay tunnel header when forwarding tenant traffic. Note that this information can be carried within existing protocol headers (when overloading of specific fields is possible) or within a separate header.

3.3.1.1. Virtual Network Context Identification

The overlay encapsulation header **MUST** contain a field which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE. The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field **MAY** be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or **MAY** express the necessary context information in other ways (e.g. a locally significant identifier).

It **SHOULD** be aligned on a 32-bit boundary so as to make it efficiently processable by the data path. It **MUST** be distributable by a control-plane or configured via a management plane.

In the case of a global identifier, this field **MUST** be large enough to scale to 100's of thousands of virtual networks. Note that there is no such constraint when using a local identifier.

3.3.1.2. Service QoS identifier

Traffic flows originating from different applications could rely on differentiated forwarding treatment to meet end-to-end availability and performance objectives. Such applications may span across one or more overlay networks. To enable such treatment, support for

multiple Classes of Service across or between overlay networks MAY be required.

To effectively enforce CoS across or between overlay networks, NVEs MAY be able to map CoS markings between networking layers, e.g., Tenant Systems, Overlays, and/or Underlay, enabling each networking layer to independently enforce its own CoS policies. For example:

- TS (e.g. VM) CoS
 - o Tenant CoS policies MAY be defined by Tenant administrators
 - o QoS fields (e.g. IP DSCP and/or Ethernet 802.1p) in the tenant frame are used to indicate application level CoS requirements
- NVE CoS
 - o NVE MAY classify packets based on Tenant CoS markings or other mechanisms (eg. DPI) to identify the proper service CoS to be applied across the overlay network
 - o NVE service CoS levels are normalized to a common set (for example 8 levels) across multiple tenants; NVE uses per tenant policies to map Tenant CoS to the normalized service CoS fields in the NVO3 header
- Underlay CoS
 - o The underlay/core network MAY use a different CoS set (for example 4 levels) than the NVE CoS as the core devices MAY have different QoS capabilities compared with NVEs.
 - o The Underlay CoS MAY also change as the NVO3 tunnels pass between different domains.

Support for NVE Service CoS MAY be provided through a QoS field, inside the NVO3 overlay header. Examples of service CoS provided part of the service tag are 802.1p and DE bits in the VLAN and PBB ISID tags and MPLS TC bits in the VPN labels.

3.3.2. Tunneling function

This section describes the underlay tunneling requirements. From an encapsulation perspective, IPv4 or IPv6 MUST be supported, both IPv4 and IPv6 SHOULD be supported, MPLS tunneling MAY be supported.

3.3.2.1. LAG and ECMP

For performance reasons, multipath over LAG and ECMP paths SHOULD be supported.

LAG (Link Aggregation Group) [IEEE 802.1AX-2008] and ECMP (Equal Cost Multi Path) are commonly used techniques to perform load-balancing of microflows over a set of a parallel links either at Layer-2 (LAG) or Layer-3 (ECMP). Existing deployed hardware implementations of LAG and ECMP uses a hash of various fields in the encapsulation (outermost) header(s) (e.g. source and destination MAC addresses for non-IP traffic, source and destination IP addresses, L4 protocol, L4 source and destination port numbers, etc). Furthermore, hardware deployed for the underlay network(s) will be most often unaware of the carried, innermost L2 frames or L3 packets transmitted by the TS. Thus, in order to perform fine-grained load-balancing over LAG and ECMP paths in the underlying network, the encapsulation MUST result in sufficient entropy to exercise all paths through several LAG/ECMP hops. The entropy information MAY be inferred from the NVO3 overlay header or underlay header. If the overlay protocol does not support the necessary entropy information or the switches/routers in the underlay do not support parsing of the additional entropy information in the overlay header, underlay switches and routers should be programmable, i.e. select the appropriate fields in the underlay header for hash calculation based on the type of overlay header.

All packets that belong to a specific flow MUST follow the same path in order to prevent packet re-ordering. This is typically achieved by ensuring that the fields used for hashing are identical for a given flow.

All paths available to the overlay network SHOULD be used efficiently. Different flows SHOULD be distributed as evenly as possible across multiple underlay network paths. For instance, this can be achieved by ensuring that some fields used for hashing are randomly generated.

3.3.2.2. DiffServ and ECN marking

When traffic is encapsulated in a tunnel header, there are numerous options as to how the Diffserv Code-Point (DSCP) and Explicit Congestion Notification (ECN) markings are set in the outer header and propagated to the inner header on decapsulation.

[RFC2983] defines two modes for mapping the DSCP markings from inner to outer headers and vice versa. The Uniform model copies the inner

DSCP marking to the outer header on tunnel ingress, and copies that outer header value back to the inner header at tunnel egress. The Pipe model sets the DSCP value to some value based on local policy at ingress and does not modify the inner header on egress. Both models SHOULD be supported.

ECN marking MUST be performed according to [RFC6040] which describes the correct ECN behavior for IP tunnels.

3.3.2.3. Handling of BUM traffic

NVO3 data plane support for either ingress replication or point-to-multipoint tunnels is required to send traffic destined to multiple locations on a per-VNI basis (e.g. L2/L3 multicast traffic, L2 broadcast and unknown unicast traffic). It is possible that both methods be used simultaneously.

There is a bandwidth vs state trade-off between the two approaches. User-definable knobs MUST be provided to select which method(s) gets used based upon the amount of replication required (i.e. the number of hosts per group), the amount of multicast state to maintain, the duration of multicast flows and the scalability of multicast protocols.

When ingress replication is used, NVEs MUST track for each VNI the related tunnel endpoints to which it needs to replicate the frame.

For point-to-multipoint tunnels, the bandwidth efficiency is increased at the cost of more state in the Core nodes. The ability to auto-discover or pre-provision the mapping between VNI multicast trees to related tunnel endpoints at the NVE and/or throughout the core SHOULD be supported.

3.4. External NVO3 connectivity

NVO3 services MUST interoperate with current VPN and Internet services. This may happen inside one DC during a migration phase or as NVO3 services are delivered to the outside world via Internet or VPN gateways.

Moreover the compute and storage services delivered by a NVO3 domain may span multiple DCs requiring Inter-DC connectivity. From a DC perspective a set of gateway devices are required in all of these cases albeit with different functionalities influenced by the overlay type across the WAN, the service type and the DC network technologies used at each DC site.

A GW handling the connectivity between NVO3 and external domains represents a single point of failure that may affect multiple tenant services. Redundancy between NVO3 and external domains MUST be supported.

3.4.1. GW Types

3.4.1.1. VPN and Internet GWs

Tenant sites may be already interconnected using one of the existing VPN services and technologies (VPLS or IP VPN). If a new NVO3 encapsulation is used, a VPN GW is required to forward traffic between NVO3 and VPN domains. Translation of encapsulations MAY be required. Internet connected Tenants require translation from NVO3 encapsulation to IP in the NVO3 gateway. The translation function SHOULD minimize provisioning touches.

3.4.1.2. Inter-DC GW

Inter-DC connectivity MAY be required to provide support for features like disaster prevention or compute load re-distribution. This MAY be provided via a set of gateways interconnected through a WAN. This type of connectivity MAY be provided either through extension of the NVO3 tunneling domain or via VPN GWs.

3.4.1.3. Intra-DC gateways

Even within one DC there may be End Devices that do not support NVO3 encapsulation, for example bare metal servers, hardware appliances and storage. A gateway device, e.g. a ToR, is required to translate the NVO3 to Ethernet VLAN encapsulation.

3.4.2. Path optimality between NVEs and Gateways

Within the NVO3 overlay, a default assumption is that NVO3 traffic will be equally load-balanced across the underlying network consisting of LAG and/or ECMP paths. This assumption is valid only as long as: a) all traffic is load-balanced equally among each of the component-links and paths; and, b) each of the component-links/paths is of identical capacity. During the course of normal operation of the underlying network, it is possible that one, or more, of the component-links/paths of a LAG may be taken out-of-service in order to be repaired, e.g.: due to hardware failure of cabling, optics, etc. In such cases, the administrator should configure the underlying network such that an entire LAG bundle in the underlying network will be reported as operationally down if there is a failure of any single component-link member of the LAG

bundle, (e.g.: N = M configuration of the LAG bundle), and, thus, they know that traffic will be carried sufficiently by alternate, available (potentially ECMP) paths in the underlying network. This is a likely an adequate assumption for Intra-DC traffic where presumably the costs for additional, protection capacity along alternate paths is not cost-prohibitive. Thus, there are likely no additional requirements on NVO3 solutions to accommodate this type of underlying network configuration and administration.

There is a similar case with ECMP, used Intra-DC, where failure of a single component-path of an ECMP group would result in traffic shifting onto the surviving members of the ECMP group. Unfortunately, there are no automatic recovery methods in IP routing protocols to detect a simultaneous failure of more than one component-path in a ECMP group, operationally disable the entire ECMP group and allow traffic to shift onto alternative paths. This problem is attributable to the underlying network and, thus, out-of-scope of any NVO3 solutions.

On the other hand, for Inter-DC and DC to External Network cases that use a WAN, the costs of the underlying network and/or service (e.g.: IPVPN service) are more expensive; therefore, there is a requirement on administrators to both: a) ensure high availability (active-backup failover or active-active load-balancing); and, b) maintaining substantial utilization of the WAN transport capacity at nearly all times, particularly in the case of active-active load-balancing. With respect to the dataplane requirements of NVO3 solutions, in the case of active-backup fail-over, all of the ingress NVE's MUST dynamically adapt to the failure of an active NVE GW when the backup NVE GW announces itself into the NVO3 overlay immediately following a failure of the previously active NVE GW and update their forwarding tables accordingly, (e.g.: perhaps through dataplane learning and/or translation of a gratuitous ARP, IPv6 Router Advertisement, etc.) Note that active-backup fail-over could be used to accomplish a crude form of load-balancing by, for example, manually configuring each tenant to use a different NVE GW, in a round-robin fashion. On the other hand, with respect to active-active load-balancing across physically separate NVE GW's (e.g.: two, separate chassis) an NVO3 solution SHOULD support forwarding tables that can simultaneously map a single egress NVE to more than one NVO3 tunnels. The granularity of such mappings, in both active-backup and active-active, MUST be unique to each tenant.

3.4.2.1. Triangular Routing Issues (Traffic Tromboning)

L2/ELAN over NVO3 service may span multiple racks distributed across different DC regions. Multiple ELANs belonging to one tenant may be

interconnected or connected to the outside world through multiple Router/VRF gateways distributed throughout the DC regions. In this scenario, without aid from an NVO3 or other type of solution, traffic from an ingress NVE destined to External gateways will take a non-optimal path that will result in higher latency and costs, (since it is using more expensive resources of a WAN). In the case of traffic from an IP/MPLS network destined toward the entrance to an NVO3 overlay, well-known IP routing techniques MAY be used to optimize traffic into the NVO3 overlay, (at the expense of additional routes in the IP/MPLS network). In summary, these issues are well known as triangular routing.

Procedures for gateway selection to avoid triangular routing issues SHOULD be provided. The details of such procedures are, most likely, part of the NVO3 Management and/or Control Plane requirements and, thus, out of scope of this document. However, a key requirement on the dataplane of any NVO3 solution to avoid triangular routing is stated above, in Section 3.4.2, with respect to active-active load-balancing. More specifically, an NVO3 solution SHOULD support forwarding tables that can simultaneously map a single egress NVE to more than one NVO3 tunnels. The expectation is that, through the Control and/or Management Planes, this mapping information MAY be dynamically manipulated to, for example, provide the closest geographic and/or topological exit point (egress NVE) for each ingress NVE.

3.5. Path MTU

The tunnel overlay header can cause the MTU of the path to the egress tunnel endpoint to be exceeded.

IP fragmentation SHOULD be avoided for performance reasons.

The interface MTU as seen by a Tenant System SHOULD be adjusted such that no fragmentation is needed. This can be achieved by configuration or be discovered dynamically.

Either of the following options MUST be supported:

- o Classical ICMP-based MTU Path Discovery [RFC1191] [RFC1981] or Extended MTU Path Discovery techniques such as defined in [RFC4821]
- o Segmentation and reassembly support from the overlay layer operations without relying on the Tenant Systems to know about the end-to-end MTU

- o The underlay network MAY be designed in such a way that the MTU can accommodate the extra tunnel overhead.

3.6. Hierarchical NVE

It might be desirable to support the concept of hierarchical NVEs, such as spoke NVEs and hub NVEs, in order to address possible NVE performance limitations and service connectivity optimizations.

For instance, spoke NVE functionality MAY be used when processing capabilities are limited. A hub NVE would provide additional data processing capabilities such as packet replication.

NVEs can be either connected in an any-to-any or hub and spoke topology on a per VNI basis.

3.7. NVE Multi-Homing Requirements

Multi-homing techniques SHOULD be used to increase the reliability of an nvo3 network. It is also important to ensure that physical diversity in an nvo3 network is taken into account to avoid single points of failure.

Multi-homing can be enabled in various nodes, from tenant systems into TORs, TORs into core switches/routers, and core nodes into DC GWs.

Tenant systems can either be L2 or L3 nodes. In the former case (L2), techniques such as LAG or STP for instance MAY be used. In the latter case (L3), it is possible that no dynamic routing protocol is enabled. Tenant systems can be multi-homed into remote NVE using several interfaces (physical NICS or vNICS) with an IP address per interface either to the same nvo3 network or into different nvo3 networks. When one of the links fails, the corresponding IP is not reachable but the other interfaces can still be used. When a tenant system is co-located with an NVE, IP routing can be relied upon to handle routing over diverse links to TORs.

External connectivity MAY be handled by two or more nvo3 gateways. Each gateway is connected to a different domain (e.g. ISP) and runs BGP multi-homing. They serve as an access point to external networks such as VPNs or the Internet. When a connection to an upstream router is lost, the alternative connection is used and the failed route withdrawn.

3.8. OAM

NVE MAY be able to originate/terminate OAM messages for connectivity verification, performance monitoring, statistic gathering and fault isolation. Depending on configuration, NVEs SHOULD be able to process or transparently tunnel OAM messages, as well as supporting alarm propagation capabilities.

Given the critical requirement to load-balance NVO3 encapsulated packets over LAG and ECMP paths, it will be equally critical to ensure existing and/or new OAM tools allow NVE administrators to proactively and/or reactively monitor the health of various component-links that comprise both LAG and ECMP paths carrying NVO3 encapsulated packets. For example, it will be important that such OAM tools allow NVE administrators to reveal the set of underlying network hops (topology) in order that the underlying network administrators can use this information to quickly perform fault isolation and restore the underlying network.

The NVE MUST provide the ability to reveal the set of ECMP and/or LAG paths used by NVO3 encapsulated packets in the underlying network from an ingress NVE to egress NVE. The NVE MUST provide the ability to provide a "ping"-like functionality that can be used to determine the health (liveness) of remote NVE's or their VNI's. The NVE SHOULD provide a "ping"-like functionality to more expeditiously aid in troubleshooting performance problems, i.e.: blackholing or other types of congestion occurring in the underlying network, for NVO3 encapsulated packets carried over LAG and/or ECMP paths.

3.9. Other considerations

3.9.1. Data Plane Optimizations

Data plane forwarding and encapsulation choices SHOULD consider the limitation of possible NVE implementations, specifically in software based implementations (e.g. servers running VSwitches)

NVE SHOULD provide efficient processing of traffic. For instance, packet alignment, the use of offsets to minimize header parsing, padding techniques SHOULD be considered when designing NVO3 encapsulation types.

The NVO3 encapsulation/decapsulation processing in software-based NVEs SHOULD make use of hardware assist provided by NICs in order to speed up packet processing.

3.9.2. NVE location trade-offs

In the case of DC traffic, traffic originated from a VM is native Ethernet traffic. This traffic can be switched by a local VM switch or ToR switch and then by a DC gateway. The NVE function can be embedded within any of these elements.

The NVE function can be supported in various DC network elements such as a VM, VM switch, ToR switch or DC GW.

The following criteria SHOULD be considered when deciding where the NVE processing boundary happens:

- o Processing and memory requirements
 - o Datapath (e.g. lookups, filtering, encapsulation/decapsulation)
 - o Control plane processing (e.g. routing, signaling, OAM)
- o FIB/RIB size
- o Multicast support
 - o Routing protocols
 - o Packet replication capability
- o Fragmentation support
- o QoS transparency
- o Resiliency

4. Security Considerations

This requirements document does not raise in itself any specific security issues.

5. IANA Considerations

IANA does not need to take any action for this draft.

6. References

6.1. Normative References

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

6.2. Informative References

- [NVOPS] Narten, T. et al, "Problem Statement: Overlays for Network Virtualization", draft-narten-nvo3-overlay-problem-statement (work in progress)
- [NVO3-framework] Lasserre, M. et al, "Framework for DC Network Virtualization", draft-lasserre-nvo3-framework (work in progress)
- [OVCPREQ] Kreeger, L. et al, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp (work in progress)
- [FLOYD] Sally Floyd, Allyn Romanow, "Dynamics of TCP Traffic over ATM Networks", IEEE JSAC, V. 13 N. 4, May 1995
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC1191] Mogul, J. "Path MTU Discovery", RFC1191, November 1990
- [RFC1981] McCann, J. et al, "Path MTU Discovery for IPv6", RFC1981, August 1996
- [RFC4821] Mathis, M. et al, "Packetization Layer Path MTU Discovery", RFC4821, March 2007
- [RFC2983] Black, D. "Diffserv and tunnels", RFC2983, October 2000
- [RFC6040] Briscoe, B. "Tunnelling of Explicit Congestion Notification", RFC6040, November 2010
- [RFC6438] Carpenter, B. et al, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC6438, November 2011
- [RFC6391] Bryant, S. et al, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC6391, November 2011

7. Acknowledgments

In addition to the authors the following people have contributed to this document:

Shane Amante, Level3

Dimitrios Stiliadis, Rotem Salomonovitch, Alcatel-Lucent

Larry Kreeger, Cisco

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
Email: nabil.bitar@verizon.com

Marc Lasserre
Alcatel-Lucent
Email: marc.lasserre@alcatel-lucent.com

Florin Balus
Alcatel-Lucent
777 E. Middlefield Road
Mountain View, CA, USA 94043
Email: florin.balus@alcatel-lucent.com

Thomas Morin
France Telecom Orange
Email: thomas.morin@orange.com

Lizhong Jin
Email : lizho.jin@gmail.com

Bhumip Khasnabish
ZTE
Email : Bhumip.khasnabish@zteusa.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: March 29, 2014

E. Gray, Ed.
Ericsson
N. Bitar
Verizon
X. Chen
Huawei Technologies
M. Lasserre
Alcatel-Lucent
T. Tsou
Huawei Technologies (USA)
September 25, 2013

NVO3 Gap Analysis - Requirements Versus Available Technology Choices
draft-ietf-nvo3-gap-analysis-00

Abstract

This document evaluates candidate protocols against the NVO3 requirements. Gaps are identified and further work recommended.

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 March 29, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Terminology and Conventions	3
2.1. Requirements Language	3
2.2. Conventions	3
2.3. Terms and Abbreviations	3
3. Operational Requirements	4
4. Management Requirements	4
5. Control Plane Requirements	4
5.1. Overall Control-Plane Requirements	5
5.2. VM-to-NVE Specific Control-Plane Requirements	7
6. Data Plane Requirements	9
7. Summary and Conclusions	14
8. Acknowledgements	14
9. IANA Considerations	15
10. Security Considerations	15
11. References	15
11.1. Normative References	15
11.2. Informative References	17
Authors' Addresses	17

1. Introduction

The initial charter of the NVO3 Working Group requires it to identify any gaps between the requirements identified and available technology solutions as a prerequisite to rechartering or concluding the Working Group (if no gaps exist). This document is intended to provide the required gap analysis.

This document provides a tabulation of candidate solutions and their ability to satisfy each requirement identified by the Working Group.

Areas of work are identified where further work is required to ensure that the requirements are met.

The major areas covered in this document include:

- o Operational Requirements
[I-D.ashwood-nvo3-operational-requirement]
- o Management Requirements (TBD)

- o Control (Plane) Requirements [I-D.kreeger-nvo3-overlay-cp]
- o Dataplane Requirements [I-D.ietf-nvo3-dataplane-requirements]

Since the Working Group has yet to complete (and in some cases adopt) documents describing requirements for some of these areas, not all areas are complete in the present version of this document.

The initial candidate technologies are:

- o NVGRE [I-D.sridharan-virtualization-nvgre],
- o VxLAN [I-D.mahalingam-dutt-dcops-vxlan],
- o L2VPN: VPLS [RFC4761][RFC4762] and EVPN [I-D.ietf-l2vpn-evpn], and
- o L3VPN [RFC4365].

2. Terminology and Conventions

2.1. Requirements Language

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

2.2. Conventions

In sections providing analysis of requirements defined in referenced documents, section numbers from each referenced document are used as they were listed in that document.

In order to avoid confusing those section numbers with the section numbering in this document, the included numbering is parenthesized.

L2VPN is represented (in tables and analysis, as a technology) by the two differing approaches: VPLS and EVPN.

2.3. Terms and Abbreviations

This document uses terms and acronyms defined in [RFC3168], [I-D.ietf-nvo3-framework], [I-D.ietf-nvo3-dataplane-requirements], [I-D.kreeger-nvo3-hypervisor-nve-cp] and [I-D.kreeger-nvo3-overlay-cp]. Acronyms are included here for convenience but are meant to remain aligned with definitions in the references included.

ECN: Explicit Congestion Notification [RFC3168]

NVA: Network Virtualization Authority [I-D.kreeger-nvo3-overlay-cp]

NVE: Network Virtualization Edge [I-D.ietf-nvo3-framework]

VAP: Virtual Access Point [I-D.ietf-nvo3-dataplane-requirements]

VNI: Virtual Network Instance [I-D.ietf-nvo3-framework]

VNIC: Virtual Network Interface Card (NIC)
[I-D.kreeger-nvo3-hypervisor-nve-cp]

VNID: Virtual Network Identifier [I-D.kreeger-nvo3-overlay-cp]

This document uses the following additional general terms and abbreviations:

DSCP: Differentiated Services Code-Point

ECMP: Equal Cost Multi-Path

L2VPN: Layer 2 Virtual Private Network

L3VPN: Layer 3 Virtual Private Network

NVO3: Network Virtualization Overlay over L3

VM: Virtual Machine

VN: Virtual Network

3. Operational Requirements

TBD

4. Management Requirements

TBD

5. Control Plane Requirements

The NVO3 Problem Statement [I-D.ietf-nvo3-overlay-problem-statement], describes 3 categories of control functions:

1. Control functions associated with implementing the Network Virtualization Authority (e.g. - signaling and control required for interactions between multiple NVA devices).

2. Control functions associated with interactions between an NVA and a Network Virtualization Edge (NVE).
3. Control functions associated with attaching and detaching a Virtual Machine (VM) from a particular Virtual Network Instance (VNI).

As sometimes happens, there is not a 1:1 mapping of the work areas defined in [I-D.ietf-nvo3-overlay-problem-statement] and requirements documents intended to address the problems that have been identified there.

Current control-plane requirement documents include the following:

- o Overall control-plane requirements [I-D.kreeger-nvo3-overlay-cp]
- o Control-plane requirements specific to VM-to-NVE interactions [I-D.kreeger-nvo3-hypervisor-nve-cp]

5.1. Overall Control-Plane Requirements

In this section, numbering of requirement headings corresponds to section numbering in [I-D.kreeger-nvo3-overlay-cp].

(3.1) Inner to Outer Address Mapping

The requirements document [I-D.kreeger-nvo3-overlay-cp] states that avoiding the need to "flood" traffic to support learning of mapping information from the data-plane is a goal of NVO3 candidate technological approaches.

For each candidate technology, (how) is the mapping of header information present in tenant traffic mapped to corresponding header information to be used in overlay encapsulation (this includes addresses, context identification, etc.) determined?

Supported Approach	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Control Protocol Acquisition?					
- - -	- - -	- - -	- - -	- - -	- - -
Data-Plane Learning?					

Table 1: Inner:Outer Address Mapping

(3.2) Underlying Network Multi-Destination Address(es)

The requirements document [I-D.kreeger-nvo3-overlay-cp] lists 3 approaches that may be used to deliver traffic to multiple destinations in an overlay virtual network:

1. Use the capabilities of the underlay network.
2. Require a sending NVE to replicate traffic.
3. Use a replication service provided within the overlay network.

For each delivery approach, it may be necessary to map specific multipoint (e.g. - broadcast, unknown destination or multicast) traffic to (for instance) addresses used to deliver this traffic via the underlay network.

For each technological approach, which delivery approaches are supported and does the technology provide a method by which an NVE needing to send multi-destination traffic can determine to what address, or addresses to which to send this traffic?

Supported Approach	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Underlay Network Capability					
NVE Sender Replication					
Replication Service					

Table 2: Multi-Destination Delivery

(3.3) VN Connect/Disconnect Notification

The requirements document [I-D.kreeger-nvo3-overlay-cp] states as an assumption that a mechanism exists in the overlay technology by which an NVE is notified of Tenant Systems attaching and detaching from a specific Virtual Network (VN).

For each candidate technology, does the technology currently support these functions?

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Connect Notification					

Disconnect Notification						
-------------------------	--	--	--	--	--	--

Table 3: Connect/Disconnect Notification

(3.4) VN Name to VNID Mapping

The requirements document [I-D.kreeger-nvo3-overlay-cp] concludes that having a means to map for a "VN Name to a "VN ID" may be useful.

For each technological approach we are considering, is this function currently available?

Function	NVGRE	VxLAN	VPLS	EVPN	L3VPN
VN-Name:VN-ID Mapping					

Table 4: VN Name to VN ID Mapping

5.2. VM-to-NVE Specific Control-Plane Requirements

In this section, numbering of requirement headings corresponds to section numbering in [I-D.kreeger-nvo3-hypervisor-nve-cp].

(4.1) VN Connect/Disconnect

The requirements document [I-D.kreeger-nvo3-hypervisor-nve-cp] states as a requirement that a mechanism must exist by which an NVE is notified when an end device requires a connection, or no longer requires a connection, to a specific Virtual Network (VN).

The requirements document further states as a requirement that the mechanism(s) used in a candidate technological approach must provide a local indicator (e.g. - 802.1Q tag) that the end device will use in sending traffic to, or receiving traffic from, the NVE (where that traffic is associated with the connected VN).

As an additional related requirement, the requirements document states that the NVE - once notified of a connection to a VN (by VN Name), needs to have a means for getting associated VN context information from the NVA.

For each candidate technology, does the technology currently support these functions?

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Connect Notification					
Local VN Indicator					
VN Name to VN Context Mapping					
Disconnect Notification					

Table 5: VN Connect/Disconnect

(4.2) VNIC Address Association

The requirements document [I-D.kreeger-nvo3-hypervisor-nve-cp] lists two approaches for acquiring VNIC address association information:

1. Data Plane Learning (i.e. - by inspecting source addresses in traffic received from an end device).
2. Explicit signaling from the end device when a specific VNIC address is to be associated with a tenant system.

Supported Approaches	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Data Plane Learning					
Explicit Signaling					

Table 6: VNIC Address Association

(4.3) VNIC Address Disassociation

TBD

(4.4) VNIC Shutdown/Startup/Migration

TBD

(4.5) VN Profile

TBD

6. Data Plane Requirements

In this section, numbering of requirement headings corresponds to section numbering in [I-D.ietf-nvo3-dataplane-requirements].

(3.1) Virtual Access Points (VAPs)

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
MUST support VAP identification					
1) Local interface	YES				
2) Local interface + fields in frame header	YES				

Table 7: VAP Identification Requirements

(3.2) Virtual Network Instance (VNI)

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
VAP are associated with a specific VNI at service instantiation time.	YES				

Table 8: VAP-VNI Association

(3.2.1) L2 VNI

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
L2 VNI MUST provide an emulated Ethernet multipoint service as if Tenant Systems are interconnected by a bridge (but instead by using a set of NV03 tunnels).					

Loop avoidance capability MUST be provided. - - -	- - -	- - -	- - -	- - -	- - -
In the absence of a management or control plane, data plane learning MUST be used to populate forwarding tables. - - -	- - -	- - -	- - -	- - -	- - -
When flooding is required, either to deliver unknown unicast, or broadcast or multicast traffic, the NVE MUST either support ingress replication or multicast. - - -	- - -	- - -	- - -	- - -	- - -
In this latter case, the NVE MUST be able to build at least a default flooding tree per VNI.					

Table 9: L2 VNI Service

(3.2.2) L3 VNI

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
L3 VNIs MUST provide virtualized IP routing and forwarding. - - -	- - -	- - -	- - -	- - -	- - -
L3 VNIs MUST support per-tenant forwarding instance with IP addressing isolation and L3 tunneling for interconnecting instances of the same VNI on NVEs.					

Table 10: L3 VNI Service

(3.3.1) NVO3 overlay header

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
An NVO3 overlay header MUST be included after the underlay tunnel header when forwarding tenant traffic.	YES	YES	YES	YES	YES

Table 11: Overlay Header

(3.3.1.1) Virtual Network Context Identification

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
The overlay encapsulation header MUST contain a field which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE.	YES	YES	YES	YES	YES

Table 12: Virtual Network Context Identification

(3.3.1.2) Service QoS identifier

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Traffic flows originating from different applications could rely on differentiated forwarding treatment to meet end-to-end availability and performance objectives.	NO				

Table 13: QoS Service Identification

(3.3.2.1) LAG and ECMP

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
For performance reasons, multipath over LAG and ECMP paths SHOULD be supported.	YES				

Table 14: Multipath Support

(3.3.2.2) DiffServ and ECN marking

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
[RFC2983] defines two modes for mapping the DSCP markings from inner to outer headers and vice versa. Both models SHOULD be supported.	NO				
---	---	---	---	-	---
ECN marking MUST be performed according to [RFC6040] which describes the correct ECN behavior for IP tunnels.	NO				

Table 15: DSCP and ECN Marking

(3.3.2.3) Handling of broadcast, unknown unicast, and multicast traffic

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
NV03 data plane support for either ingress replication or point-to-multipoint tunnels is required to send traffic destined to multiple locations on a per-VNI basis (e.g. L2/L3 multicast traffic, L2 broadcast and unknown	YES	YES	YES	YES	YES

unicast traffic).					
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

Table 16: Handling of Broadcast, Unknown Unicast, and Multicast Traffic

(3.4) External NVO3 connectivity

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
NVO3 services MUST interoperate with current VPN and Internet services. This may happen inside one DC during a migration phase or as NVO3 services are delivered to the outside world via Internet or VPN gateways.	YES				

Table 17: Interoperation

(3.5) Path MTU

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Classical ICMP-based MTU Path Discovery ([RFC1191], [RFC1981]) or Extended MTU Path Discovery techniques such as defined in [RFC4821].	NO				
Segmentation and reassembly support from the overlay layer operations without relying on the Tenant Systems to know about the end-to-end MTU.	YES				

Table 18: Path MTU

(3.7) NVE Multi-Homing Requirements

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Multi-homing techniques SHOULD be used to increase the reliability of an NV03 network.	NO				

Table 19: Multihoming

(3.8) OAM

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
NVE MAY be able to originate/terminate OAM messages for connectivity verification, performance monitoring, statistic gathering and fault isolation. Depending on configuration, NVEs SHOULD be able to process or transparently tunnel OAM messages, as well as supporting alarm propagation capabilities.	NO				

Table 20: OAM Messaging

7. Summary and Conclusions

TBD

8. Acknowledgements

The Authors would like to acknowledge the technical contributions of Florin Balus, Luyuan Fang, Sue Hares, Wim Henderickx, Yuichi Ikejiri, Rangaraju Iyengar, Mircea Pisica, Evelyn Roch, Ali Sajassi, Peter Ashwood-Smith and Lucy Yong as well as the initial help in editing the XML source for the document from Tom Taylor.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

Security considerations of the requirements documents referenced by this analysis document apply.

11. References

11.1. Normative References

[I-D.ashwood-nvo3-operational-requirement]

Ashwood-Smith, P., Iyengar, R., Tsou, T., Sajassi, A., Boucadair, M., Jacquenet, C., and M. Daikoku, "NVO3 Operational Requirements", draft-ashwood-nvo3-operational-requirement-03 (work in progress), July 2013.

[I-D.ietf-l2vpn-evpn]

Sajassi, A., Aggarwal, R., Henderickx, W., Balus, F., Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-04 (work in progress), July 2013.

[I-D.ietf-nvo3-dataplane-requirements]

Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", draft-ietf-nvo3-dataplane-requirements-01 (work in progress), July 2013.

[I-D.ietf-nvo3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-03 (work in progress), July 2013.

[I-D.ietf-nvo3-overlay-problem-statement]

Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-04 (work in progress), July 2013.

[I-D.kreeger-nvo3-hypervisor-nve-cp]

Kreeger, L., Narten, T., and D. Black, "Network Virtualization Hypervisor-to-NVE Overlay Control Protocol Requirements", draft-kreeger-nvo3-hypervisor-nve-cp-01 (work in progress), February 2013.

[I-D.kreeger-nvo3-overlay-cp]

Kreeger, L., Dutt, D., Narten, T., Black, D., and M. Sridharan, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp-04 (work in progress), June 2013.

[I-D.mahalingam-dutt-dcops-vxlan]

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-04 (work in progress), May 2013.

[I-D.sridharan-virtualization-nvgre]

Sridharan, M., Greenberg, A., Wang, Y., Garg, P., Venkataramiah, N., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-03 (work in progress), August 2013.

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

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

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

[RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.

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

[RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.

[RFC4762] Lasserre, M. and V. Kompella, "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, January 2007.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.

11.2. Informative References

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

Authors' Addresses

Eric Gray (editor)
Ericsson
120 Morris Avenue
Pitman, New Jersey 08071
USA

Email: eric.gray@ericsson.com

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, Massachusetts 02145
USA

Email: nabil.bitar@verizon.com

Xiaoming Chen
Huawei Technologies

Email: ming.chen@huawei.com

Marc Lasserre
Alcatel-Lucent

Email: marc.lasserre@alcatel-lucent.com

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara, California 95050
USA

Phone: +1 408 330 4424
Email: Tina.Tsou.Zouting@huawei.com
URI: <http://tinatsou.weebly.com/contact.html>

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: April 25, 2014

S. Hartman
Painless Security
D. Zhang
Huawei
M. Wasserman
Painless Security
October 22, 2013

Security Requirements of NVO3
draft-ietf-nvo3-security-requirements-01

Abstract

The draft provides a list of security requirements to benefit the design of NOV3 security mechanisms. In addition, this draft introduces the candidate techniques which could be used to fulfill such security requirements.

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 RFC 2119 [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 25, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. NVO3 Overlay Architecture	3
4. Threat Model	4
4.1. Outsider Capabilities	4
4.2. Insider Capabilities	5
4.3. Security Issues In Scope and Out of Scope	5
5. Security Requirements and Candidate Approaches	6
5.1. Control/Data Traffic within Overlay	6
5.1.1. Control Plane Security	6
5.1.2. Data Plane	9
5.2. Control/Data Traffic between NVEs and Hypervisors	10
5.2.1. Distributed Deployment of NVE and Hypervisor	11
5.3. Key Management	13
6. IANA Considerations	14
7. Security Considerations	14
8. Acknowledgements	14
9. References	15
9.1. Normative References	15
9.2. Informative References	15
Authors' Addresses	16

1. Introduction

Security is a key issue which needs to be considered in the design of a data center network. This document discusses the security risks that a NVO3 network may encounter and the security requirements that a NVO3 network needs to fulfill. In addition, this draft attempts to discuss the security techniques which could be applied to fulfill such requirements.

The remainder of this document is organized as follows. Section 2 introduces the terms used in this memo. Section 3 gives a briefly introduction of the NVO3 network architecture. Section 4 discusses the attack model of this work. Section 5 describes the essential security requirements which should be fulfilled in the generation of a NVO3 network.

2. Terminology

This document uses the same terminology as found in the NVO3 Framework document [I-D.ietf-nvo3-framework] and [I-D.kreeger-nvo3-hypervisor-nve-cpl]. Some of the terms defined in the framework document have been repeated in this section for the convenience of the reader, along with additional terminology that is used by this document.

Tenant System (TS): A physical or virtual system that can play the role of a host, or a forwarding element such as a router, switch, firewall, etc. It belongs to a single tenant and connects to one or more VNs of that tenant.

End System (ES): An end system of a tenant, which can be, e.g., a virtual machine (VM), a non-virtualized server, or a physical appliance. A TS is attached to a Network Virtualization Edge (NVE) node.

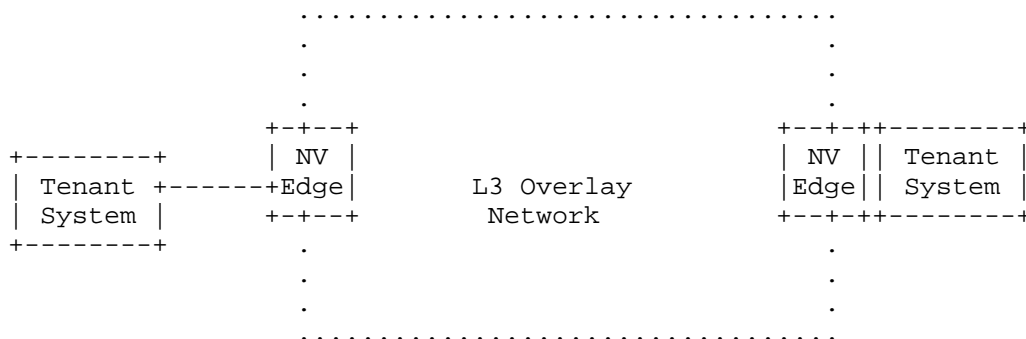
Network Virtualization Edge (NVE): An NVE implements network virtualization functions that allow for L2/L3 tenant separation and tenant-related control plane activity. An NVE contains one or more tenant service instances whereby a TS interfaces with its associated instance. The NVE also provides tunneling overlay functions.

Virtual Network (VN): This is a virtual L2 or L3 domain that belongs to a tenant.

Network Virtualization Authority (NVA). A back-end system that is responsible for distributing and maintaining the mapping information for the entire overlay system. Note that the WG never reached consensus on what to call this architectural entity within the overlay system, so this term is subject to change.

NVO3 device: In this memo, the devices (e.g., NVE and NVA) work cooperatively to provide NVO3 overlay functionalities are called as NVO3 devices.

3. NVO3 Overlay Architecture



This figure illustrates a simple nov3 overlay example where NVEs provide a logical L2/L3 interconnect for the TSEs that belong to a specific tenant network over L3 networks. A packet from a tenant system is encapsulated when they reach the egress NVE. Then encapsulated packet is then sent to the remote NVE through a proper tunnel. When reaching the ingress NVE, the packet is decapsulated and forwarded to the target tenant system. The address advertisements and tunnel mappings are distributed among the NVEs through either distributed control protocols or by certain centralized servers (called NVAs).

4. Threat Model

To benefit the discussion, in this analysis work, attacks are classified into two categories: inside attacks and outside attacks. An attack is considered as an inside attack if the adversary performing the attack (inside attacker or insider) has got certain privileges in changing the configuration or software of a NVO3 device and initiates the attack within the overlay security perimeter. In contrast, an attack is referred to as an outside attack if the adversary performing the attack (outside attacker or outsider) has no such privilege and can only initiate the attacks from compromised TSeS (or the network devices of the underlying network which the overlay is located upon). Note that in a complex attack inside and outside attacking operations may be performed in a well organized way to expand the damages caused by the attack.

4.1. Outsider Capabilities

The following capabilities of outside attackers MUST be considered in the design of a NOV3 security mechanism:

1. Eavesdropping on the packets,

2. Replaying the intercepted packets, and
3. Generating illegal packets and injecting them into the network.

With a successful outside attack, an attacker may be able to:

1. Analyze the traffic pattern within the network,
2. Disrupt the network connectivity or degrade the network service quality, or
3. Access the contents of the data/control packets if they are not properly encrypted.

4.2. Insider Capabilities

It is assumed that an inside attacker can perform any types of outside attacks from the inside or outside of the overlay perimeter. In addition, in an inside attack, an attacker may use already obtained privilege to, for instance,

1. Interfere with the normal operations of the overlay as a legal entity, by sending packets containing invalid information or with improper frequencies,
2. Perform spoofing attacks and impersonate another legal device to communicate with victims using the cryptographic information it obtained, and
3. Access the contents of the data/control packets if they are encrypted with the keys held by the attacker.

4.3. Security Issues In Scope and Out of Scope

During the specification of security requirements, the following security issues needs to be considered:

1. Insecure underlying network. It is normally assumed that a underlying network connecting NOV3 devices (NVEs and NVAs) is secure if it is located within a data center and cannot be directly accessed by tenants. However, in a virtual data center scenario, a NVO3 overlay scatters across different sites which are connected through the public network. Outside attacks may be raised from the underlying network.
2. Insider attacker. During the design of a security solution for a NVO3 network, the inside attacks raised from compromised NVO3 devices (NVEs and NVAs) needs to be considered.

3. Insecure tenant network. It is reasonable to consider the conditions where the network connecting TSeS and NVEs is accessible to outside attackers.

The following issues are out of scope of consideration in this document:

1. In this memo it is assumed that security protocols, algorithms, and implementations provide the security properties for which they are designed; attacks depending on a failure of this assumption are out of scope. As an example, an attack caused by a weakness in a cryptographic algorithm is out of scope, while an attack caused by failure to use confidentiality when confidentiality is a security requirement is in scope.
2. In practice an attacker controlling an underlying network device may break the communication of the overlays by discarding or delaying the delivery of the packets passing through it. However, this type of attack is out of scope.

5. Security Requirements and Candidate Approaches

This section introduces the security requirements and candidate solutions.

5.1. Control/Data Traffic within Overlay

This section analyzes the security issues in the control and data plans of a NVO3 overlay.

5.1.1. Control Plane Security

REQ1: A NVO3 security solution MUST enable two NVO3 devices (NVE or NVA) to perform mutual authentication before exchanging control packets.

This requirement is used to prevent an attacker from impersonating a legal NVO3 device and sending out bogus control packets without being detected.

The authentication between devices can be performed as a part of automated key management protocols (e.g., IKEv2[RFC5996], EAP[RFC4137], etc.). After such an authentication procedure, a device can find out whether its peer holds valid security credentials and is the one who it has claimed. Additionally, the keys shared between the devices can be also used for the authentication purpose. For instance, assumed a NVE and a NVA have shared a secret key without known by any other third parties.

The NVE can ensure that a device that it is communicating with is the NVA if the device can prove that it possesses the shared key.

a: The identity of the network devices SHOULD be verified during authentication.

In some authentication mechanisms, instead of verifying the peers' identities, the authentication result can only prove that a device joining the authentication is a legal member of a group. However, for a better damage confining capability to insider attacker, it is recommended to verify the devices' identities during authentication. Therefore, an insider attacker cannot impersonate others, even when it holds legal credentials or keys.

REQ2: Before accepting a control packet, the device receiving the packet MUST verify whether the packet comes from one which has the privilege to send that packet.

This is an authorization requirement. A device needs to clarify the roles (e.g., a NVE or a NVA) that its authentication peer acts as in the overlay. Therefore, if a compromised NVE uses its credentials to impersonate a NVA to communicate with other NVEs, it will be detected. In addition, authorization is important for enforcing the VN isolation, a device only can distribute control packets within the VNs it is involved within. If a control packet about a VN is sent from a NVE which is not authorized to support the VN, the packet will not be accepted.

Normally, it is assumed that the access control operations are based on the authentication results. The simple authorization mechanisms (such as ACLs which filters packets based on the packet addresses) can be used as auxiliary approaches since they are relatively easy to bypass if attackers can access to the network and modify packets.

REQ3: Integrity, confidentiality, and origin Authentication protection for Control traffics

It is the responsibility of a NV03 overlay to protect the control packets transported over the overlay against the attacks raised from the underlying network.

a: The integrity and origin authentication of the packets MUST be guaranteed.

With this requirement, the receiver can ensure that the packets are from the legitimate sender, not replayed, and not modified during the transportation.

b: The signaling packets SHOULD be encrypted.

On many occasions, the signaling packets can be transported in plaintext. However, In the cases where the information contained within the signaling packets are sensitive or valuable to attackers , the signaling packets related with that tenant need be encrypted.

To achieve such objectives, when the network devices exchange control plane packets, integrated security mechanisms or underlying security protocols need to provided. In addition, cryptographic keys need to be deployed manually in advance or dynamically generated by using certain automatic key management protocols (e.g., TLS [RFC5246]). The keys are used to generate digests for or encrypt control packets.

REQ4: The toleration of DOS attacks

a: Frequency in distributing control packets within in the overlay MUST be limited.

The issues within DOS attacks also need to be considered in designing the overlay control plane. For instance, in the VXLAN solution[I-D.mahalingam-dutt-dcops-vxlan], an attacker attached to a NVE can try to manipulate the NVE to keep multicasting control packets by sending a large amount of ARP packets to query the inexistent VMs. In order to mitigate this type of attack, the NVEs SHOULD be only allowed to send signaling packet in the overlay with a limited frequency. When there are centralized servers (e.g., the backend oracles providing mapping information for NVEs[I-D.ietf-nvo3-overlay-problem-statement], or the SDN controllers) are located within the overlay, the potential security risks caused by DDOS attack on such servers can be more serious.

b: Mitigation of amplification attacks SHOULD be provided.

During the design of the control plane, it is important to consider the amplification effects. For instance, if NVEs may generate a large response to a short request, an attacker may send spoofed requests to the NVEs with the source address of a victim. Then the NVEs will send the response to the victim and result in DDOS attacks.

If the amplification effect cannot be avoided in the control protocol, the requirements 1,2,3, and 4a can all be used to benefit the mitigation of this type of attacks.

REQ5: The key management solution MUST be able to confine the scope of key distribution and provide different keys to isolate the control traffic according to different security requirements.

a: It SHOULD be guaranteed that different keys are used to secure the control packets exchanged within different tenant networks.

This requirement can be used to provide a basic attack confinement capability. The compromise of a NVE working within a tenant will not result in the key leakage of other tenant networks.

b: It SHOULD be guaranteed that different keys are used to secure the control packets exchanges with different VNs.

This requirement can be used to provide a better attack confinement capability for the control plane. The compromise of a NVE working within a VN will not result in the key leakage of other VNs. However, since there is only a single key used for securing the data traffic within a VN, an attacker which has compromised a NVE within the VN may be able to impersonate any other NVEs within the VN to send out bogus control packets. In addition, the key management overheads introduced by key revocation also need to be considered[RFC4046]. When a NVE stops severing a VN, the key used for the VN needs to be revoked, and a new key needs to be distributed for the NVO3 devices still within the VN.

If we expect to provide a even stronger confinement capability and prevent a compromised NVE from impersonating other NVEs even when they are in the same VN, different NVEs working inside a VN need to secure their signaling packets with different keys.

If there is automated key management deployed, the authentication and authorization can be used to largely mitigate the isolation issues. When a NVE attempts to join a VN, the NVE needs to be authenticated and prove that it have sufficient privileges. Then, a new key (or a set of keys) will be generated to secure its control packet exchanged with this VN.

5.1.2. Data Plane

[I-D.ietf-nvo3-framework] specifies a NVO3 overlay needs to generate tunnels between NVEs for data transportation. When a data packet reaches the boundary of a overlay, it will be encapsulated and forwarded to the destination NVE through a proper tunnel.

REQ6: Integrity, confidentiality, and origin authentication protection for data traffics

a: The integrity and origin authentication of data traffics MUST be guaranteed when the underlying network is not secure.

During the transportation of data packets, it is the responsibility of the NVO3 overlay to deal with the attacks from the underlying network. For instance, an inside attacker compromising a underlying network device may intercept an encapsulated data packet transported within a tunnel, modify the contents in the encapsulating tunnel packet and, transfer it into another tunnel without being detected. When the modified packet reaches a NVE, the NVE may decapsulated the data packet and forward it into a VN according to the information within the encapsulating header generated by the attacker. Similarly, a compromised NVE may try to redirect the data packets within a VN into another VN by adding improper encapsulating tunnel headers to the data packets.

Under such circumstances, in order to enforce the VN isolation property, underlying security protocols need to provided. Signatures or digests need to be generated for both data packets and the encapsulating tunnel headers in order to provide data origin authentication and integrity protection.

b: The confidentiality protection of data traffics SHOULD be provided, when the underlying network is not secure.

If the data traffics from the TSes is sensitive, they needs to be encrypted during the tunnels. However, if the data traffics is not valuable and sensitive, the encryption is not necessary.

REQ7: Different tunnels SHOULD be secured with different keys

This requirement can be used to provide a basic attack confinement capability. When different tunnels secured with different keys, the compromise of a key in a tunnel will not affect the security of others.

5.2. Control/Data Traffic between NVEs and Hypervisors

Assume there is a VNE providing a logical L2/L3 interconnect for a set of TSes. Apart from data traffics, the NVE and certain TSes (i.e., Hypervisors) also need to exchange signaling packets in order to facilitate, e.g., VM online detection, VM migration detection, or auto-provisioning/service discovery [I-D.ietf-nvo3-framework].

The NVE and its associated TSes can be deployed in a distributed way (e.g., a NVE is implemented in an individual device, and VMs are located on servers) or in a co-located way (e.g., a NVE and the TSes it serves are located on the same server).

5.2.1. Distributed Deployment of NVE and Hypervisor

In this case, the data and control traffic between the NVE and the TSes are exchanged over network.

5.2.1.1. Control Plane

REQ8: Mutual authentication **MUST** be performed between a NVE and a TS at the beginning of their communication, if the network connecting them is not secure.

Mutual authentication is used to guarantee that an attacker cannot impersonate a legal NVE or a hypervisor without being detected.

There are various ways to perform mutual authentication. If there are auto key management mechanism (e.g., IKEv2, EAP), the NVE and the TS can use their credential to perform authentication. If there a key pre-distributed between a NVE and a TS, an entity can also use the key verify the identity of is remote peer.

If practice, a NVE and a TS may simply use IP or MAC addresses to identify each other. This type of technique can be used as a complementary approach although it may becomes vulnerable if attackers can inject bogus control packets the network and modify the packets transported between the NVE and TS.

REQ9: Before accepting a control packet, the receiver device **MUST** verify whether the packet comes from one which has the privilege to send that packet.

This is an authorization requirement. A device needs to clarify the roles (e.g., a TS or a NVE) of the device that it is communicating with. Therefore, if a compromised TS attempts to use it credentials to impersonate a NVE to communicate with other TSes, it will be detected.

Authorization is very important to guarantee the isolation property. For instance, if a compromised hypervisor tries to elevate its privilege and interfere the VNs that it is not supposed to be involved within, its attempt will be detected and rejected.

Normally, it is assumed that the access control operations are based on the authentication results. The simple authorization mechanisms (such as ACLs which filters packets based on the packet addresses) can be used as complementary solutions.

REQ10: Integrity, Confidentiality, and Origin Authentication for Control Packets

a:The security solution of a NVE network MUST be able to provide integrity protection and origin authentication for the control packets exchanged between a NVE and a TS if they have to use an insecure network to transport their packet.

This requirement can prevent an attacker from illegally interfere with the normal operations of NVEs and TSes by injecting bogus control packets into the network.

b:The confidentiality protection for the control packet exchange SHOULD be provided.

When the contents of the control packets (e.g., the location of a ES, when a VM migration happens) are sensitive to a tenant, the control packet needs to be encrypted.

There are various security protocols (such as IPsec, SSL, and TCP-AO) can be used for transport control packets. In addition, it is possible to define integrated security solutions for the control packets.

In order to secure the control traffic, cryptographic keys need to be distributed to generate digests or signatures for the control packets. Such cryptographic keys can be manually deployed in advance or dynamically generated with certain automatic key management protocols (e.g., TLS [RFC5246]).

REQ11: The key management solution MUST be able to confine the scope of key distribution and provide different keys to isolate the control traffic according to different security requirements.

a: If assuming TSes (hypervisors) will not be compromised, the TSes belonging to different Tenants MUST use different keys to secure the control packet exchanges with their NVE.

This requirement is used to enforce the security boundaries of different tenant networks. Since different tenants belong to different security domains and may be competitive to each other, the control plane traffics need to be carefully isolated so that an attacker from a tenant cannot affect the operations of another tenant network.

b: If assuming the hypervisors can be compromised, the TSeS belonging to different VNs MUST use different keys to secure the control packets exchanges with their NVE.

Therefore, if a key used for a VN is compromised, other VNs will not be affected. This requirement is used to ensure the VN isolation property.

5.2.1.2. Data Plane

REQ12: The data traffic isolation of different VNs MUST be guaranteed.

In [I-D.ietf-nvo3-overlay-problem-statement], the data plane isolation requirement amongst different VNs has been discussed. The traffic within a virtual network can only be transited into another one in a controlled fashion (e.g., via a configured router and/or a security gateway). Therefore, if the NVE supports multiple VNs concurrently, the data traffic in different VNs MUST be isolated.

a: The security solution of a NVE network MUST be able to provide integrity protection and origin authentication for the data packets exchanged between a NVE and a TS if they have to use an insecure network to transport their data packet.

In practice, the data traffics in different VNs can be isolated physically or by using VPN technologies. If the network connecting the NVE and the TSeS is potentially accessible to attackers, security solutions need to be considered to prevent an attacker locating in the middle between the NVE and TS from modifying the VN identification information in the packet headers so as to manipulate the NVE to transport the data packets within a VN to another. The security protocols such as IPsec and TCP-AO, can be used to enforce the isolation property if necessary.

The key management requirement R11 can be applied here for data traffic

5.3. Key Management

REQ13: A security solution for NVO3 SHOULD provide automated key management mechanisms.

In the cases where there are a large amount of NVEs working within a NVO3 overlay, manual key management may become infeasible. First, it could be burdensome to deploy pre-shared keys for thousands of NVEs, not to mention that multiple keys may need to be deployed on a single device for different purposes. Key derivation can be used to mitigate this problem. Using key derivation functions, multiple keys for different usages can be derived from a pre-shared master key. However, key derivation cannot protect against the situation where a system was incorrectly trusted to have the key used to perform the derivation. If the master key were somehow compromised, all the resulting keys would need to be changed [RFC4301]. In addition, VM migration will introduce challenges to manual key management. The migration of a VM in a VN may cause the change of the NVEs which are involved within the VN. When a NVE is newly involved within a VN, it needs to get the key to join the operations within the VN. If a NVE stops supporting a VN, it should not keep the keys associated with that VN. All those key updates need to be performed at run time, and difficult to be handled by human beings. As a result, it is reasonable to introduce automated key management solutions such as EAP [RFC4137] for NVO3 overlays.

Without the support automated key management mechanisms, some security functions of certain security protocols cannot work properly. For instance, the anti-replay mechanism of IPsec is turned off without the support of automated key management mechanisms. Therefore, if IPsec is selected to protect the control packets. In this case, the system may suffer from the replay attacks.

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

TBD

8. Acknowledgements

Thanks a lot for the comments from Melinda Shore, and Zu Qiang.

9. References

9.1. Normative References

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

9.2. Informative References

- [I-D.ietf-ipsecme-ad-vpn-problem]
Manral, V. and S. Hanna, "Auto Discovery VPN Problem Statement and Requirements", draft-ietf-ipsecme-ad-vpn-problem-09 (work in progress), July 2013.
- [I-D.ietf-nvo3-framework]
Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-03 (work in progress), July 2013.
- [I-D.ietf-nvo3-overlay-problem-statement]
Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-04 (work in progress), July 2013.
- [I-D.kreeger-nvo3-hypervisor-nve-cp]
Kreeger, L., Narten, T., and D. Black, "Network Virtualization Hypervisor-to-NVE Overlay Control Protocol Requirements", draft-kreeger-nvo3-hypervisor-nve-cp-01 (work in progress), February 2013.
- [I-D.mahalingam-dutt-dcops-vxlan]
Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-05 (work in progress), October 2013.
- [RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture", RFC 4046, April 2005.
- [RFC4137] Vollbrecht, J., Eronen, P., Petroni, N., and Y. Ohba, "State Machines for Extensible Authentication Protocol (EAP) Peer and Authenticator", RFC 4137, August 2005.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

Authors' Addresses

Sam Hartman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Email: hartmans@painless-security.com
URI: <http://www.painless-security.com>

Dacheng Zhang
Huawei
Beijing
China

Email: zhangdacheng@huawei.com

Margaret Wasserman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Phone: +1 781 405 7464
Email: mrw@painless-security.com
URI: <http://www.painless-security.com>

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 25, 2014

D. Black
EMC
J. Hudson
Brocade
L. Kreeger
Cisco
M. Lasserre
Alcatel-Lucent
T. Narten
IBM
October 22, 2013

An Architecture for Overlay Networks (NVO3)
draft-narten-nvo3-arch-01

Abstract

This document presents a high-level overview architecture for building overlay networks in NVO3. The architecture is given at a high-level, showing the major components of an overall system. An important goal is to divide the space into individual smaller components that can be implemented independently and with clear interfaces and interactions with other components. It should be possible to build and implement individual components in isolation and have them work with other components with no changes to other components. That way implementers have flexibility in implementing individual components and can optimize and innovate within their respective components without requiring changes to other components.

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 25, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	3
2. Terminology	3
3. Background	4
3.1. VN Service (L2 and L3)	5
3.2. Network Virtualization Edge (NVE)	6
3.3. Network Virtualization Authority (NVA)	7
3.4. VM Orchestration Systems	8
4. Network Virtualization Edge (NVE)	9
4.1. NVE Co-located With Server Hypervisor	9
4.2. Split-NVE	10
4.3. NVE State	11
5. Tenant System Types	12
5.1. Overlay-Aware Network Service Appliances	12
5.2. Bare Metal Servers	12
5.3. Gateways	13
5.4. Distributed Gateways	13
6. Network Virtualization Authority	14
6.1. How an NVA Obtains Information	14
6.2. Internal NVA Architecture	15
6.3. NVA External Interface	15
7. NVE-to-NVA Protocol	17
7.1. NVE-NVA Interaction Models	17
7.2. Direct NVE-NVA Protocol	18
7.3. Propagating Information Between NVEs and NVAs	19
8. Federated NVAs	20
8.1. Inter-NVA Peering	22
9. Control Protocol Work Areas	23
10. NVO3 Data Plane Encapsulation	23
11. Operations and Management	24
12. Summary	24
13. Acknowledgments	24

14. IANA Considerations	24
15. Security Considerations	24
16. Informative References	24
Appendix A. Change Log	26
A.1. Changes From -00 to -01	26
Authors' Addresses	26

1. Introduction

This document presents a high-level architecture for building overlay networks in NVO3. The architecture is given at a high-level, showing the major components of an overall system. An important goal is to divide the space into smaller individual components that can be implemented independently and with clear interfaces and interactions with other components. It should be possible to build and implement individual components in isolation and have them work with other components with no changes to other components. That way implementers have flexibility in implementing individual components and can optimize and innovate within their respective components without necessarily requiring changes to other components.

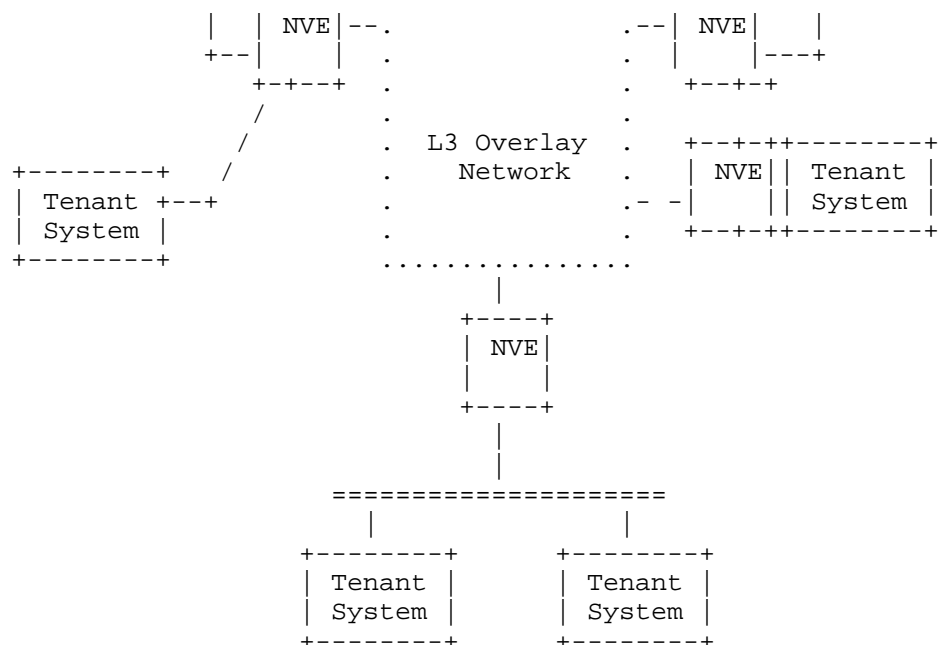
The motivation for overlay networks is given in [I-D.ietf-nvo3-overlay-problem-statement]. "Framework for DC Network Virtualization" [I-D.ietf-nvo3-framework] provides a framework for discussing overlay networks generally and the various components that must work together in building such systems. This document differs from the framework document in that it doesn't attempt to cover all possible approaches within the general design space. Rather, it describes one particular approach.

This document is intended to be a concrete strawman that can be used for discussion within the IETF NVO3 WG on what the NVO3 architecture should look like.

2. Terminology

This document uses the same terminology as [I-D.ietf-nvo3-framework]. In addition, the following terms are used:

NV Domain A Network Virtualization Domain is an administrative construct that defines a Network Virtualization Authority (NVA), the set of Network Virtualization Edges (NVEs) associated with that NVA, and the set of virtual networks the NVA manages and supports. NVEs are associated with a (logically centralized) NVA, and an NVE supports communication for any of the virtual networks in the domain.



The dotted line indicates a network connection (i.e., IP).

Figure 1: NV03 Generic Reference Model

The following subsections describe key aspects of an overlay system in more detail. Section 3.1 describes the service model (Ethernet vs. IP) provided to Tenant Systems. Section 3.2 describes NVEs in more detail. Section 3.3 introduces the Network Virtualization Authority, from which NVEs obtain information about virtual networks. Section 3.4 provides background on VM orchestration systems and their use of virtual networks.

3.1. VN Service (L2 and L3)

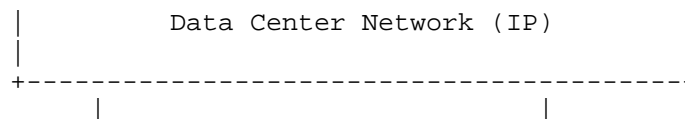
A Virtual Network provides either L2 or L3 service to connected tenants. For L2 service, VNs transport Ethernet frames, and a Tenant System is provided with a service that is analogous to being connected to a specific L2 C-VLAN. L2 broadcast frames are delivered to all (and multicast frames delivered to a subset of) the other Tenant Systems on the VN. To a Tenant System, it appears as if they are connected to a regular L2 Ethernet link. Within NVO3, tenant frames are tunneled to remote NVEs based on the MAC addresses of the frame headers as originated by the Tenant System. On the underlay, NVO3 packets are forwarded between NVEs based on the outer addresses of tunneled packets.

For L3 service, VNs transport IP datagrams, and a Tenant System is provided with a service that only supports IP traffic. Within NVO3, tenant frames are tunneled to remote NVEs based on the IP addresses of the packet originated by the Tenant System; any L2 destination addresses provided by Tenant Systems are effectively ignored.

L2 service is intended for systems that need native L2 Ethernet service and the ability to run protocols directly over Ethernet (i.e., not based on IP). L3 service is intended for systems in which all the traffic can safely be assumed to be IP. It is important to note that whether NVO3 provides L2 or L3 service to a Tenant System, the Tenant System does not generally need to be aware of the distinction. In both cases, the virtual network presents itself to the Tenant System as an L2 Ethernet interface. An Ethernet interface is used in both cases simply as a widely supported interface type that essentially all Tenant Systems already support. Consequently, no special software is needed on Tenant Systems to use an L3 vs. an L2 overlay service.

3.2. Network Virtualization Edge (NVE)

Tenant Systems connect to NVEs via a Tenant System Interface (TSI). The TSI logically connects to the NVE via a Virtual Access Point (VAP) as shown in Figure 2. To the Tenant System, the TSI is like a NIC; the TSI presents itself to a Tenant System as a normal network interface. On the NVE side, a VAP is a logical network port (virtual or physical) into a specific virtual network. Note that two different Tenant Systems (and TSIs) attached to a common NVE can share a VAP (e.g., TS1 and TS2 in Figure 2) so long as they connect to the same Virtual Network.



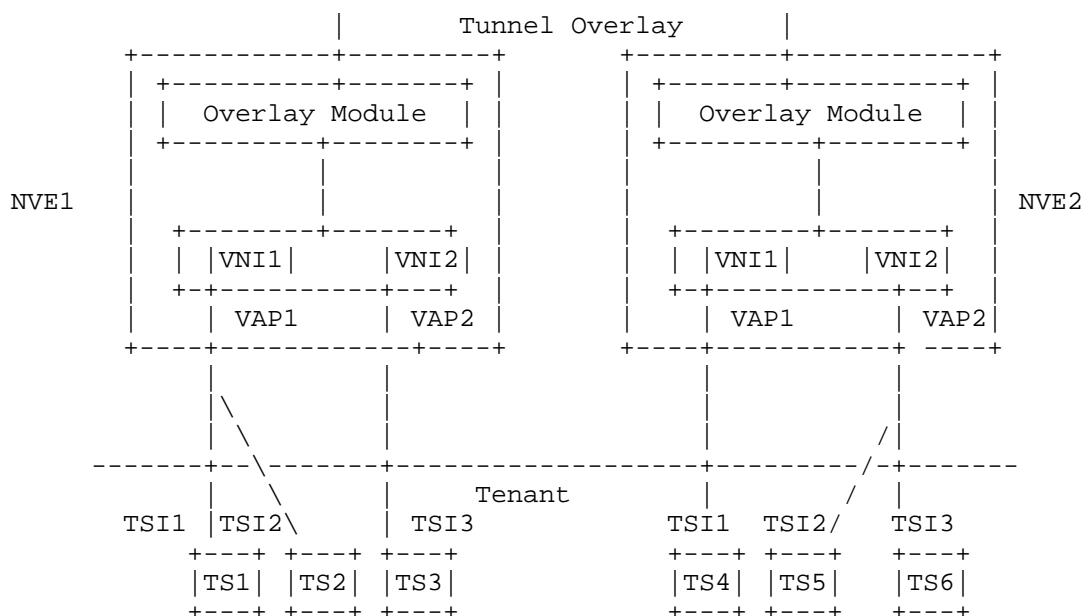


Figure 2: NVE Reference Model

The Overlay Module performs the actual encapsulation and decapsulation of tunneled packets. The NVE maintains state about the virtual networks it is a part of so that it can provide the Overlay Module with such information as the destination address of the NVE to tunnel a packet to, or the Context ID that should be placed in the encapsulation header to identify the virtual network a tunneled packet belong to.

On the data center network side, the NVE sends and receives native IP traffic. When ingressing traffic from a Tenant System, the NVE identifies the egress NVE to which the packet should be sent, adds an overlay encapsulation header, and sends the packet on the underlay network. When receiving traffic from a remote NVE, an NVE strips off the encapsulation header, and delivers the (original) packet to the appropriate Tenant System.

Conceptually, the NVE is a single entity implementing the NVO3 functionality. In practice, there are a number of different implementation scenarios, as described in detail in Section 4.

3.3. Network Virtualization Authority (NVA)

Address dissemination refers to the process of learning, building and distributing the mapping/forwarding information that NVEs need in

order to tunnel traffic to each other on behalf of communicating Tenant Systems. For example, in order to send traffic to a remote Tenant System, the sending NVE must know the destination NVE for that Tenant System.

One way to build and maintain mapping tables is to use learning, as 802.1 bridges do [IEEE-802.1Q]. When forwarding traffic to multicast or unknown unicast destinations, an NVE could simply flood traffic everywhere. While flooding works, it can lead to traffic hot spots and can lead to problems in larger networks.

Alternatively, NVEs can make use of a Network Virtualization Authority (NVA). An NVA is the entity that provides address mapping and other information to NVEs. NVEs interact with an NVA to obtain any required address mapping information they need in order to properly forward traffic on behalf of tenants. The term NVA refers to the overall system, without regards to its scope or how it is implemented. NVAs provide a service, and NVEs access that service via an NVE-to-NVA protocol.

Even when an NVA is present, learning could be used as a fallback mechanism, should the NVA be unable to provide an answer or for other reasons. This document does not consider flooding approaches in detail, as there are a number of benefits in using an approach that depends on the presence of an NVA.

NVAs are discussed in more detail in Section 6.

3.4. VM Orchestration Systems

VM Orchestration systems manage server virtualization across a set of servers. Although VM management is a separate topic from network virtualization, the two areas are closely related. Managing the creation, placement, and movements of VMs also involves creating, attaching to and detaching from virtual networks. A number of existing VM orchestration systems have incorporated aspects of virtual network management into their systems.

When a new VM image is started, the VM Orchestration system determines where the VM should be placed, interacts with the hypervisor on the target server to load and start the server and controls when a VM should be shutdown or migrated elsewhere. VM Orchestration systems also have knowledge about how a VM should connect to a network, possibly including the name of the virtual network to which a VM is to connect. The VM orchestration system can pass such information to the hypervisor when a VM is instantiated. VM orchestration systems have significant (and sometimes global) knowledge over the domain they manage. They typically know on what

servers a VM is running, and meta data associated with VM images can be useful from a network virtualization perspective. For example, the meta data may include the addresses (MAC and IP) the VMs will use and the name(s) of the virtual network(s) they connect to.

VM orchestration systems run a protocol with an agent running on the hypervisor of the servers they manage. That protocol can also carry information about what virtual network a VM is associated with. When the orchestrator instantiates a VM on a hypervisor, the hypervisor interacts with the NVE in order to attach the VM to the virtual networks it has access to. In general, the hypervisor will need to communicate significant VM state changes to the NVE. In the reverse direction, the NVE may need to communicate network connectivity information back to the hypervisor. Example VM orchestration systems in use today include VMware's vCenter Server or Microsoft's System Center Virtual Machine Manager. Both can pass information about what virtual networks a VM connects to down to the hypervisor. The protocol used between the VM orchestration system and hypervisors is generally proprietary.

It should be noted that VM orchestration systems may not have direct access to all networking related information a VM uses. For example, a VM may make use of additional IP or MAC addresses that the VM management system is not aware of.

4. Network Virtualization Edge (NVE)

As introduced in Section 3.2 an NVE is the entity that implements the overlay functionality. This section describes NVEs in more detail. An NVE will have two external interfaces:

Tenant Facing: On the tenant facing side, an NVE interacts with the hypervisor (or equivalent entity) to provide the NVO3 service. An NVE will need to be notified when a Tenant System "attaches" to a virtual network (so it can validate the request and set up any state needed to send and receive traffic on behalf of the Tenant System on that VN). Likewise, an NVE will need to be informed when the Tenant System "detaches" from the virtual network so that it can reclaim state and resources appropriately.

DCN Facing: On the data center network facing side, an NVE interfaces with the data center underlay network, sending and receiving tunneled IP packets to and from the underlay. The NVE may also run a control protocol with other entities on the network, such as the Network Virtualization Authority.

4.1. NVE Co-located With Server Hypervisor

When server virtualization is used, the entire NVE functionality will typically be implemented as part of the hypervisor and/or virtual switch on the server. In such cases, the Tenant System interacts with the hypervisor and the hypervisor interacts with the NVE. Because the interaction between the hypervisor and NVE is implemented entirely in software on the server, there is no "on-the-wire" protocol between Tenant Systems (or the hypervisor) and the NVE that needs to be standardized. While there may be APIs between the NVE and hypervisor to support necessary interaction, the details of such an API are not in-scope for the IETF to work on.

Implementing NVE functionality entirely on a server has the disadvantage that server CPU resources must be spent implementing the NVO3 functionality. Experimentation with overlay approaches and previous experience with TCP and checksum adapter offloads suggests that offloading certain NVE operations (e.g., encapsulation and decapsulation operations) onto the physical network adaptor can produce performance improvements. As has been done with checksum and /or TCP server offload and other optimization approaches, there may be benefits to offloading common operations onto adaptors where possible. Just as important, the addition of an overlay header can disable existing adaptor offload capabilities that are generally not prepared to handle the addition of a new header or other operations associated with an NVE.

While the details of how to split the implementation of specific NVE functionality between a server and its network adaptors is outside the scope of IETF standardization, the NVO3 architecture should support such separation. Ideally, it may even be possible to bypass the hypervisor completely on critical data path operations so that packets between a TS and its VN can be sent and received without having the hypervisor involved in each individual packet operation.

4.2. Split-NVE

Another possible scenario leads to the need for a split NVE implementation. A hypervisor running on a server could be aware that NVO3 is in use, but have some of the actual NVO3 functionality implemented on an adjacent switch to which the server is attached. While one could imagine a number of link types between a server and the NVE, the simplest deployment scenario would involve a server and NVE separated by a simple L2 Ethernet link, across which LLDP runs. A more complicated scenario would have the server and NVE separated by a bridged access network, such as when the NVE resides on a ToR, with an embedded switch residing between servers and the ToR.

While the above talks about a scenario involving a hypervisor, it should be noted that the same scenario can apply to Network Service

Appliances as discussed in Section 5.1. In general, when this document discusses the interaction between a hypervisor and NVE, the discussion applies to Network Service Appliances as well.

For the split NVE case, protocols will be needed that allow the hypervisor and NVE to negotiate and setup the necessary state so that traffic sent across the access link between a server and the NVE can be associated with the correct virtual network instance. Specifically, on the access link, traffic belonging to a specific Tenant System would be tagged with a specific VLAN C-TAG that identifies which specific NVO3 virtual network instance it belongs to. The hypervisor-NVE protocol would negotiate which VLAN C-TAG to use for a particular virtual network instance. More details of the protocol requirements for functionality between hypervisors and NVEs can be found in [I-D.kreeger-nvo3-hypervisor-nve-cp].

4.3. NVE State

NVEs maintain internal data structures and state to support the sending and receiving of tenant traffic. An NVE may need some or all of the following information:

1. An NVE keeps track of which attached Tenant Systems are connected to which virtual networks. When a Tenant System attaches to a virtual network, the NVE will need to create or update local state for that virtual network. When the last Tenant System detaches from a given VN, the NVE can reclaim state associated with that VN.
2. For tenant unicast traffic, an NVE maintains a per-VN table of mappings from Tenant System (inner) addresses to remote NVE (outer) addresses.
3. For tenant multicast (or broadcast) traffic, an NVE maintains a per-VN table of mappings and other information on how to deliver multicast (or broadcast) traffic. If the underlying network supports IP multicast, the NVE could use IP multicast to deliver tenant traffic. In such a case, the NVE would need to know what IP underlay multicast address to use for a given VN. Alternatively, if the underlying network does not support multicast, an NVE could use serial unicast to deliver traffic. In such a case, an NVE would need to know which destinations are subscribers to the tenant multicast group. An NVE could use both approaches, switching from one mode to the other depending on such factors as bandwidth efficiency and group membership sparseness.

4. An NVE maintains necessary information to encapsulate outgoing traffic, including what type of encapsulation and what value to use for a Context ID within the encapsulation header.
5. In order to deliver incoming encapsulated packets to the correct Tenant Systems, an NVE maintains the necessary information to map incoming traffic to the appropriate VAP and Tenant System.
6. An NVE may find it convenient to maintain additional per-VN information such as QoS settings, Path MTU information, ACLs, etc.

5. Tenant System Types

This section describes a number of special Tenant System types and how they fit into an NVO3 system.

5.1. Overlay-Aware Network Service Appliances

Some Network Service Appliances [I-D.ietf-nvo3-nve-nva-cp-req] (virtual or physical) provide tenant-aware services. That is, the specific service they provide depends on the identity of the tenant making use of the service. For example, firewalls are now becoming available that support multi-tenancy where a single firewall provides virtual firewall service on a per-tenant basis, using per-tenant configuration rules and maintaining per-tenant state. Such appliances will be aware of the VN an activity corresponds to while processing requests. Unlike server virtualization, which shields VMs from needing to know about multi-tenancy, a Network Service Appliance explicitly supports multi-tenancy. In such cases, the Network Service Appliance itself will be aware of network virtualization and either embed an NVE directly, or implement a split NVE as described in Section 4.2. Unlike server virtualization, however, the Network Service Appliance will not be running a traditional hypervisor and the VM Orchestration system may not interact with the Network Service Appliance. The NVE on such appliances will need to support a control plane to obtain the necessary information needed to fully participate in an NVO3 Domain.

5.2. Bare Metal Servers

Many data centers will continue to have at least some servers operating as non-virtualized (or "bare metal") machines running a traditional operating system and workload. In such systems, there will be no NVE functionality on the server, and the server will have no knowledge of NVO3 (including whether overlays are even in use). In such environments, the NVE functionality can reside on the first-hop physical switch. In such a case, the network administrator would

(manually) configure the switch to enable the appropriate NVO3 functionality on the switch port connecting the server and associate that port with a specific virtual network. Such configuration would typically be static, since the server is not virtualized, and once configured, is unlikely to change frequently. Consequently, this scenario does not require any protocol or standards work.

5.3. Gateways

Gateways on VNs relay traffic onto and off of a virtual network. Tenant Systems use gateways to reach destinations outside of the local VN. Gateways receive encapsulated traffic from one VN, remove the encapsulation header, and send the native packet out onto the data center network for delivery. Outside traffic enters a VN in a reverse manner.

Gateways can be either virtual (i.e., implemented as a VM) or physical (i.e., as a standalone physical device). For performance reasons, standalone hardware gateways may be desirable in some cases. Such gateways could consist of a simple switch forwarding traffic from a VN onto the local data center network, or could embed router functionality. On such gateways, network interfaces connecting to virtual networks will (at least conceptually) embed NVE (or split-NVE) functionality within them. As in the case with Network Service Appliances, gateways will not support a hypervisor and will need an appropriate control plane protocol to obtain the information needed to provide NVO3 service.

Gateways handle several different use cases. For example, a virtual network could consist of systems supporting overlays together with legacy Tenant Systems that do not. Gateways could be used to connect legacy systems supporting, e.g., L2 VLANs, to specific virtual networks, effectively making them part of the same virtual network. Gateways could also forward traffic between a virtual network and other hosts on the data center network or relay traffic between different VNs. Finally, gateways can provide external connectivity such as Internet or VPN access.

5.4. Distributed Gateways

The relaying of traffic from one VN to another deserves special consideration. The previous section described gateways performing this function. If such gateways are centralized, traffic between TSeS on different VNs can take suboptimal paths, i.e., triangular routing results in paths that always traverse the gateway. As an optimization, individual NVEs can be part of a distributed gateway that performs such relaying, reducing or completely eliminating triangular routing. In a distributed gateway, each ingress NVE can

perform such relaying activity directly, so long as it has access to the policy information needed to determine whether cross-VN communication is allowed. Having individual NVEs be part of a distributed gateway allows them to tunnel traffic directly to the destination NVE without the need to take suboptimal paths.

The NVO3 architecture should [must? or just say it does?] support distributed gateways. Such support requires that NVO3 control protocols include mechanisms for the maintenance and distribution of policy information about what type of cross-VN communication is allowed so that NVEs acting as distributed gateways can tunnel traffic from one VN to another as appropriate.

6. Network Virtualization Authority

Before sending to and receiving traffic from a virtual network, an NVE must obtain the information needed to build its internal forwarding tables and state as listed in Section 4.3. An NVE obtains such information from a Network Virtualization Authority.

The Network Virtualization Authority (NVA) is the entity that provides address mapping and other information to NVEs. NVEs interact with an NVA to obtain any required information they need in order to properly forward traffic on behalf of tenants. The term NVA refers to the overall system, without regards to its scope or how it is implemented.

6.1. How an NVA Obtains Information

There are two primary ways in which an NVA can obtain the address dissemination information it manages. The NVA can obtain information either from the VM orchestration system, or directly from the NVEs themselves.

On virtualized systems, the NVA may be able to obtain the address mapping information associated with VMs from the VM orchestration system itself. If the VM orchestration system contains a master database for all the virtualization information, having the NVA obtain information directly to the orchestration system would be a natural approach. Indeed, the NVA could effectively be co-located with the VM orchestration system itself. In such systems, the VM orchestration system communicates with the NVE indirectly through the hypervisor.

However, as described in Section 4 not all NVEs are associated with hypervisors. In such cases, NVAs cannot leverage VM orchestration protocols to interact with an NVE and will instead need to peer directly with them. By peering directly with an NVE, NVAs can obtain

information about the TSeS connected to that NVE and can distribute information to the NVE about the VNSeS those TSeS are associated with. For example, whenever a Tenant System attaches to an NVE, that NVE would notify the NVA that the TS is now associated with that NVE. Likewise when a TS detaches from an NVE, that NVE would inform the NVA. By communicating directly with NVESeS, both the NVA and the NVE are able to maintain up-to-date information about all active tenants and the NVESeS to which they are attached.

6.2. Internal NVA Architecture

For reliability and fault tolerance reasons, an NVA would be implemented in a distributed or replicated manner without single points of failure. How the NVA is implemented, however, is not important to an NVE so long as the NVA provides a consistent and well-defined interface to the NVE. For example, an NVA could be implemented via database techniques whereby a server stores address mapping information in a traditional (possibly replicated) database. Alternatively, an NVA could be implemented in a distributed fashion using an existing (or modified) routing protocol to maintain and distribute mappings. So long as there is a clear interface between the NVE and NVA, how an NVA is architected and implemented is not important to an NVE.

A number of architectural approaches could be used to implement NVAs themselves. NVAs manage address bindings and distribute them to where they need to go. One approach would be to use BGP (possibly with extensions) and route reflectors. Another approach could use a transaction-based database model with replicated servers. Because the implementation details are local to an NVA, there is no need to pick exactly one solution technology, so long as the external interfaces to the NVESeS (and remote NVAs) are sufficiently well defined to achieve interoperability.

6.3. NVA External Interface

[note: the following section discusses various options that the WG has not yet expressed an opinion on. Discussion is encouraged.]

Conceptually, from the perspective of an NVE, an NVA is a single entity. An NVE interacts with the NVA, and it is the NVA's responsibility for ensuring that interactions between the NVE and NVA result in consistent behavior across the NVA and all other NVESeS using the same NVA. Because an NVA is built from multiple internal components, an NVA will have to ensure that information flows to all internal NVA components appropriately.

One architectural question is how the NVA presents itself to the NVE. For example, an NVA could be required to provide access via a single IP address. If NVEs only have one IP address to interact with, it would be the responsibility of the NVA to handle NVA component failures, e.g., by using a "floating IP address" that migrates among NVA components to ensure that the NVA can always be reached via the one address. Having all NVA accesses through a single IP address, however, adds constraints to implementing robust failover, load balancing, etc.

[Note: the following is a strawman proposal.]

In the NVO3 architecture, an NVA is accessed through one or more IP addresses (ir IP address/port combination). If multiple IP addresses are used, each IP address provides equivalent functionality, meaning that an NVE can use any of the provided addresses to interact with the NVA. Should one address stop working, an NVE is expected to failover to another. While the different addresses result in equivalent functionality, one address may be more respond more quickly than another, e.g., due to network conditions, load on the server, etc.

[Note: should we support the following?] To provide some control over load balancing, NVA addresses may have an associated priority. Addresses are used in order of priority, with no explicit preference among NVA addresses having the same priority. To provide basic load-balancing among NVAs of equal priorities, NVEs use some randomization input to select among equal-priority NVAs. Such a priority scheme facilitates failover and load balancing, for example, allowing a network operator to specify a set of primary and backup NVAs.

[note: should we support the following? It would presumably add considerable complexity to the NVE.] It may be desirable to have individual NVA addresses responsible for a subset of information about an NV Domain. In such a case, NVEs would use different NVA addresses for obtaining or updating information about particular VNs or TS bindings. A key question with such an approach is how information would be partitioned, and how an NVE could determine which address to use to get the information it needs.

Another possibility is to treat the information on which NVA addresses to use as cached (soft-state) information at the NVEs, so that any NVA address can be used to obtain any information, but NVEs are informed of preferences for which addresses to use for particular information on VNs or TS bindings. That preference information would be cached for future use to improve behavior - e.g., if all requests for a specific subset of VNs are forwarded to a specific NVA component, the NVE can optimize future requests within that subset by sending them directly to that NVA component via its address.

7. NVE-to-NVA Protocol

[Note: this and later sections are a bit sketchy and need work. Discussion is encouraged.]

As outlined in Section 4.3, an NVE needs certain information in order to perform its functions. To obtain such information from an NVA, an NVE-to-NVA protocol is needed. The NVE-to-NVA protocol provides two functions. First it allows an NVE to obtain information about the location and status of other TSes with which it needs to communicate. Second, the NVE-to-NVA protocol provides a way for NVEs to provide updates to the NVA about the TSes attached to that NVE (e.g., when a TS attaches or detaches from the NVE), or about communication errors encountered when sending traffic to remote NVEs. For example, an NVE could indicate that a destination it is trying to reach at a destination NVE is unreachable for some reason.

While having a direct NVE-to-NVA protocol might seem straightforward, the existence of existing VM orchestration systems complicates the choices an NVE has for interacting with the NVA.

7.1. NVE-NVA Interaction Models

An NVE interacts with an NVA in at least two (quite different) ways:

- o NVEs supporting VMs and hypervisors can obtain necessary information entirely through the hypervisor-facing side of the NVE. Such an approach is a natural extension to existing VM orchestration systems supporting server virtualization because an existing protocol between the hypervisor and VM Orchestration system already exists and can be leveraged to obtain any needed information. Specifically, VM orchestration systems used to create, terminate and migrate VMs already use well-defined (though typically proprietary) protocols to handle the interactions between the hypervisor and VM orchestration system. For such systems, it is a natural extension to leverage the existing orchestration protocol as a sort of proxy protocol for handling the interactions between an NVE and the NVA. Indeed, existing implementation already do this.
- o Alternatively, an NVE can obtain needed information by interacting directly with an NVA via a protocol operating over the data center underlay network. Such an approach is needed to support NVEs that are not associated with systems performing server virtualization (e.g., as in the case of a standalone gateway) or where the NVE needs to communicate directly with the NVA for other reasons.

[Note: The following paragraph is included to stimulate discussion, and the WG will need to decide what direction it wants to take.]

The WG The NVO3 architecture should support both of the above models, as in practice, it is likely that both models will coexist in practice and be used simultaneously in a deployment. Existing virtualization environments are already using the first model. But they are not sufficient to cover the case of standalone gateways -- such gateways do not support virtualization and do not interface with existing VM orchestration systems. Also, A hybrid approach might be desirable in some cases where the first model is used to obtain the information, but the latter approach is used to validate and further authenticate the information before using it.

7.2. Direct NVE-NVA Protocol

An NVE can interact directly with an NVA via an NVE-to-NVA protocol. Such a protocol can be either independent of the NVA internal protocol, or an extension of it. Using a dedicated protocol provides architectural separation and independence between the NVE and NVA. The NVE and NVA interact in a well-defined way, and changes in the NVA (or NVE) do not need to impact each other. Using a dedicated protocol also ensures that both NVE and NVA implementations can evolve independently and without dependencies on each other. Such independence is important because the upgrade path for NVEs and NVAs is quite different. Upgrading all the NVEs at a site will likely be

more difficult in practice than upgrading NVAs because of their large number - one on each end device. In practice, it is assumed that an NVE will be implemented once, and then (hopefully) not again, whereas an NVA (and its associated protocols) are more likely to evolve over time as experience is gained from usage.

Requirements for a direct NVE-NVA protocol can be found in [I-D.ietf-nvo3-nve-nva-cp-req]

7.3. Propagating Information Between NVEs and NVAs

[Note: This section has been completely redone to move away from the push/pull discussion at an abstract level.]

Information flows between NVEs and NVAs in both directions. The NVA maintains information about all VNs in the NV Domain, so that NVEs do not need to do so themselves. NVEs obtain from the NVA information about where a given remote TS destination resides. NVAs in turn obtain information from NVEs about the individual TSs attached to those NVEs.

While the NVA could push information about every virtual network to every NVE, such an approach scales poorly and is unnecessary. In practice, a given NVE will only need and want to know about VNs to which it is attached. Thus, an NVE should be able to subscribe to updates only for the virtual networks it is interested in receiving updates for. The NVO3 architecture supports a model where an NVE is not required to have full mapping tables for all virtual networks in an NV Domain.

Before sending unicast traffic to a remote TS, an NVE must know where the remote TS currently resides. When a TS attaches to a virtual network, the NVE obtains information about that VN from the NVA. The NVA can provide that information to the NVE at the time the TS attaches to the VN, either because the NVE requests the information when the attach operation occurs, or because the VM orchestration system has initiated the attach operation and provides associated mapping information to the NVE at the same time. A similar process can take place with regards to obtaining necessary information needed for delivery of tenant broadcast or multicast traffic.

There are scenarios where an NVE may wish to query the NVA about individual mappings within an VN. For example, when sending traffic to a remote TS on a remote NVE, that TS may become unavailable (e.g., because it has migrated elsewhere or has been shutdown, in which case the remote NVE may return an error indication). In such situations, the NVE may need to query the NVA to obtain updated mapping information for a specific TS, or verify that the information is

still correct despite the error condition. Note that such a query could also be used by the NVA as an indication that there may be an inconsistency in the network and that it should take steps to verify that the information it has about the current state and location of a specific TS is still correct.

For very large virtual networks, the amount of state an NVE needs to maintain for a given virtual network could be significant. Moreover, an NVE may only be communicating with a small subset of the TSes on such a virtual network. In such cases, the NVE may find it desirable to maintain state only for those destinations it is actively communicating with. In such scenarios, an NVE may not want to maintain full mapping information about all destinations on a VN. Should it then need to communicate with a destination for which it does not have mapping information, however, it will need to be able to query the NVA on demand for the missing information on a per-destination basis.

The NVO3 architecture will need to support a range of operations between the NVE and NVA. Requirements for those operations can be found in [I-D.ietf-nvo3-nve-nva-cp-req].

8. Federated NVAs

An NVA provides service to the set of NVEs in its NV Domain. Each NVA manages network virtualization information for the virtual networks within its NV Domain. An NV domain is administered by a single entity.

In some cases, it will be necessary to expand the scope of a specific VN or even an entire NV domain beyond a single NVA. For example, multiple data centers managed by the same administrator may wish to operate all of its data centers as a single NV region. Such cases are handled by having different NVAs peer with each other to exchange mapping information about specific VNs. NVAs operate in a federated manner with a set of NVAs operating as a loosely-coupled federation of individual NVAs. If a virtual network spans multiple NVAs (e.g., located at different data centers), and an NVE needs to deliver tenant traffic to an NVE at a remote NVA, it still interacts only with its NVA, even when obtaining mappings for NVEs associated with domains at a remote NVA.

Figure 3 shows a scenario where two separate NV Domains (1 and 2) share information about Virtual Network "1217". VM1 and VM2 both connect to the same Virtual Network (1217), even though the two VMs are in separate NV Domains. There are two cases to consider. In the first case, NV Domain B (NVB) does not allow NVE-A to tunnel traffic directly to NVE-B. There could be a number of reasons for this. For

example, NV Domains 1 and 2 may not share a common address space (i.e., require traversal through a NAT device), or for policy reasons, a domain might require that all traffic between separate NV Domains be funneled through a particular device (e.g., a firewall). In such cases, NVA-2 will advertise to NVA-1 that VM1 on virtual network 1217 is available, and direct that traffic between the two nodes go through IP-G. IP-G would then decapsulate received traffic from one NV Domain, translate it appropriately for the other domain and re-encapsulate the packet for delivery.

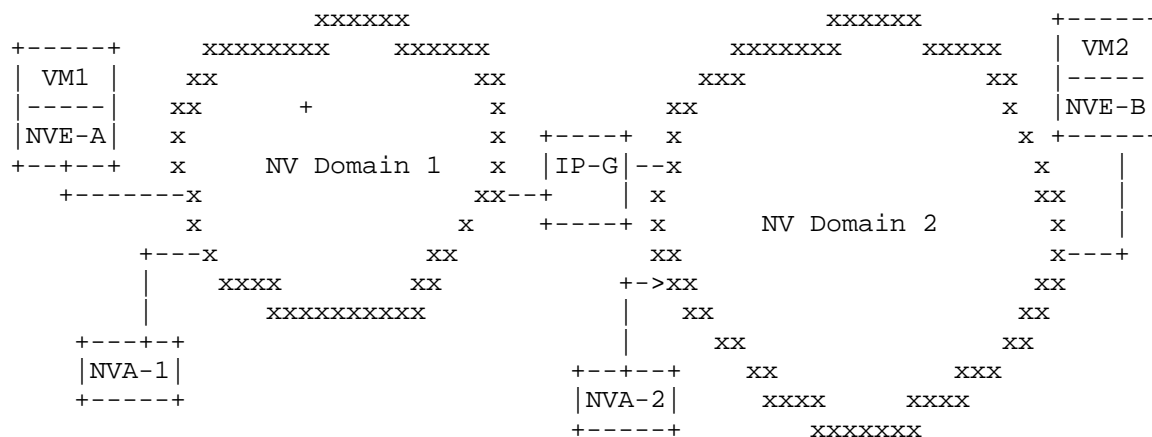


Figure 3: VM1 and VM2 are in different NV Domains.

NVAs at one site share information and interact with NVAs at other sites, but only in a controlled manner. It is expected that policy and access control will be applied at the boundaries between different sites (and NVAs) so as to minimize dependencies on external NVAs that could negatively impact the operation within a site. It is an architectural principle that operations involving NVAs at one site not be immediately impacted by failures or errors at another site. (Of course, communication between NVEs in different NVO3 domains may be impacted by such failures or errors.) It is a strong requirement that an NVA continue to operate properly for local NVEs even if external communication is interrupted (e.g., should communication between a local and remote NVA fail).

At a high level, a federation of interconnected NVAs has some analogies to BGP and Autonomous Systems. Like an Autonomous System, NVAs at one site are managed by a single administrative entity and do not interact with external NVAs except as allowed by policy. Likewise, the interface between NVAs at different sites is well defined, so that the internal details of operations at one site are largely hidden to other sites. Finally, an NVA only peers with other

NVAs that it has a trusted relationship with, i.e., where a virtual network is intended to span multiple NVAs.

[Note: the following are motivations for having a federated NVA model and are intended for discussion. Depending on discussion, these may be removed from future versions of this document.] Reasons for using a federated model include:

- o Provide isolation between NVAs operating at different sites at different geographic locations.
- o Control the quantity and rate of information updates that flow (and must be processed) between different NVAs in different data centers.
- o Control the set of external NVAs (and external sites) a site peers with. A site will only peer with other sites that are cooperating in providing an overlay service.
- o Allow policy to be applied between sites. A site will want to carefully control what information it exports (and to whom) as well as what information it is willing to import (and from whom).
- o Allow different protocols and architectures to be used to for intra- vs. inter-NVA communication. For example, within a single data center, a replicated transaction server using database techniques might be an attractive implementation option for an NVA, and protocols optimized for intra-NVA communication would likely be different from protocols involving inter-NVA communication between different sites.
- o Allow for optimized protocols, rather than using a one-size-fits all approach. Within a data center, networks tend to have lower-latency, higher-speed and higher redundancy when compared with WAN links interconnecting data centers. The design constraints and tradeoffs for a protocol operating within a data center network are different from those operating over WAN links. While a single protocol could be used for both cases, there could be advantages to using different and more specialized protocols for the intra- and inter-NVA case.

8.1. Inter-NVA Peering

To support peering between different NVAs, an inter-NVA protocol is needed. The inter-NVA protocol defines what information is exchanged between NVAs. It is assumed that the protocol will be used to share addressing information between data centers and must scale well over WAN links.

9. Control Protocol Work Areas

The NVO3 architecture consists of two major distinct entities: NVEs and NVAs. In order to provide isolation and independence between these two entities, the NVO3 architecture calls for well defined protocols for interfacing between them. For an individual NVA, the architecture calls for a single conceptual entity, that could be implemented in a distributed or replicated fashion. While the IETF may choose to define one or more specific architectural approaches to building individual NVAs, there is little need for it to pick exactly one approach to the exclusion of others. An NVA for a single domain will likely be deployed as a single vendor product and thus their is little benefit in standardizing the internal structure of an NVA.

Individual NVAs peer with each other in a federated manner. The NVO3 architecture calls for a well-defined interface between NVAs.

Finally, a hypervisor-to-NVE protocol is needed to cover the split-NVE scenario described in Section 4.2.

10. NVO3 Data Plane Encapsulation

When tunneling tenant traffic, NVEs add encapsulation header to the original tenant packet. The exact encapsulation to use for NVO3 does not seem to be critical. The main requirement is that the encapsulation support a Context ID of sufficient size [I-D.ietf-nvo3-dataplane-requirements]. A number of encapsulations already exist that provide a VN Context of sufficient size for NVO3. For example, VXLAN [I-D.mahalingam-dutt-dcops-vxlan] has a 24-bit VXLAN Network Identifier (VNI). NVGRE [I-D.sridharan-virtualization-nvgre] has a 24-bit Tenant Network ID (TNI). MPLS-over-GRE provides a 20-bit label field. While there is widespread recognition that a 12-bit VN Context would be too small (only 4096 distinct values), it is generally agreed that 20 bits (1 million distinct values) and 24 bits (16.8 million distinct values) are sufficient for a wide variety of deployment scenarios.

[Note: the following paragraph is included for WG discussion. Future versions of this document may omit this text.]

While one might argue that a new encapsulation should be defined just for NVO3, no compelling requirements for doing so have been identified yet. Moreover, optimized implementations for existing encapsulations are already starting to become available on the market (i.e., in silicon). If the IETF were to define a new encapsulation format, it would take at least 2 (and likely more) years before optimized implementations of the new format would become available in products. In addition, a new encapsulation format would not likely

displace existing formats, at least not for years. Thus, there seems little reason to define a new encapsulation. However, it does make sense for NVO3 to support multiple encapsulation formats, so as to allow NVEs to use their preferred encapsulations when possible. This implies that the address dissemination protocols must also include an indication of supported encapsulations along with the address mapping details.

11. Operations and Management

The simplicity of operating and debugging overlay networks will be critical for successful deployment. Some architectural choices can facilitate or hinder OAM. Related OAM drafts include [I-D.ashwood-nvo3-operational-requirement].

12. Summary

This document provides a start at a general architecture for overlays in NVO3. The architecture calls for three main areas of protocol work:

1. A hypervisor-to-NVE protocol to support Split NVEs as discussed in Section 4.2.
2. An NVE to NVA protocol for address dissemination.
3. An NVA-to-NVA protocol for exchange of information about specific virtual networks between NVAs.

It should be noted that existing protocols or extensions of existing protocols are applicable.

13. Acknowledgments

Helpful comments and improvements to this document have come from Lizhong Jin, Dennis (Xiaohong) Qin and Lucy Yong.

14. IANA Considerations

This memo includes no request to IANA.

15. Security Considerations

Yep, kind of sparse. But we'll get there eventually. :-)

16. Informative References

[I-D.ashwood-nvo3-operational-requirement]

Ashwood-Smith, P., Iyengar, R., Tsou, T., Sajassi, A., Boucadair, M., Jacquenet, C., and M. Daikoku, "NVO3 Operational Requirements", draft-ashwood-nvo3-operational-requirement-03 (work in progress), July 2013.

[I-D.ietf-nvo3-dataplane-requirements]

Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", draft-ietf-nvo3-dataplane-requirements-01 (work in progress), July 2013.

[I-D.ietf-nvo3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-03 (work in progress), July 2013.

[I-D.ietf-nvo3-nve-nva-cp-req]

Kreeger, L., Dutt, D., Narten, T., and D. Black, "Network Virtualization NVE to NVA Control Protocol Requirements", draft-ietf-nvo3-nve-nva-cp-req-00 (work in progress), July 2013.

[I-D.ietf-nvo3-overlay-problem-statement]

Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-04 (work in progress), July 2013.

[I-D.kreeger-nvo3-hypervisor-nve-cp]

Kreeger, L., Narten, T., and D. Black, "Network Virtualization Hypervisor-to-NVE Overlay Control Protocol Requirements", draft-kreeger-nvo3-hypervisor-nve-cp-01 (work in progress), February 2013.

[I-D.mahalingam-dutt-dcops-vxlan]

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-05 (work in progress), October 2013.

[I-D.sridharan-virtualization-nvgre]

Sridharan, M., Greenberg, A., Wang, Y., Garg, P., Venkataramiah, N., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-03 (work in progress), August 2013.

[IEEE-802.1Q]

IEEE 802.1Q-2011, ., "IEEE standard for local and metropolitan area networks: Media access control (MAC) bridges and virtual bridged local area networks, ", August 2011.

Appendix A. Change Log

A.1. Changes From -00 to -01

1. Editorial and clarity improvements.
2. Replaced "push vs. pull" section with section more focussed on triggers where an event implies or triggers some action.
3. Clarified text on co-located NVE to show how offloading NVE functionality onto adaptors is desirable.
4. Added new section on distributed gateways.
5. Expanded Section on NVA external interface, adding requirement for NVE to support multiple IP NVA addresses.

Authors' Addresses

David Black
EMC

Email: david.black@emc.com

Jon Hudson
Brocade
120 Holger Way
San Jose, CA 95134
USA

Email: jon.hudson@gmail.com

Lawrence Kreeger
Cisco

Email: kreeger@cisco.com

Marc Lasserre
Alcatel-Lucent

Email: marc.lasserre@alcatel-lucent.com

Thomas Narten
IBM

Email: narten@us.ibm.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 4, 2014

P. Quinn
Cisco Systems, Inc.
P. Agarwal
Broadcom
R. Fernando
L. Kreeger
D. Lewis
F. Maino
Cisco Systems, Inc.
M. Smith
N. Yadav
Insieme Networks
October 1, 2013

Generic Protocol Extension for VXLAN
draft-quinn-vxlan-gpe-01.txt

Abstract

This draft describes a mechanism for adding multi-protocol support to Virtual eXtensible Local Area Network (VXLAN). Protocol identification is carried in the VXLAN header and is used to describe the encapsulated payload.

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 4, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	3
2. VXLAN Without Protocol Extension	4
3. Generic Protocol Extension VXLAN (VXLAN-gpe)	5
3.1. VXLAN Header	5
4. Backward Compatibility	6
4.1. VXLAN VTEP to VXLAN-gpe VTEP	6
4.2. VXLAN-gpe VTEP to VXLAN VTEP	6
4.3. IP Type of Service/Traffic Class	6
5. VXLAN-gpe Examples	7
6. Security Considerations	9
7. Acknowledgments	10
8. IANA Considerations	11
9. References	12
9.1. Normative References	12
9.2. Informative References	12
Authors' Addresses	13

1. Introduction

Virtual eXtensible Local Area Network [VXLAN] defines an encapsulation format that encapsulates Ethernet frames in an outer UDP/IP transport. The VXLAN header does not specify the protocol being encapsulated and therefore is currently limited to encapsulating only Ethernet frame payloads. As data centers evolve, the need to carry other protocols in an encapsulated IP packet is required. Rather than defining yet another encapsulation, VXLAN can be extended to indicate the inner protocol, thus broadening the applicability of VXLAN.

This document describes extending VXLAN to support additional payload types beyond Ethernet frames. To support this capability, two elements of the existing VXLAN header are modified.

1. A reserved bit is allocated, and set in the VXLAN header.
2. A 16 bit Protocol Type field is present in the VXLAN header.

These two changes allow for the VXLAN header to support many different types of payloads, all the while maintaining backward compatibility with existing VXLAN deployments.

2. VXLAN Without Protocol Extension

As described in the introduction, the VXLAN header has no protocol identifier that indicates the type of payload being carried by VXLAN. Because of this, VXLAN is limited to an Ethernet payload.

The VXLAN header defines flags (some defined, some reserved), the VXLAN network identifier (VNI) field and several reserved bits. The flags provide flexibility to define how the reserved bits can be used to change the definition of the VXLAN header.

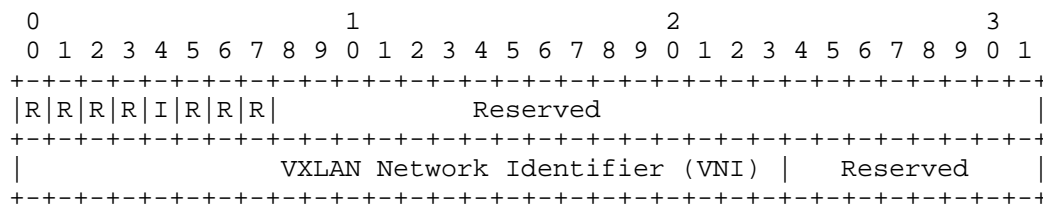


Figure 1: VXLAN Header

3. Generic Protocol Extension VXLAN (VXLAN-gpe)

3.1. VXLAN Header

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

P Bit: Flag bit 5 is defined as the P bit. The P bit **MUST** be set to 1 to indicate the presence of the 16 bit protocol type field in the lower 16 bits of the first word.

P = 0 indicates that the payload **MUST** conform to VXLAN as defined in [VXLAN].

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

Protocol Type Field: The lower 16 bits of the first word are used to carry a protocol type. This protocol type field contains the protocol, as defined in in [RFC1700] and in [ETYPES], of the encapsulated payload packet.

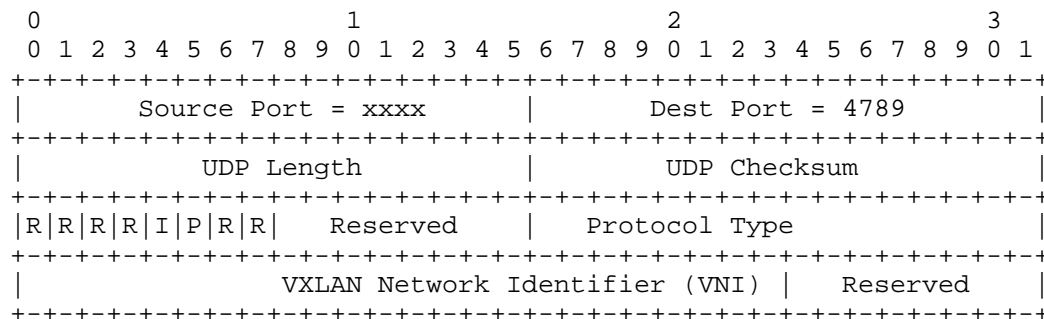


Figure 2: UDP + VXLAN-gpe

4. Backward Compatibility

In order to ensure compatibility with existing VXLAN deployments, P = 0 indicates that the encapsulated payload MUST be Ethernet.

4.1. VXLAN VTEP to VXLAN-gpe VTEP

If a packet is sent from a VXLAN VTEP to a VXLAN-gpe VTEP, the P bit MUST be set to 0, and the remaining fields remain as described in [VXLAN]. The encapsulated payload MUST be Ethernet.

4.2. 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 P bit will be set to 1 and the Protocol Type set to 0x6558. The VXLAN VTEP will ignore the P bit and the Protocol Type, and treat the packet as a VXLAN packet (i.e. the payload is Ethernet)

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

4.3. IP Type of Service/Traffic Class

When a VXLAN-gpe VTEP performs IPv4 encapsulation, the inner IPv4 Type of Service field MAY be copied from the encapsulated packet to the Type of Service or Traffic Class field in the outer IPv4 or IPv6 header respectively.

Similarly, when a VXLAN-gpe VTEP performs IPv6 encapsulation, the inner IPv6 Traffic Class field MAY be copied from the encapsulated packet to the Type of Service or Traffic Class field in the outer IPv4 or IPv6 header respectively.

5. VXLAN-gpe Examples

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

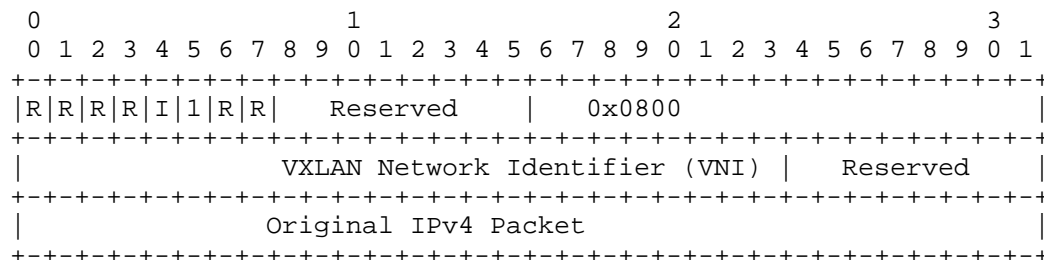


Figure 3: IPv4 and VXLAN-gpe

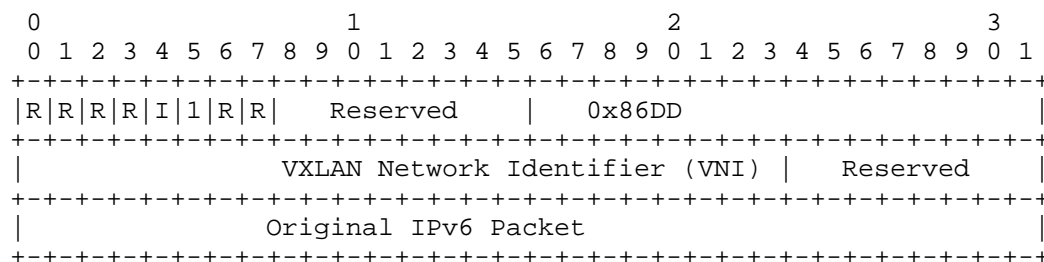


Figure 4: IPv6 and VXLAN-gpe

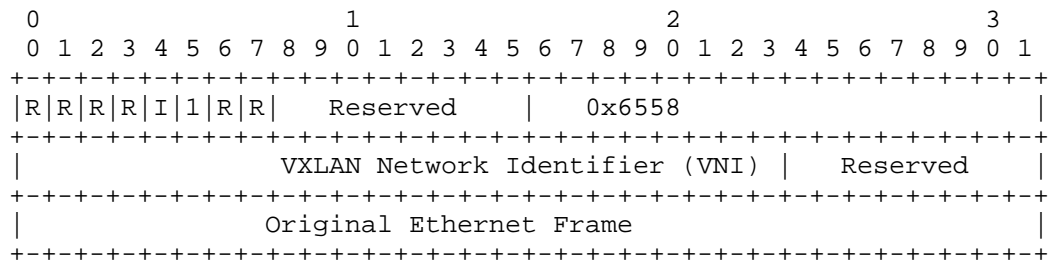


Figure 5: Ethernet and VXLAN-gpe

6. 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.

7. Acknowledgments

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

8. IANA Considerations

This document creates no new requirements on IANA namespaces [RFC5226].

9. References

9.1. 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.

9.2. Informative References

- [ETYPES] The IEEE Registration Authority, "IEEE 802 Numbers", 2012, <<http://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xml>>.
- [RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, October 1994.
- [VXLAN] Dutt, D., Mahalingam, M., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", 2013.

Authors' Addresses

Paul Quinn
Cisco Systems, Inc.

Email: paulq@cisco.com

Puneet Agarwal
Broadcom

Email: pagarwal@broadcom.com

Rex Fernando
Cisco Systems, Inc.

Email: rex@cisco.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: kreeger@cisco.com

Michael Smith
Insieme Networks

Email: michsmit@insiemenetworks.com

Navindra Yadav
Insieme Networks

Email: nyadav@insiemenetworks.com

Network Working Group
Internet-Draft
Expires: March 30, 2014

B. Sarikaya
F. Xia
Huawei USA
September 26, 2013

DHCP Options for Configuring Multicast Addresses in VXLAN
draft-sarikaya-dhc-vxlan-multicast-02.txt

Abstract

This document defines DHCPv4 and DHCPv6 options for assigning multicast addresses for the Tunnel End Point in the Virtual eXtensible Local Area Network (VXLAN) environments. New DHCP options are defined which allow a VXLAN Tunnel End Point to request any source multicast address for the newly created virtual machine, the address of the Rendezvous Point (RP) and possibly address(es) for the virtual machine.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). 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 March 30, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	4
3. Overview of the protocol	4
4. DHCPv6 Options	5
4.1. VXLAN Network Identifier Option	5
4.2. IPv6 multicast address for the VNI Option	6
4.3. IPv6 Rendezvous Point Address Option	7
5. DHCPv4 Options	7
5.1. VXLAN Network Identifier Option	7
5.2. VXLAN Multicast Address Option	8
5.3. VXLAN Rendezvous Point Address Option	8
6. Client Operation	9
7. Server Operation	10
8. Security Considerations	11
9. IANA considerations	11
10. Acknowledgements	11
11. References	11
11.1. Normative References	11
11.2. Informative References	12
Authors' Addresses	13

1. Introduction

Data center networks are being increasingly used by telecom operators as well as by enterprises. Currently these networks are organized as one large Layer 2 network in a single building. In some cases such a network is extended geographically using virtual Local Area Network (VLAN) technologies still as an even larger Layer 2 network connecting the virtual machines (VM), each with its own MAC address.

Another important requirement was growing demand for multitenancy, i.e. multiple tenants each with their own isolated network domain. In a data center hosting multiple tenants, each tenant may independently assign MAC addresses and VLAN IDs and this may lead to potential duplication.

What we need is IP based tunneling scheme based overlay network called Virtual eXtensible Local Area Network (VXLAN). VXLAN overlays a Layer 2 network over a Layer 3 network. Each overlay, identified by the VXLAN Network Identifier (VNI). This allows up to 16M VXLAN segments to coexist within the same administrative domain [I-D.mahalingam-dutt-dcops-vxlan]. In VXLAN, each MAC frame is transmitted after encapsulation, i.e. an outer Ethernet header, an IPv4/IPv6 header, UDP header and VXLAN header are added. Outer Ethernet header indicates an IPv4 or IPv6 payload. VXLAN header contains 24-bit VNI.

VXLAN tunnel end point (VTEP) is the hypervisor on the server which houses the VM. VXLAN encapsulation is only known to the VTEP, the VM never sees it. Also the tunneling is stateless, each MAC frame is encapsulated independent on any other MAC frame.

Instead of using UDP header, Generic Routing Encapsulation (GRE) encapsulation can be used. A 24-bit Virtual Subnet Identifier (VSID) is placed in the GRE key field. The resulting encapsulation is called Network Virtualization using Generic Routing Encapsulation (NVGRE) [I-D.sridharan-virtualization-nvgre]. Note that VSID is similar to VNI. Although VXLAN terminology is used throughout, the protocol defined in this document applies to VXLAN as well as NVGRE.

In this document, we develop a protocol to assign multicast addresses to the VXLAN tunnel end points using Dynamic Host Configuration Protocol (DHCP). Multicast communication in VXLAN is used for sending broadcast MAC frames, e.g. the Address Resolution Protocol (ARP) broadcast frame. Multicast communication can also be used to transmit multicast frames and unknown MAC destination frames.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. The terminology in this document is based on the definitions in [I-D.mahalingam-dutt-dcops-vxlan]

3. Overview of the protocol

Multicast addresses to be assigned by the DHCP server are administratively scoped multicast addresses, in IPv4 [RFC2365] and in IPv6 [RFC4291]. The steps involved in the protocol are explained below for IPv4:

Creation of a VM

In this step, VTEP receives a request from the Management Node to create a Virtual Machine with a VXLAN Network Identifier.

DHCP Operation

VTEP starts DHCP state machine by sending DHCPDISCOVER message to the default router, e.g. the Top of Rack (ToR) switch. ToR switch could be DHCP server or most possibly DHCP relay with DHCP server located upstream. VTEP MUST include the VXLAN Multicast Address and VXLAN Rendezvous Point Address options defined in this document. VTEP sends the VXLAN Network Identifier in the newly defined VNI DHCP Option. DHCP server replies with DHCPOFFER message. DHCP server sends administratively scope IPv4 multicast address and RP address to VTEP. VTEP checks this message and if it sees the options it requested, DHCP server is confirmed to support the multicast address options. DHCPREQUEST message from VTEP and DHCPACK message from DHCP server complete DHCP message exchange.

VTEP as Multicast Source

After receiving the required information, the VTEP as multicast source communicates with the Rendezvous Point in order to build the multicast tree.

VTEP as Listener

After receiving the required information, the VTEP as listener communicates with the edge router by sending MLD Report to join the multicast group.

IPv6 operation is slightly different:

Creation of a VM

In this step, VTEP receives a request from the Management Node to create a Virtual Machine with a VXLAN Network Identifier.

DHCP Operation

VTEP starts DHCP state machine by sending DHCPv6 Solicit message to the default router, e.g. the Top of Rack (ToR) switch. ToR switch could be DHCP server or most possibly DHCP relay with DHCP server located upstream. VTEP MUST include the options defined in this document. DHCP server replies with DHCPv6 Advertise message. VTEP checks this message and if it sees the options it requested, DHCP server is confirmed to support multicast address options. DHCPv6 Request message from VTEP and DHCPv6 Reply message from DHCPv6 server complete DHCP message exchange.

VTEP as Multicast Source

After receiving the required information, the VTEP as multicast source communicates with the Rendezvous Point in order to build the multicast tree.

VTEP as Listener

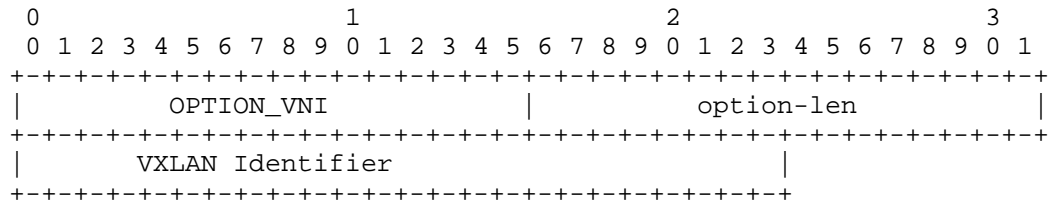
After receiving the required information, the VTEP as listener communicates with the edge router by sending MLD Report to join the multicast group.

4. DHCPv6 Options

4.1. VXLAN Network Identifier Option

Different VXLAN Network Identifiers (VNI) need different multicast groups, and even rendezvous point addresses (for load balancing). At the same time, different VNIs need different address spaces for VM, that is, two VMs belongs to different VNIs probably have the same IP address.

Because of the reasons stated above, a DHCP VNI Option is defined as follows.



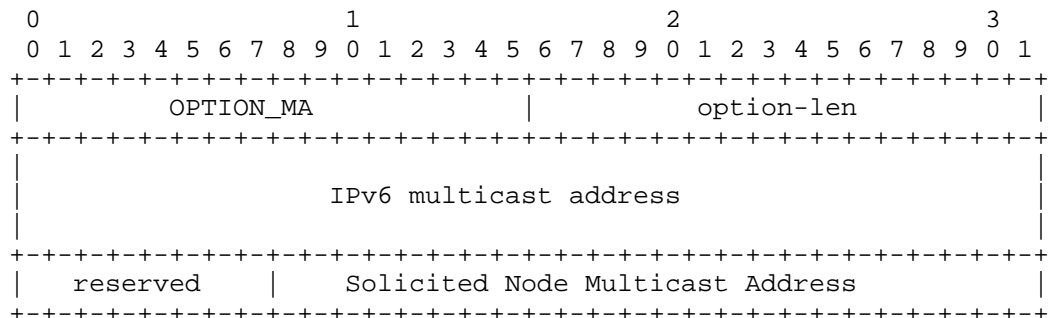
option-code OPTION_VNI (TBD).

option-len 7.

VXLAN Network Identifier 3.

4.2. IPv6 multicast address for the VNI Option

The option allows the VTEP to send the VNI and solicited-node multicast address to DHCP server and receive administratively scoped IPv6 multicast address.



option-code OPTION_MA (TBD).

option-len 24.

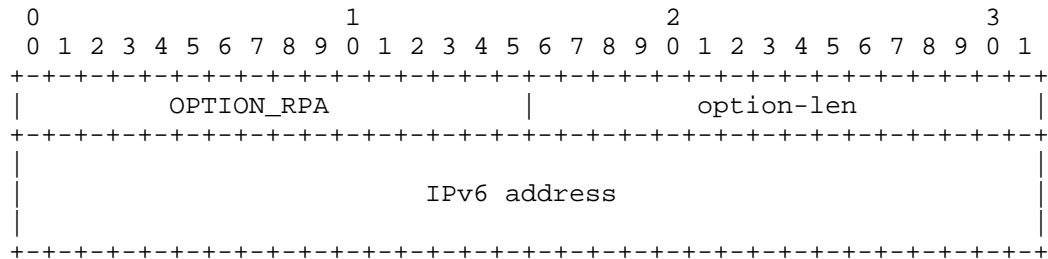
IPv6 multicast address An IPv6 address.

reserved must be set to zero

Solicited Node Multicast Address as in RFC 4861.

4.3. IPv6 Rendezvous Point Address Option

The option allows the VTEP to receive RP address for Any Source Multicast group from DHCP server.



option-code OPTION_RPA (TBD).

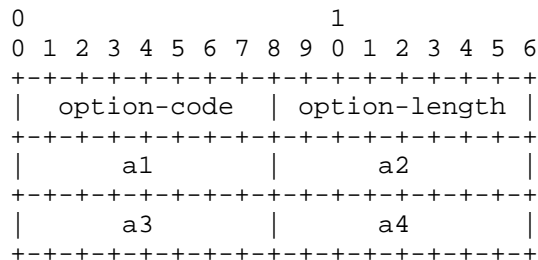
option-len 20.

IPv6 multicast address An IPv6 address

5. DHCPv4 Options

5.1. VXLAN Network Identifier Option

The option allows the VTEP to send the VNI to DHCP server.



Option-code
VXLAN Network Identifier Option (TDB)

Option-len
4.

a1-a4

VTEP as DHCP Client sets a1-a3 to VNI and a4 to zero.

5.2. VXLAN Multicast Address Option

The option allows the VTEP to send the VNI DHCP server and receive administratively scoped IPv4 multicast address.

```

0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
+---+---+---+---+---+---+---+---+
| option-code | option-length |
+---+---+---+---+---+---+---+---+
|      a1      |      a2      |
+---+---+---+---+---+---+---+---+
|      a3      |      a4      |
+---+---+---+---+---+---+---+---+

```

Option-code

VXLAN Multicast Address Option (TDB)

Option-len

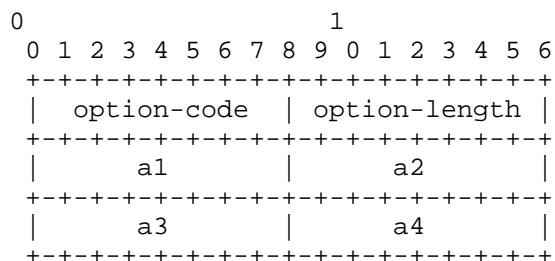
4.

a1-a4

VTEP as DHCP Client sets a1-a4 to zero, DHCP server sets a1-a4 to the multicast address.

5.3. VXLAN Rendezvous Point Address Option

This option is used to receive VXLAN Rendezvous Point address from DHCP server.



Option-code
VXLAN Rendezvous Point Address Option (TBD)

Option-len
4.

a1-a4
VXLAN Rendezvous Point Address an IPv4 address

6. Client Operation

In DHCPv4, the client, VTEP MUST set 'htype' and 'chaddr' fields to specify the client link-layer address type and the link-layer address. The client must set the hardware type, 'htype' to 1 for Ethernet [RFC1700] and 'chaddr' is set to the MAC address of the virtual machine.

The client MUST set VXLAN Multicast Address Option to zero. The client MUST set VXLAN Rendezvous Point Address Option to zero. The client MUST set VXLAN Network Identifier Option to the VXLAN network identifier assigned to the virtual machine.

In DHCPv6, the client MUST use OPTION_CLIENT_LINKLAYER_ADDR defined in [RFC6939] to send the MAC address. In this option, link-layer type MUST be set to 1 for Ethernet and link-layer address MUST be set to the MAC address of VM. Note that in [RFC6939], OPTION_CLIENT_LINKLAYER_ADDR is defined to be used in Relay-Forward DHCP message. In this document this option MUST be sent in DHCPv6 Solicit message.

The client MUST set IPv6 VNI Option OPTION_VNI to the VXLAN network identifier assigned to the virtual machine.

The Client MUST set IPv6 multicast address for the VNI Option's multicast address field to zero.

The client MUST set IPv6 Rendezvous Point Address Option's IPv6 multicast address field to zero.

The client MUST set Solicited Node Multicast Address to zero if the neighbor discovery message is sent to all-nodes multicast address. The client MUST set Solicited Node Multicast Address to the low-order 24 bits of an address of the destination if the neighbor discovery message is sent to the solicited-node multicast address.

7. Server Operation

If DHCPv4 server is configured to support VXLAN multicast address assignments, it SHOULD look for VXLAN Multicast Address Option and VXLAN Rendezvous Point Address Option in DHCPDISCOVER message. The server MUST return in VXLAN Multicast Address Option's a1-a4 an organization-local scope IPv4 multicast address (239.192.0.0/14) [RFC2365]. The server MUST use the VNI value for obtaining the organization-local scope IPv4 multicast address. VNI value is directly copied to 239.192.0.0/14 if the first 6 bits are zero, i.e. no overflow ranges need to be used. Otherwise, either of 239.0.0.0/10, 239.64.0.0/10 and 239.128.0.0/10 overflow ranges SHOULD be used. Note that these ranges can accomodate the VNI in its entirety.

The server MUST set VXLAN Rendezvous Point Address Option's VXLAN Rendezvous Point Address field to IPv4 unicast address of the Rendezvous Point for the any source multicast Rendezvous Point router. How this assignment is done is out of scope.

If DHCPv6 server is configured to support VXLAN multicast address assignments it SHOULD look for IPv6 multicast address for the VNI Option and IPv6 Rendezvous Point Address Option in DHCPv6 Solicit message. The server MUST return in IPv6 multicast address field an Admin-Local scope IPv6 multicast address (FF04/16) by copying the VNI of the virtual machine to the least significant 24 bits of the group ID field and setting all other bits to zero if Solicited Node Multicast Address field received from the client was set to zero. Otherwise the Solicited Node Multicast Address field is copied to bits 47-24 of the group ID field and all leading bits are set to zero.

The server MUST assign IPv6 Rendezvous Point Address Option's IPv6 address field to the Rendezvous Point router's address in charge of this multicast group. The unicast address MUST BE assigned according to the rules defined in [RFC3956].

8. Security Considerations

The security considerations in [RFC2131], [RFC2132] and [RFC3315] apply. Special considerations in [I-D.mahalingam-dutt-dcops-vxlan] are also applicable.

9. IANA considerations

IANA is requested to assign the OPTION_VNI and OPTION_MA and OPTION_RPA and VXLAN Network Identifier and VXLAN Multicast Address and VXLAN Rendezvous Point Address Option Codes in the registry maintained for DHCPv4 and DHCPv6.

10. Acknowledgements

The authors are grateful to Bernie Volz for providing comments that helped us improve the document.

11. References

11.1. Normative References

- [RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", RFC 1700, October 1994.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, November 2004.
- [RFC2365] Meyer, D., "Administratively Scoped IP Multicast", BCP 23, RFC 2365, July 1998.

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC6939] Halwasia, G., Bhandari, S., and W. Dec, "Client Link-Layer Address Option in DHCPv6", RFC 6939, May 2013.

11.2. Informative References

- [I-D.mahalingam-dutt-dcops-vxlan]
Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-04 (work in progress), May 2013.
- [I-D.sridharan-virtualization-nvgre]
Sridharan, M., Greenberg, A., Wang, Y., Garg, P., Venkataramiah, N., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-03 (work in progress), August 2013.

Authors' Addresses

Behcet Sarikaya
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075

Phone: +1 972-509-5599
Email: sarikaya@ieee.org

Frank Xia
Huawei USA
Nanjing, China

Phone: +1 972-509-5599
Email: xiayangsong@huawei.com

Network working group
Internet Draft
Category: Standard Track

L. Yong
X. Xu
Huawei

Expires: April 2014

October 17, 2013

NVGRE and VXLAN Encapsulation Extension for L3 Overlay
draft-yong-l3vpn-nvgre-vxlan-encap-03

Status of this Memo

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

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

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

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

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

This Internet-Draft will expire on April 17, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract

Both NVGRE and VXLAN encapsulations were originally designed for L2 overlay only. This draft proposes the enhancement on both to support L3 overlay as well. The proposed method completely decouples the L3 overlay from the L2 overlay in terms of encoding schema and data processing.

Conventions 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 RFC-2119 [RFC2119].

Table of Contents

1. Introduction.....	3
2. NVGRE Encapsulation Extension for L3 Overlay.....	3
3. VXLAN Encapsulation Extension for L3 Overlay.....	3
4. Security Considerations.....	4
5. IANA Considerations.....	5
6. References.....	5
6.1. Normative References.....	5
6.2. Informative References.....	5

1. Introduction

Network Virtualization Overlay [NVO3FRWK] explicitly states that both L2 and L3 overlays are needed in practice. However both NVGRE encapsulation [NVGRE] and VXLAN encapsulation [VXLAN] were originally designed for L2 overlay only.

This document proposes enhancements to NVGRE and VXLAN encapsulations to allow the same data encapsulation semantics for both L2 overlay and L3 overlay. The benefits of this approach are generalizing the data encapsulation semantics for overlay technologies, maintaining L3 overlay natively, and decoupling it from L2 overlay completely.

2. NVGRE Encapsulation Extension for L3 Overlay

NVGER [NVGRE] leverages the GRE protocol [RFC2890] and specifies that the protocol type field in the GRE header MUST be filled with the value of 0x6558, which means for Transparent Ethernet.

This document proposes the protocol type field to be filled with the value of 0x6558, 0x0800(IPv4), or 0x86dd(IPv6). The value of 0x0800 and 0x86dd means that the payload is IP. The value 0x6558 MUST be used if the inner header is an Ethernet header. When NVGRE encapsulation is used for L3 overlay, it MUST use the value of 0x0800 or 0x86dd in the protocol type field and MUST encode an IPv4 or IPv6 header as the inner header. Other fields in the outer header and the GRE header remain the same.

To support backward compatibility, when the remote tunnel end point only support the NVGRE described in [NVGRE], the tunnel end point that supports NVGRE described in this document MUST only encapsulate L2 packets. This capability can be either manually configured or be dynamically informed. How tunnel end points inform each other the encapsulation capabilities is beyond the scope of this document. Note that a tunnel may have more than two end points.

3. VXLAN Encapsulation Extension for L3 Overlay

This document proposes adding a protocol type field in the VXLAN header as shown below. It takes 16 bits from the reserved 24 bits as the protocol type field. The remained 8 reserved bits MUST be filled with zero. For L2 overlay encapsulation, the protocol type field MUST be filled with the value of 0x6558 and inner header MUST be an Ethernet header. For L3 overlay encapsulation, the protocol type

field MUST be filled with the value of 0x0800(IPv4) or 0x86dd(IPv6), and inner header MUST be an IPv4 or IPv6 header. Other fields in the outer header and VXLAN header remain the same.

```

      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:
      As described in VXLAN [VXLAN]
Outer IP Header:
      As described in VXLAN [VXLAN]
Outer UDP Header:
      As described in VXLAN [VXLAN]
VXLAN Header:
+++++
|R|R|R|R|I|R|R|R|   Reserved   |Prot. Type=0x6558/0x0800/0x86dd|
+++++
|               VXLAN Network Identifier (VNI) |   Reserved   |
+++++
Inner Header:
+++++
|               Ethernet header or IP Header               ~
+++++

```

To be backward compatible with the existing VXLAN encapsulation [VXLAN], the value 0x0000 in the Protocol Type field MUST be treated as Ethernet payload too. When the end points of a tunnel support different VXLAN formats, i.e. one, say A, supports old VXLAN format and another, say B, supports the new format described in this document, B MUST only encapsulate L2 packets and set value 0x0000 in the protocol type field. This capability can be either manually configured at B or be dynamically informed. How tunnel end points inform each other the encapsulation capabilities is beyond the scope of this document. Note that a tunnel may have more than two end points.

Having protocol type field in the VXLAN header enables other overlay payload type beside L2 and L3 overlays. The application for other payload type is for future study.

4. Security Considerations

The mechanism proposed in this document does not add any additional security concern beside what has been described in the NVGRE [NVGRE] and VXLAN [VXLAN].

5. IANA Considerations

The document does not require any IANA action.

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC2119, March 1997.
- [RFC2890] Dommety, G., "Key and Sequence Number Extension to GRE", RFC2890, September 2000

6.2. Informative References

- [NVO3FRWK] Lasserre, M., et al, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-03.txt, work in progress.
- [NVGRE] Sridharan, M., et al, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-03, work in progress
- [VXLAN] Mahalingam, M., Dutt, D., etc, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-05.txt, work in progress

Authors' Addresses

Lucy Yong
Huawei Technologies, USA

Phone: 918-808-1918
Email: lucy.yong@huawei.com

Xiaohu Xu
Huawei Technologies,
Beijing, China

Phone: +86-10-60610041
Email: xuxiaohu@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

E. Crabbe, Ed.
Google
L. Yong, Ed.
Huawei USA
X. Xu, Ed.
Huawei Technologies
October 21, 2013

Generic UDP Encapsulation for IP Tunneling
draft-yong-tsvwg-gre-in-udp-encap-02

Abstract

This document describes a method of encapsulating arbitrary protocols within GRE and UDP headers. In this encapsulation, the source UDP port may be used as an entropy field for purposes of loadbalancing while the payload protocol may be identified by the GRE Protocol Type.

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 24, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. Procedures	3
4. Encapsulation Considerations	6
5. Backward Compatibility	7
6. IANA Considerations	7
7. Security Considerations	7
7.1. Vulnerability	7
8. Acknowledgements	8
9. Contributing Authors	8
10. References	9
10.1. Normative References	9
10.2. Informative References	10
Authors' Addresses	10

1. Introduction

Load balancing, or more specifically, statistical multiplexing of traffic using Equal Cost Multi-Path (ECMP) and/or Link Aggregation Groups (LAGs) in IP networks is a widely used technique for creating higher capacity networks out of lower capacity links. Most existing routers in IP networks are already capable of distributing IP traffic flows over ECMP paths and/or LAGs on the basis of a hash function performed on flow invariant fields in IP packet headers and their payload protocol headers. Specifically, when the IP payload is a User Datagram Protocol (UDP)[RFC0768] or Transmission Control Protocol (TCP) packet, router hash functions frequently operate on the five-tuple of the source IP address, the destination IP address, the source port, the destination port, and the protocol/next-header

Several tunneling techniques are in common use in IP networks, such as Generic Routing Encapsulation (GRE) [RFC2784], MPLS [RFC4023] and L2TPv3 [RFC3931]. GRE is an increasingly popular encapsulation choice, especially in environments where MPLS is unavailable or unnecessary. Unfortunately, use of common GRE endpoints may reduce the entropy available for use in load balancing, especially in environments where the GRE Key field [RFC2890] is not readily available for use as entropy in forwarding decisions.

This document defines a generic GRE-in-UDP encapsulation for tunneling arbitrary network protocol payloads across an IP network environment where ECMP or LAGs are used. The GRE header provides payload protocol de-multiplexing by way of it's protocol type field [RFC2784] while the UDP header provides additional entropy by way of it's source port.

This encapsulation method requires no changes to the transit IP network. Hash functions in most existing IP routers may utilize and benefit from the use of a GRE-in-UDP tunnel without needing any change or upgrade to their ECMP implementations. The encapsulation mechanism is applicable to a variety of IP networks including Data Center and wide area networks.

2. Terminology

The terms defined in [RFC0768] are used in this document.

3. Procedures

When a tunnel ingress device conforming to this document receives a packet, the ingress MUST encapsulate the packet in UDP and GRE headers and set the destination port of the UDP header to [TBD] Section 6. The ingress device must also insert the payload protocol type in the GRE Protocol Type field. The ingress device SHOULD set the UDP source port based on flow invariant fields from the payload header, otherwise it should be set to a randomly selected constant value, e.g. zero, to avoid packet flow reordering. How a tunnel ingress generates entropy from the payload is outside the scope of this document. The tunnel ingress MUST encode its own IP address as the source IP address and the egress tunnel endpoint IP address. The TTL field in the IP header must be set to a value appropriate for delivery of the encapsulated packet to the tunnel egress endpoint.

When the tunnel egress receives a packet, it must remove the outer UDP and GRE headers. Section 5 describes the error handling when this entity is not instantiated at the tunnel egress.

To simplify packet processing at the tunnel egress, packets destined to this assigned UDP destination port [TBD] SHOULD have their UDP checksum and Sequence flags set to zero because the egress tunnel only needs to identify this protocol. Although IPv6 [RFC2460] restricts the processing a packet with the UDP checksum of zero, [RFC6935] and [RFC6936] relax this constraint to allow the zero UDP checksum.

The tunnel ingress may set the GRE Key Present, Sequence Number Present, and Checksum Present bits and associated fields in the GRE header defined by [RFC2784] and [RFC2890].

In addition IPv6 nodes MUST conform to the following:

1. the IPv6 tunnel ingress and egress SHOULD follow the node requirements specified in Section 4 of [RFC6936] and the usage requirements specified in Section 5 of [RFC6936]
2. IPv6 transit nodes SHOULD follow the requirements 9, 10, 11 specified in Section 5 of [RFC6936].

The format of the GRE-in-UDP encapsulation for both IPv4 and IPv6 outer headers is shown in the following figures:

```

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

```

IPv4 Header:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|Version|  IHL  |Type of Service|          Total Length          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          Identification          |Flags|      Fragment Offset      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  Time to Live |Protocol=17[UDP]|          Header Checksum          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Source IPv4 Address                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Destination IPv4 Address                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

UDP Header:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|          Source Port = XXXX          |          Dest Port = TBD          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          UDP Length          |          UDP Checksum          |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

GRE Header:

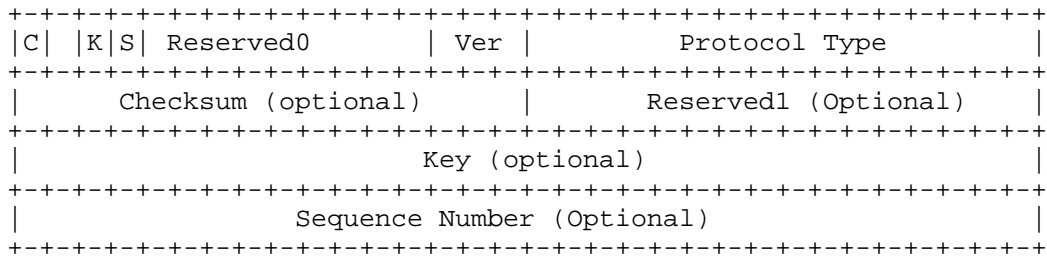
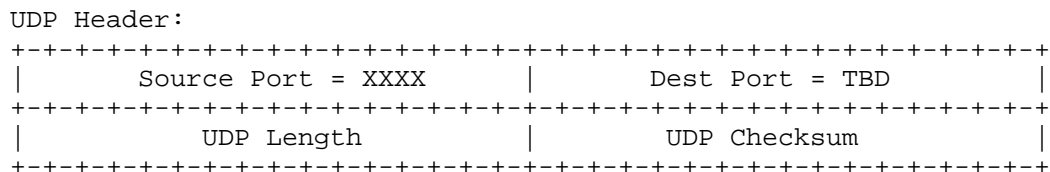
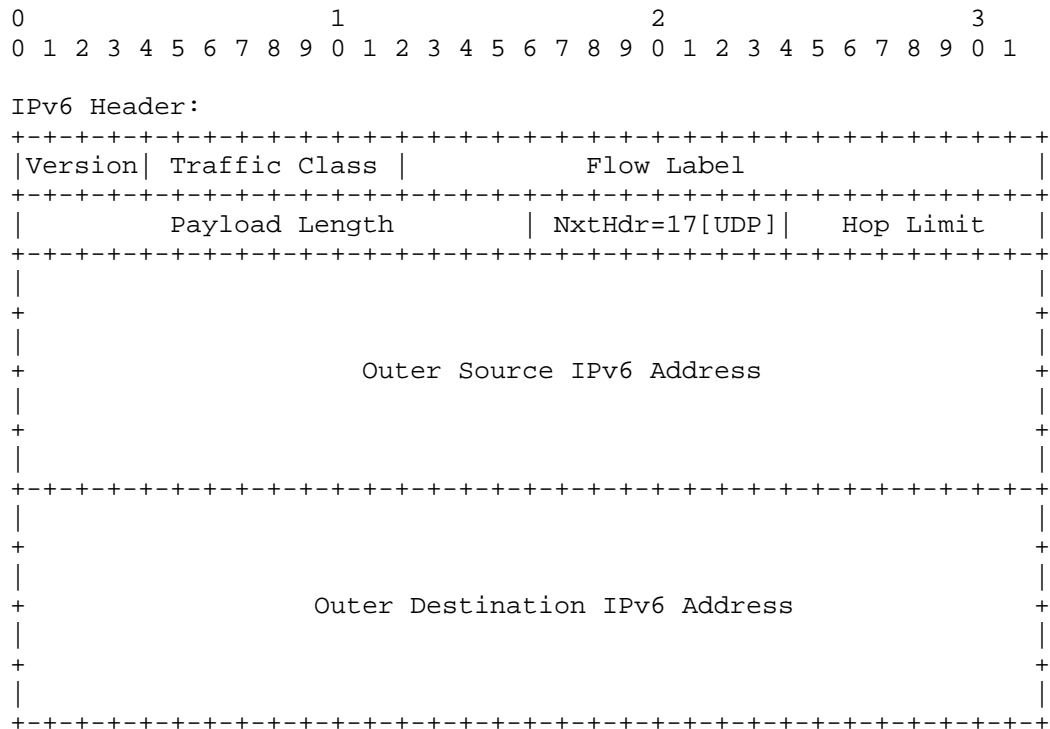


Figure 1: UDP+GRE IPv4 headers



GRE Header:

C	K S	Reserved0		Ver		Protocol Type	
Checksum (optional)				Reserved1 (Optional)			
Key (optional)							
Sequence Number (Optional)							

Figure 2: UDP+GRE IPv6 headers

The total overhead increase for a UDP+GRE tunnel without use of optional GRE fields, representing the lowest total overhead increase, is 32 bytes in the case of IPv4 and 52 bytes in the case of IPv6. The total overhead increase for a UDP+GRE tunnel with use of GRE Key, Sequence and Checksum Fields, representing the highest total overhead increase, is 44 bytes in the case of IPv4 and 64 bytes in the case of IPv6.

4. Encapsulation Considerations

GRE-in-UDP encapsulation allows the tunneled traffic to be unicast, broadcast, or multicast traffic. Entropy may be generated from the header of tunneled unicast or broadcast/multicast packets at tunnel ingress. The mapping mechanism between the tunneled multicast traffic and the multicast capability in the IP network is transparent and independent to the encapsulation and is outside the scope of this document.

If tunnel ingress must perform fragmentation on a packet before encapsulation, it MUST use the same source UDP port for all packet fragments. This ensures that the transit routers will forward the packet fragments on the same path. GRE-in-UDP encapsulation introduces some overhead as mentioned in section 3, which reduces the effective Maximum Transmission Unit (MTU) size. An operator should factor in this addition overhead bytes when considering an MTU size for the payload to reduce the likelihood of fragmentation.

To ensure the tunneled traffic gets the same treatment over the IP network, prior to the encapsulation process, tunnel ingress should process the payload to get the proper parameters to fill into the IP header such as DiffServ [[RFC2983]]. Tunnel end points that support ECN MUST use the method described in [RFC6040] for ECN marking propagation. This process is outside of the scope of this document.

Note that the IPv6 header [RFC2460] contains a flow label field that may be used for load balancing in an IPv6 network [RFC6438]. Thus in an IPv6 network, either GRE-in-UDP or flow labels may be used in order to improve load balancing performance. Use of GRE-in-UDP encapsulation provides a unified hardware implementation for load balancing in an IP network independent of the IP version(s) in use.

5. Backward Compatibility

It is assumed that tunnel ingress routers must be upgraded in order to support the encapsulations described in this document.

No change is required at transit routers to support forwarding of the encapsulation described in this document.

If a router that is intended for use as a tunnel egress does not support the GRE-in-UDP encapsulation described in this document, it will not be listening on destination port [TBD]. In these cases, the router will conform to normal UDP processing and respond to the tunnel ingress with an ICMP message indicating "port unreachable" according to [RFC0792]. Upon receiving this ICMP message, the tunnel ingress MUST NOT continue to use GRE-in-UDP encapsulation toward this tunnel egress without management intervention.

6. IANA Considerations

IANA is requested to make the following allocation: Service Name: GRE-in-UDP Transport Protocol(s): UDP Assignee: IESG iesg@ietf.org Contact: IETF Chair chair@ietf.org Description: GRE-in-UDP Encapsulation Reference: [This.I-D] Port Number: TBD Service Code: N/A A Known Unauthorized Uses: N/A Assignment Notes: N/A

7. Security Considerations

7.1. Vulnerability

Neither UDP nor GRE encapsulation effects security for the payload protocol. When using GRE-in-UDP, Network Security in a network is similar to that of a network using GRE.

Use of ICMP for signaling of the GRE-in-UDP encapsulation capability adds a security concern. Tunnel ingress devices may want to validate the origin of ICMP Port Unreachable messages before taking action. The mechanism for performing this validation is out of the scope of this document.

In an instance where the UDP src port is not set based on the flow invariant fields from the payload header, a random port SHOULD be

selected in order to minimize the vulnerability to off-path attacks. [RFC6056] How the src port randomization occurs is outside scope of this document.

8. Acknowledgements

The Authors would like to thank Vivek Kumar, Ron Bonica, Joe Touch, Ruediger Geib, Gorrry Fairhurst, and David Black for their review and valuable input on this draft.

9. Contributing Authors

The following people all contributed significantly to this document and are listed below in alphabetical order:

John E. Drake
Juniper Networks

Email: jdrake@juniper.net

Adrian Farrel
Juniper Networks

Email: adrian@olddog.co.uk

Vishwas Manral
Hewlett-Packard Corp.
3000 Hanover St, Palo Alto.

Email: vishwas.manral@hp.com

Carlos Pignataro
Cisco Systems
7200-12 Kit Creek Road
Research Triangle Park, NC 27709 USA

Email: cpignata@cisco.com

Yongbing Fan
China Telecom
Guangzhou, China.
Phone: +86 20 38639121

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.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [RFC2890] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, September 2000.
- [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.
- [RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.
- [RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.
- [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.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, November 2011.
- [RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, April 2013.
- [RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, April 2013.

10.2. Informative References

- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, March 2005.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, April 2007.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, November 2012.

Authors' Addresses

Edward Crabbe (editor)
Google
1600 Amphitheatre Parkway
Mountain View, CA 94102
US

Email: edward.crabbe@gmail.com

Lucy Yong (editor)
Huawei USA
5340 Legacy Drive
San Jose, TX 75025
US

Email: lucy.yong@huawei.com

Xiaohu Xu (editor)
Huawei Technologies
Beijing
China

Email: xuxiaohu@huawei.com