

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 16, 2014

P. Ashwood-Smith  
Huawei Technologies  
R. Iyengar  
T. Tsou  
Huawei Technologies USA  
A. Sajassi  
Cisco Technologies  
M. Boucadair  
C. Jacquenet  
France Telecom  
M. Daikoku  
KDDI corporation  
July 15, 2013

NVO3 Operational Requirements  
draft-ashwood-nvo3-operational-requirement-03

Abstract

This document provides framework and requirements for Network Virtualization over Layer 3 (NVO3) Operations, Administration, and Maintenance (OAM). This document for the most part gathers requirements from existing IETF drafts and RFCs which have already extensively studied this subject for different data planes and layering. As a result this draft is high level and broad. We begin to ask which are truly required for NVO3 and expect the list to be narrowed by the working group as subsequent versions of this draft are created.

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 16, 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
1.1. OSI Definitions of OAM . . . . .	3
1.2. Requirements Language . . . . .	5
1.3. Relationship with Other OAM Work . . . . .	5
2. Terminology . . . . .	6
3. NVO3 Reference Model . . . . .	6
4. OAM Framework for NVO3 . . . . .	7
4.1. OAM Layering . . . . .	7
4.2. OAM Domains . . . . .	8
5. NVO3 OAM Requirements . . . . .	9
5.1. Discovery . . . . .	9
5.2. Connectivity Fault Management . . . . .	9
5.2.1. Connectivity Fault Detection . . . . .	9
5.2.2. Connectivity Fault Verification . . . . .	9
5.2.3. Connectivity Fault localization . . . . .	10
5.2.4. Connectivity Fault Notification and Alarm Suppression . . . . .	10
5.3. Frame Loss . . . . .	10
5.4. Frame Delay . . . . .	10
5.5. Frame Delay Variation . . . . .	10
5.6. Frame Throughput . . . . .	10
5.7. Frame Discard . . . . .	10
5.8. Availability . . . . .	11
5.9. Data Path Forwarding . . . . .	11
5.10. Scalability . . . . .	11
5.11. Extensibility . . . . .	11
5.12. Security . . . . .	11
5.13. Transport Independence . . . . .	12
5.14. Application Independence . . . . .	12
5.15. Prioritization . . . . .	12
6. Items for Further Discussion . . . . .	12
7. IANA Considerations . . . . .	14

8. Security Considerations . . . . .	14
9. Acknowledgements . . . . .	14
10. References . . . . .	14
10.1. Normative References . . . . .	14
10.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

This document provides framework and requirements for Network virtualization over Layer 3(NVO3) Operation, Administration, and Maintenance (OAM). Given that this OAM subject is far from new and has been under extensive investigation by various IETF working groups (and several other standards bodies) for many years, this document draws from existing work, starting with [RFC6136]. As a result, sections of [RFC6136] have been reused with minor changes with the permission of the authors.

NVO3 OAM requirements are expected to be a subset of IETF/IEEE etc. work done so far; however, we begin with a full set of requirements and expect to prune them through several iterations of this document.

### 1.1. OSI Definitions of OAM

The scope of OAM for any service and/or transport/network infrastructure technologies can be very broad in nature. OSI has defined the following five generic functional areas commonly abbreviated as "FCAPS" [NM-Standards]:

- o Fault Management,
- o Configuration Management,
- o Accounting Management,
- o Performance Management, and
- o Security Management.

This document focuses on the Fault, Performance and to a limited extent the Configuration Management aspects. Other functional aspects of FCAPS and their relevance (or not) to NVO3 are for further study.

Fault Management can typically be viewed in terms of the following categories:

- o Fault Detection;

- o Fault Verification;
- o Fault Isolation;
- o Fault Notification and Alarm Suppression;
- o Fault Recovery.

Fault detection deals with mechanism(s) that can detect both hard failures such as link and device failures, and soft failures, such as software failure, memory corruption, misconfiguration, etc. Fault detection relies upon a set of mechanisms that first allow the observation of an event, then the use of a protocol to dynamically notify a network/system operator (or management system) about the event occurrence, then the use of diagnostic tools to assess the nature and severity of the fault.

After verifying that a fault has occurred along the data path, it is important to be able to isolate the fault to the level of a given device or link. Therefore, a fault isolation mechanism is needed in Fault Management. A fault notification mechanism should be used in conjunction with a fault detection mechanism to notify the devices upstream and downstream to the fault detection point. The fault notification mechanism should also notify NMS systems.

The terms "upstream" and "backward" are used here to denote the direction(s) from which data traffic is flowing. The terms "downstream" and "forward" denote the direction(s) to which data traffic is forwarded.

For example, when there is a client/server relationship between two layered networks (e.g., the NVO3 layer is a client of the outer IP server layer, while the inner IP layer is a client of the NVO3 server layer 2), fault detection at the server layer may result in the following fault notifications:

- o Sending a forward fault notification from the server layer to the client layer network(s) using the fault notification format appropriate to the client layer.
- o Sending a backward fault notification to the server layer, if applicable, in the reverse direction.
- o Sending a backward fault notification to the client layer, if applicable, in the reverse direction.

Finally, fault recovery deals with recovering from the detected failure by switching to an alternate available data path (depending

on the nature of the fault) using alternate devices or links. In fact, the controller can provision another virtual network, thus automatically resolving the reported problem.

The controller may also directly monitor the status of virtual network components such as Network Virtualization Edge elements (NVEs) [NVO3-framework] in order to respond to their failures. In addition to forward and backward fault notifications, the controller may deliver notifications to a higher level orchestration component, e.g., one responsible for Virtual Machine (VM) provisioning and management.

Note, given that the IP network on which NVO3 resides is usually self healing, it is expected that recovery by the NVO3 layer would not normally be required, although there may be a requirement for that layer to log that the problem has been detected and resolved. The special cases of a static IP overlay network, or possibly of a centrally controlled IP overlay network, may, however, require NVO3 involvement in fault recovery.

Performance Management deals with mechanism(s) that allow determining and measuring the performance of the network/services under consideration. Performance Management can be used to verify the compliance to both the service-level and network-level metric objectives/specifications. Performance Management typically consists of measuring performance metrics, e.g., Frame Loss, Frame Delay, Frame Delay Variation (aka Jitter), Frame throughput, Frame discard, etc., across managed entities when the managed entities are in available state. Performance Management is suspended across unavailable managed entities.

## 1.2. 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].

## 1.3. Relationship with Other OAM Work

This document leverages requirements that originate with other OAM work, specifically the following:

- o [RFC6136] provides a template and some of the high level requirements and introductory wording.
- o [IEEE802.1ag] is expected to provide a subset of the requirements for NVO3 both at the Tenant level and also within the L3 Overlay network.

- o [Y.1731] is expected to provide a subset of the requirements for NVO3 at the Tenant level.
- o Section 3.8 of [NVO3-DP-Reqs] lists several requirements specifically concerning ECMP/LAG.

## 2. Terminology

The terminology defined in [NVO3-framework] and [NVO3-DP-Reqs] is used throughout this document. We introduce no new terminology.

## 3. NVO3 Reference Model

Figure 1 below reproduces the generic NVO3 reference model as per [NVO3-framework].

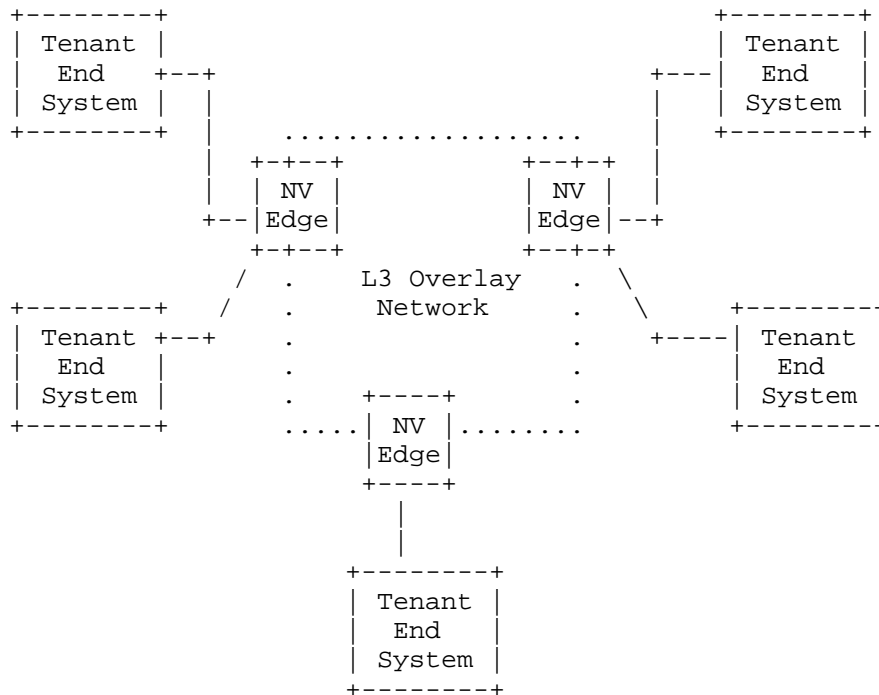


Figure 1: Generic reference model for DC network virtualization over a Layer3 infrastructure

Figure 2 below, reproduces the Generic reference model for the NV Edge (NVE) as per [NVO3-DP-Reqs].

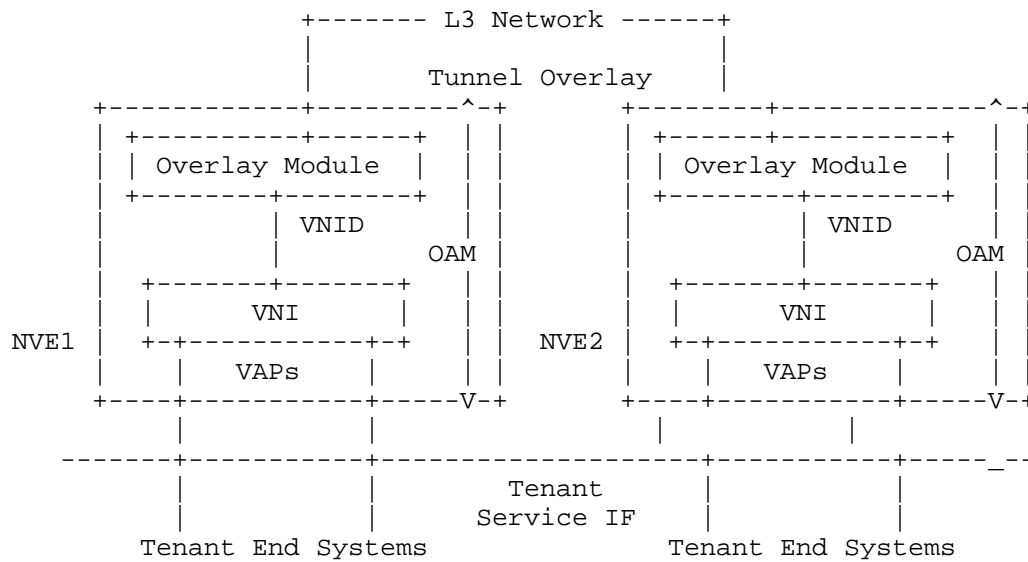


Figure 2: Generic reference model for NV Edge

#### 4. OAM Framework for NVO3

Figure 1 showed the generic reference model for a DC network virtualization over an L3 (or L3VPN) infrastructure while Figure 2 showed the generic reference model for the Network Virtualization (NV) Edge.

L3 network(s) or L3 VPN networks (either IPv6 or IPv4, or a combination thereof), provide transport for an emulated layer 2 created by NV Edge devices. Unicast and multicast tunneling methods (de-multiplexed by Virtual Network Identifier (VNID)) are used to provide connectivity between the NV Edge devices. The NV Edge devices then present an emulated layer 2 network to the Tenant End Systems at a Virtual Network Interface (VNI) through Virtual Access Points (VAPs). The NV Edge devices map layer 2 unicast to layer 3 unicast point-to-point tunnels and may either map layer 2 multicast to layer 3 multicast tunnels or may replicate packets onto multiple layer 3 unicast tunnels.

##### 4.1. OAM Layering

The emulated layer 2 network is provided by the NV Edge devices to which the Tenant End Systems are connected. This network of NV Edges can be operated by a single service provider or can span across multiple administrative domains. Likewise, the L3 Overlay Network can be operated by a single service provider or span across multiple administrative domains.

While each of the layers is responsible for its own OAM, each layer may consist of several different administrative domains. Figure 3 shows an example.

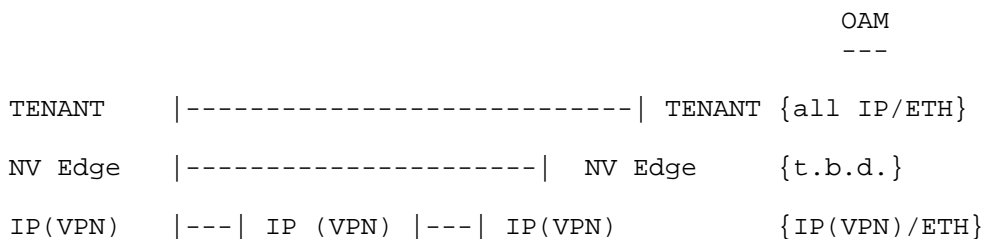


Figure 3: OAM layers in an NVO3 network

For example, at the bottom, at the L3 IP overlay network layer IP(VPN) and/or Ethernet OAM mechanisms are used to probe link by link, node to node etc. OAM addressing here means physical node loopback or interface addresses.

Further up, at the NV Edge layer, NVO3 OAM messages are used to probe the NV Edge to NV Edge tunnels and NV Edge entity status. OAM addressing here likely means the physical node loopback together with the VNI (to de-multiplex the tunnels).

Finally, at the Tenant layer, the IP and/or Ethernet OAM mechanisms are again used but here they are operating over the logical L2/L3 provided by the NV-Edge through the VAP. OAM addressing at this layer deals with the logical interfaces on Vswitches and Virtual Machines.

#### 4.2. OAM Domains

Complex OAM relationships exist as a result of the hierarchical layering of responsibility and of breaking up of end-to-end responsibility.

The OAM domain above NVO3, is expected to be supported by existing IP and L2 OAM methods and tools.



The OAM domain below NVO3, is expected to be supported by existing IP /L2 and MPLS OAM methods and tools. Where this layer is actually multiple domains spliced together, the existing methods to deal with these boundaries are unchanged. Note however that exposing LAG/ECMP detailed behavior may result in additional requirements to this domain, the details of which will be specified in the future versions of this draft.

When we refer to an OAM domain in this document, or just 'domain', we therefore refer to a closed set of NV Edges and the tunnels which interconnect them. Inter-domain OAM considerations will be specified in the future versions of this draft.

## 5. NVO3 OAM Requirements

The following numbered requirements originate from [RFC6136]. All are included however where they seem obviously not relevant (to the present authors) an explanation as to why is included.

### 5.1. Discovery

R1) NVO3 OAM MUST allow an NV Edge device to dynamically discover other NV Edge devices that share the same VNI within a given NVO3 domain. This may be based on a discovery mechanism used to set up data path forwarding between NVEs.

### 5.2. Connectivity Fault Management

#### 5.2.1. Connectivity Fault Detection

R2) NVO3 OAM MUST allow proactive connectivity monitoring between two or more NV Edge devices that support the same VNIs within a given NVO3 domain. NVO3 OAM MAY act as a protection trigger. That is, automatic recovery from transmission facility failure by switchover to a redundant replacement facility may be triggered by notifications from NVO3 OAM.

R3) NVO3 OAM MUST allow monitoring/tracing of all possible paths in the underlay network between a specified set of two or more NV Edge devices. Using this feature, equal cost paths that traverse LAG and/or ECMP may be differentiated.

#### 5.2.2. Connectivity Fault Verification

R4) NVO3 OAM MUST allow connectivity fault verification between two or more NV Edge devices that support the same VNI within a given NVO3 domain.

#### 5.2.3. Connectivity Fault localization

R5) NVO3 OAM MUST allow connectivity fault localization between two or more NV Edge devices that support the same VNI within a given NVO3 domain.

#### 5.2.4. Connectivity Fault Notification and Alarm Suppression

R6) NVO3 OAM MUST support fault notification to be triggered as a result of the faults occurring in the underneath network infrastructure. This fault notification SHOULD be used for the suppression of redundant service-level alarms.

#### 5.3. Frame Loss

R7) NVO3 OAM MUST support measurement of per VNI frame loss between two NV Edge devices that support the same VNI within a given NVO3 domain.

#### 5.4. Frame Delay

R8) NVO3 OAM MUST support measurement of per VNI two-way frame delay between two NV edge devices that support the same VNI within a given NVO3 domain.

R9) NVO3 OAM MUST support measurement of per VNI one-way frame delay between two NV Edge devices that support the same VNI within a given NVO3 domain.

#### 5.5. Frame Delay Variation

R10) NVO3 OAM MUST support measurement of per VNI frame delay variation between two NV Edge devices that support the same VNI within a given NVO3 domain.

#### 5.6. Frame Throughput

R11) NVO3 OAM MAY [\*\*\* Should this be stronger? \*\*\*] support measurement of per VNI frame throughput (in frames and bytes) between two NV Edge devices that support the same VNI within a given NVO3 domain. This feature could be an effective way to confirm whether or not assigned path bandwidth conforms to service level agreement before providing the path between two NV Edge devices.

#### 5.7. Frame Discard

R12) NVO3 OAM MAY support measurement of per VNI frame discard between two NV Edge devices that support the same VNI within a given

NVO3 domain. This feature MAY be effective to monitor bursty traffic between two NV Edge devices.

#### 5.8. Availability

A service may be considered unavailable if the service frames/packets do not reach their intended destination (e.g., connectivity is down) or the service is degraded (e.g., frame loss and/or frame delay and/or delay variation threshold is exceeded). Entry and exit conditions may be defined for the unavailable state. Availability itself may be defined in the context of a service type. Since availability measurement may be associated with connectivity, frame loss, frame delay, and frame delay variation measurements, no additional requirements are specified currently.

#### 5.9. Data Path Forwarding

R13) NVO3 OAM frames MUST be forwarded along the same path (i.e., links (including LAG members) and nodes) as the NVO3 data frames.

R14) NVO3 OAM frames MUST provide a mechanism to exercise/trace all data paths that result due to ECMP/LAG hops in the underlay network.

#### 5.10. Scalability

R15) NVO3 OAM MUST be scalable such that an NV edge device can support proactive OAM for each VNI that is supported by the device. (Note - Likely very hard to achieve with hash based ECMP/LAG).

#### 5.11. Extensibility

R16) NVO3 OAM should be extensible such that new functionality and information elements related to this functionality can be introduced in the future.

R17) NVO3 OAM MUST be defined such that devices not supporting the OAM are able to forward the OAM frames in a similar fashion as the regular NVO3 data frames/packets.

#### 5.12. Security

R18) NVO3 OAM frames MUST be prevented from leaking outside their NVO3 domain.

R19) NVO3 OAM frames from outside an NVO3 domain MUST be prevented from entering the said NVO3 domain when such OAM frames belong to the same level or to a lower-level OAM. (Trivially met because hierarchical domains are independent technologies.)

R20) NVO3 OAM frames from outside an NVO3 domain MUST be transported transparently inside the NVO3 domain when such OAM frames belong to a higher-level NVO3 domain. (Trivially met because hierarchical domains are independent technologies).

#### 5.13. Transport Independence

Similar to transport requirement from [RFC6136], we expect NVO3 OAM will leverage the OAM capabilities of the transport layer (e.g., IP underlay).

R21) NVO3 OAM MAY allow adaptation/interworking with its IP underlay OAM functions. For example, this would be useful to allow fault notifications from the IP layer to be sent to the NVO3 layer and likewise exposure of LAG / ECMP will require such non-independence.

#### 5.14. Application Independence

R22) NVO3 OAM MUST [\*\*\* discuss -- is this too strong? \*\*\*] be independent of the application technologies and specific application OAM capabilities.

[Comment -- ECM: Noticed Nicira implementation has a dedicated NVP manager node to play the role of FCAPS here. It is both application layer and OAM layer. May not meet this requirement. In reality, due to the nature of overlay network, very often, vendors are going to make everything all together to a dedicated manager node.]

#### 5.15. Prioritization

R23) NVO3 OAM messages MUST be preferentially treated in NVE and between NVEs, since NVO3 OAM MAY be used to trigger protection switching. As noted above (R2), protection switching is the automatic replacement of a failed transmission facility with a working one providing equal or greater capacity, typically within a few tens of milliseconds from fault detection.

[Comment -- ECM: giving NVO3 OAM messages priority treatment may interfere with measurements of frame delay and jitter.]

### 6. Items for Further Discussion

This section identifies a set of operational items which may be elaborated further if these items fall within the scope of the NVO3.

- o VNID renumbering support

- \* Means to change the VNID assigned to a given instance MUST [\*\*\* discuss: is this too strong? \*\*\*] be supported.
- \* System convergence subsequent to VNID renumbering MUST NOT take longer than a few seconds, to minimize impact on the tenant systems.
- \* A VNE MUST be able to map a VNID with a virtual network context.
- o VNI migration and management operations
  - \* Means to delete an existing VNI MUST be supported.
  - \* Means to add a new VNI MUST be supported.
  - \* Means to merge several VNIs MAY be supported.
  - \* Means to retrieve reporting data per VNI MUST be supported.
  - \* Means to monitor the network resources per VNI MUST be supported.
- o Support of planned maintenance operations on the NVO3 infrastructure
  - \* Graceful procedure to allow for planned maintenance operation on NVE MUST be supported. This includes undoing any configuration changes made for maintenance purposes after completion of the maintenance.
- o Support for communication among virtual networks
  - \* For global reachability purposes, communication among virtual networks MUST be supported. This can be enforced using a NAT function.
- o Activation of new network-related services to the NVO3
  - \* Means to assist in activating new network services (e.g., multicast) without impacting running service should be supported.
- o Inter-operator NVO3 considerations
  - \* As NVO3 may be deployed over inter-operator infrastructure, coordinating OAM actions in each individual domain are required to ensure an end-to-end OAM. In particular, this assumes

existence of agreements on the measurement and monitoring methods, fault detection and repair actions, extending QoS classes (e.g., DSCP mapping policies), etc.

[[DISCUSSION NOTE: Should inter-operator issues be declared out of scope?]]

## 7. IANA Considerations

This memo includes no request to IANA.

## 8. Security Considerations

TBD

## 9. Acknowledgements

The authors are grateful for the contributions of David Black, Dennis Qin, Erik Smith and Ziyi Yang to this latest version.

## 10. References

### 10.1. Normative References

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

### 10.2. Informative References

[IEEE802.1ag]  
IEEE, "IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks, Amendment 5: Connectivity Fault Management", 2007.

[IEEE802.1ah]  
IEEE, "IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks, Amendment 6: Provider Backbone Bridges", 2008.

[NM-Standards]  
ITU-T, "ITU-T Recommendation M.3400 (02/2000) - TMN Management Functions", February 2000.

[NVO3-DP-Reqs]  
Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", October 2012.

## [NVO3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", July 2012.

[RFC6136] Sajassi, A. and D. Mohan, "Layer 2 Virtual Private Network (L2VPN) Operations, Administration, and Maintenance (OAM) Requirements and Framework", RFC 6136, March 2011.

[Y.1731] ITU-T, "ITU-T Recommendation Y.1731 (02/08) - OAM functions and mechanisms for Ethernet based networks", February 2008.

## Authors' Addresses

Peter Ashwood-Smith  
Huawei Technologies  
303 Terry Fox Drive, Suite 400  
Kanata, Ontario K2K 3J1  
Canada

Phone: +1 613 595-1900  
Email: Peter.AshwoodSmith@huawei.com

Ranga Iyengar  
Huawei Technologies USA  
2330 Central Expy  
Santa Clara, CA 95050  
USA

Email: ranga.Iyengar@huawei.com

Tina Tsou  
Huawei Technologies USA  
2330 Central Expy  
Santa Clara, CA 95050  
USA

Email: Tina.Tsou.Zouting@huawei.com

Ali Sajassi  
Cisco Technologies  
170 West Tasman Drive  
San Jose, CA 95134  
USA

Email: [sajassi@cisco.com](mailto:sajassi@cisco.com)

Mohamed Boucadair  
France Telecom  
Rennes 35000  
France

Email: [mohamed.boucadair@orange.com](mailto:mohamed.boucadair@orange.com)

Christian Jacquenet  
France Telecom  
Rennes 35000  
France

Email: [christian.jacquenet@orange.com](mailto:christian.jacquenet@orange.com)

Masahiro Daikoku  
KDDI corporation  
3-10-10, Iidabashi, Chiyoda-ku  
Tokyo 1028460  
Japan

Email: [ms-daikoku@kddi.com](mailto:ms-daikoku@kddi.com)



Internet Engineering Task Force  
Internet Draft  
Intended status: Informational  
Expires: May 2013

Nabil Bitar  
Verizon

Marc Lasserre  
Florin Balus  
Alcatel-Lucent

Thomas Morin  
France Telecom Orange

Lizhong Jin  
Bhumip Khasnabish  
ZTE

November 28, 2012

NVO3 Data Plane Requirements  
draft-bl-nvo3-dataplane-requirements-03.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), 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 May 28, 2013.

## Copyright Notice

Copyright (c) 2012 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

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,a.k.a.: 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.

DC: Data Center

BUM: Broadcast, Unknown Unicast, Multicast traffic

TS: Tenant System

VAP: Virtual Access Point

VNI: Virtual Network Instance

VNID: VNI ID

## 2. Data Path Overview

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

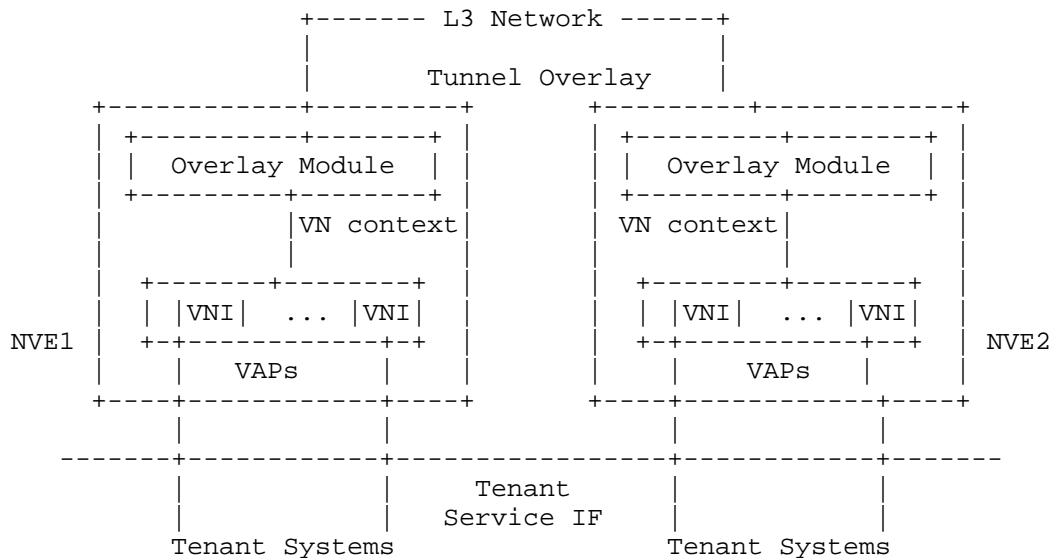


Figure 1 : Generic reference model for NV Edge

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 MAC addresses and L3 tunnel destination addresses. 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 source MAC address is learned against the VAP or the NVO3 tunnel 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 be able to build at least a default flooding tree per VNI. 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:

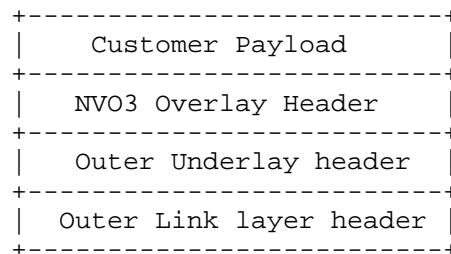


Figure 2 : NVO3 encapsulated frame

where

- . Customer payload: 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.

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 NOT require provisioning touches and SHOULD NOT use intermediate hand-offs, for example VLANs.

#### 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,a.k.a.: 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 NV03 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  
ZTE  
Email : lizhong.jin@zte.com.cn

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

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 22, 2013

Y. Gu  
Y. Li  
Huawei  
Oct 19, 2012

The mechanism and signalling between TES and NVE  
draft-gu-nvo3-tes-nve-mechanism-01

Abstract

This draft introduces the interaction required between TES to NVE when NVE is located in an external box to TES. The signaling between TES and NVE has to be designed carefully to reflect all the interaction requirements. This document describes the relevant considerations for such design and also provides a basic analysis of the potential reusable protocols. Currently this draft focuses on the general interaction procedures with relevant parameters and the signaling design consideration. It may be extended to show more detailed signalling design recommendation and/or solution recommendation in the future with the progress of NVO3's work.

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, 2013.

Copyright Notice

Copyright (c) 2012 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. Terminologies and concepts . . . . .	6
3. TES to NVE Interaction . . . . .	9
3.1. Interaction Intentions . . . . .	9
3.2. VM Lifetime Events . . . . .	9
3.2.1. VM Creation . . . . .	9
3.2.2. VM Pre-associate with NVE . . . . .	10
3.2.3. VM Associate with NVE . . . . .	10
3.2.4. VM Suspension . . . . .	10
3.2.5. VM Resume . . . . .	11
3.2.6. VM Migration . . . . .	11
3.2.7. VM Termination . . . . .	11
3.2.8. VM Full Lifecycle Sketch . . . . .	11
3.3. Events, Interaction and Parameters . . . . .	13
3.3.1. VM Pre-association . . . . .	13
3.3.2. VM Association . . . . .	14
3.3.3. VM Suspension . . . . .	15
3.3.4. VM Resume . . . . .	15
3.3.5. VM Emigration . . . . .	16
3.3.6. VM Immigration . . . . .	16
3.3.7. VM Termination . . . . .	17
3.3.8. Keep-alive . . . . .	17
3.3.9. NVE Local Changes . . . . .	18
3.4. Signalling Design Considerations . . . . .	18
3.4.1. General Requirements . . . . .	18
3.4.2. Consideration . . . . .	19
3.4.3. Signalling States Machine . . . . .	19
4. Security Considerations . . . . .	20
5. Appendix 1: Mechanism Analysis . . . . .	20
5.1. IEEE 802.1Qbg . . . . .	20
5.1.1. Brief Introduction . . . . .	21
5.2. BGP . . . . .	23
5.3. External Controller . . . . .	23
6. References . . . . .	23
6.1. Normative Reference . . . . .	23
6.2. Informative Reference . . . . .	23
Authors' Addresses . . . . .	24

## 1. Introduction

Tenant End System (TES) is the physical host where tenant deploys their applications. Tenants' applications can be deployed on a physical server directly or on a virtual machine resided on a physical server. Tenant's virtual network, or say virtual data center, is an overlay network which is built on the underlying network, but logically independent of the underlying network. Network Virtualization Edge (NVE) is implemented with virtualization functions to encapsulate or decapsulate a tenant's packet that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses). A Tenant End System attaches to a Network Virtualization Edge (NVE) node, either directly or via a switched network (typically Ethernet). TES and NVE can be on the same physical server or on the separate devices. Fig1 to Fig3 show different NVE location cases. While TES and NVE are on the same physical server, the interaction between TES and NVE is via some proprietary internal interface which does not require a standard signaling protocol. Therefore such scenario is not the target of this document. For all the other scenarios, as long as the signaling between TES and NVE is visible to network developer, it is in the scope of this draft. We tried to examine the different locations of NVE to make sure the signaling interaction between NVE and TES cover as possible scenarios as possible.

- o (NVE Location 1) NVE and TES are co-located in a physical server. VM connects to NVE on Hypervisor. In this case, there should be some mechanism to assist Hypervisor know of VM changes, including adding, deleting and migration. Both VM and Hypervisor, as well as network service appliance, are controlled by VM Manager. VM Manager is aware of any VM identity and event, hence it can easily notify NVE about the information through some internal interface. A publically available standard protocol is not necessary in this case. Refer to Fig1.

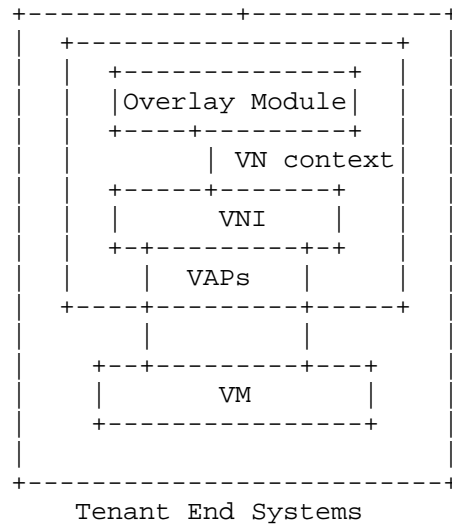


Figure 1

- o (NVE Location 2) TES connects to NVE on an external network entity next to it. VM is controlled by VM Manager, while NVE is controlled by some other management entity like network management system. Hence proprietary protocol between TES and NVE may not fit all the scenarios. A standard protocol to signal between TES and NVE is mandatory in this case. Refer to Fig2.

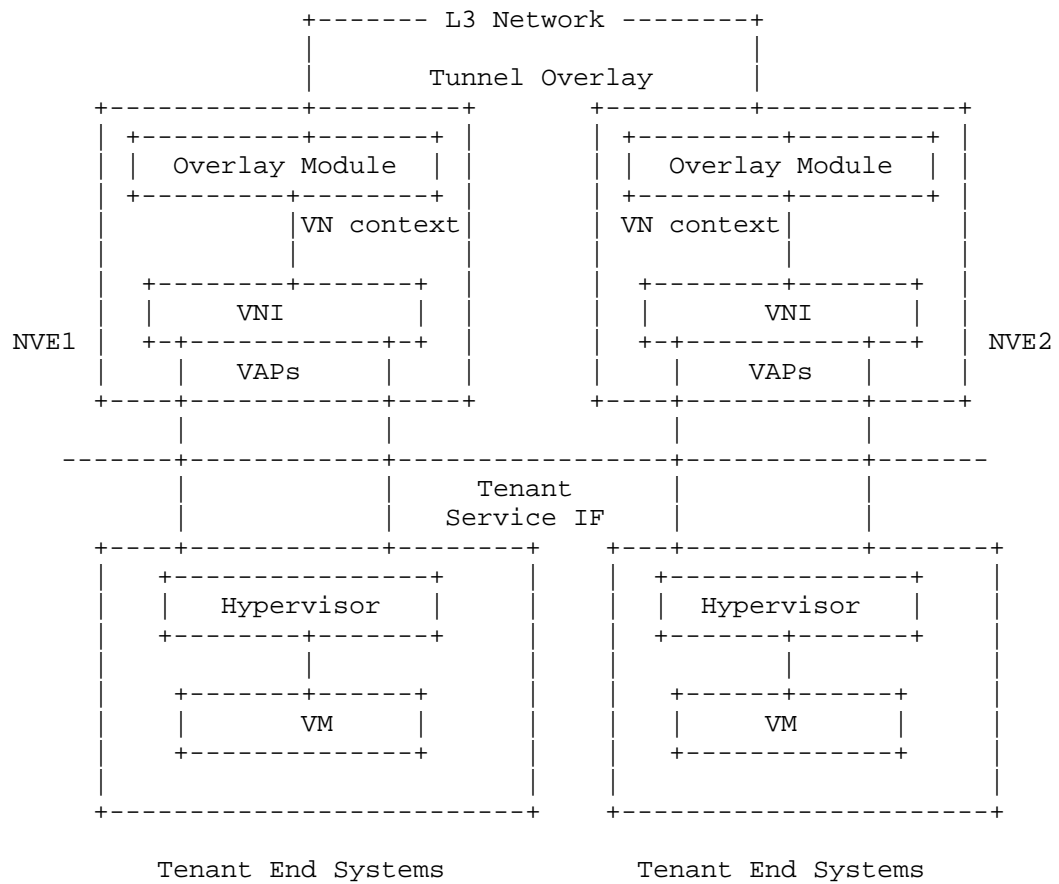


Figure 2: NVE Location3: VM connects to NVE on external network entity

- o (NVE Location 3) TES and NVE are indirectly connected. Refer to Fig3.



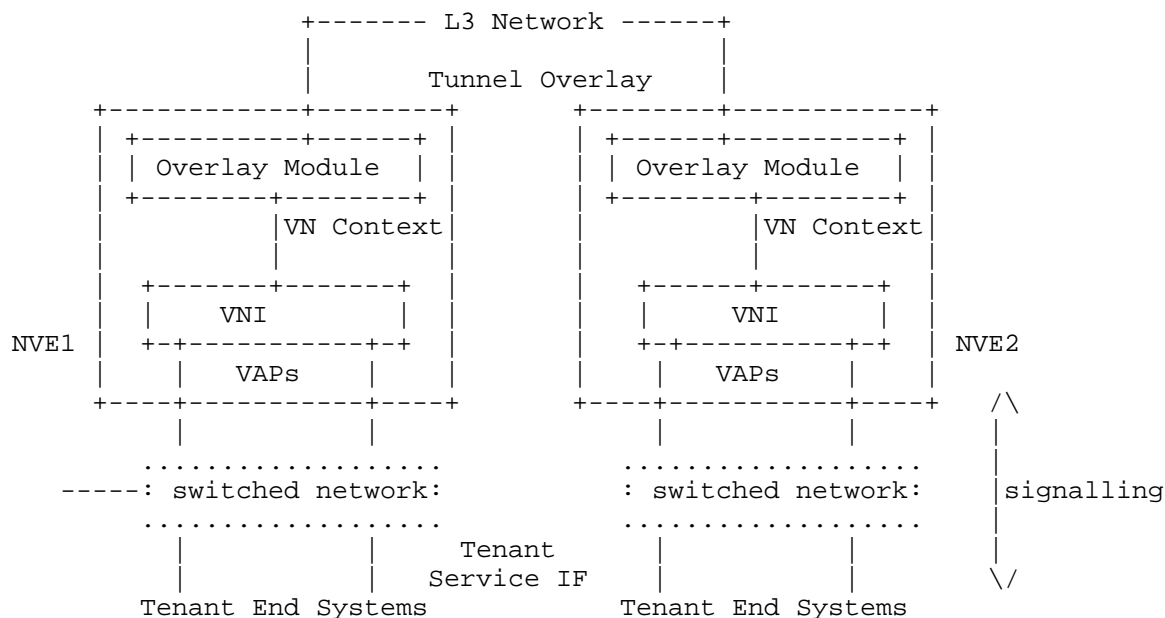


Figure 3: Reference model when TES and NVE are indirectly connected

In the mail list discussion, more than one mechanisms to be used between TES and NVE were discussed, including VDP (VSI Discovery and Configuration Protocol), BGP and others.. This draft is not going to make assertion about which protocol is better. We believe that each candidate protocol can, with some revision or updating, be used to exchange necessary events and information between TES and NVE. The final decision on which one to be used does not only depend on functionalities, but also some other aspects, e.g. lightweight to be implemented on server, widely deployment in the industry, efficiency and performance etc.

This draft first presents the recommended procedures of the TES and NVE signalling, key parameters of each step, and issues need to be addressed. Then a set of signaling design considerations are provided, which can be used as design requirements for the future signalling definition. In the appendix, we give a brief analysis on two existing protocols and also show how they can be revised to adapt to TES and NVE signaling.

## 2. Terminologies and concepts

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 document uses terms defined in [framework].

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

**VNI:** Virtual Network Instance. This is one instance of a virtual overlay network. Two Virtual Networks are isolated from one another and may use overlapping addresses.

**Virtual Network Context or VN Context:** Field that is part of the overlay encapsulation header 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).

**VNID:** Virtual Network Identifier. In the case where the VN context has global significance, this is the ID value that is carried in each data packet in the overlay encapsulation that identifies the Virtual Network the packet belongs to.

**NVE:** Network Virtualization Edge. It is a network entity that sits on the edge of the NVO3 network. It implements network virtualization functions that allow for L2 and/or L3 tenant separation and for hiding tenant addressing information (MAC and IP addresses). An NVE could be implemented as part of a virtual switch within a hypervisor, a physical switch or router, a Network Service Appliance or even be embedded within an End Station.

**Underlay or Underlying Network:** This is the network that provides the connectivity between NVEs. The Underlying Network can be completely unaware of the overlay packets. Addresses within the Underlying Network are also referred to as "outer addresses" because they exist in the outer encapsulation. The Underlying Network can use a completely different protocol (and address family) from that of the overlay.

**Data Center (DC):** A physical complex housing physical servers, network switches and routers, Network Service Appliances and networked storage. The purpose of a Data Center is to provide application and/or compute and/or storage services. One such service is virtualized data center services, also known as Infrastructure as

a Service.

VM: Virtual Machine. Several Virtual Machines can share the resources of a single physical computer server using the services of a Hypervisor (see below definition).

Hypervisor: Server virtualization software running on a physical compute server that hosts Virtual Machines. The hypervisor provides shared compute/memory/storage and network connectivity to the VMs that it hosts. Hypervisors often embed a Virtual Switch (see below).

Virtual Switch: A function within a Hypervisor (typically implemented in software) that provides similar services to a physical Ethernet switch. It switches Ethernet frames between VMs' virtual NICs within the same physical server, or between a VM and a physical NIC card connecting the server to a physical Ethernet switch. It also enforces network isolation between VMs that should not communicate with each other.

Tenant: A customer who consumes virtualized data center services offered by a cloud service provider. A single tenant may consume one or more Virtual Data Centers hosted by the same cloud service provider.

Tenant End System: It defines an end system of a particular tenant, which can be for instance a virtual machine (VM), a non-virtualized server, or a physical appliance.

Virtual Access Points (VAPs): Tenant End Systems are connected to the Tenant Instance through Virtual Access Points (VAPs). The VAPs can be in reality physical ports on a ToR or virtual ports identified through logical interface identifiers (VLANs, internal VSwitch Interface ID leading to a VM).

VN Name: A globally unique name for a VN. The VN Name is not carried in data packets originating from End Stations, but must be mapped into an appropriate VN-ID for a particular encapsulating technology. Using VN Names rather than VN-IDs to identify VNs in configuration files and control protocols increases the portability of a VDC and its associated VNs when moving among different administrative domains (e.g. switching to a different cloud service provider).

VSI: Virtual Station Interface. Typically, a VSI is a virtual NIC connected directly with a VM. [Qbg]

### 3. TES to NVE Interaction

#### 3.1. Interaction Intentions

While TES is a non-virtualized physical server, a single physical interface on NVE is exclusively attached to a single tenant and the attachment doesn't change very frequently. In this case, NVE can be pre-configured with tenant's network properties and policies to execute appropriate packet processing. And when a physical server moves, which means a server change its attach point to the network, the new NVE, to which the server is going to attach with in the new location, can also be preconfigured. In this case, there is no need to proceed signalling between TES and NVE.

While TES is a virtualized server with multiple VMs, the interaction between TES and NVE becomes necessary. A physical interface on NVE can be attached to multiple VMs, which could belong to the same or different tenants, and VMs can be moved to new locations without physical shutdown, which means NVE not able to know VMs' attachment and/or detachment by checking the physical port. As described in [framework], NVE need to establish Virtual Network Instance for each tenant virtual network attached to it through physical interface, NVE must be able to know which tenants are attached to it and the corresponding VMs belongs to each tenants. So that NVE must be able to 1) identify and distinguish VMs attached to NVE through the same physical interface; 2) identify which tenant the VM belongs to; 3) get the network policies that is associated with the tenant. That's why a interaction signalling between TES and NVE is needed. Of course the signalling between TES and NVE are not limited to the above intentions. While looking into the detail processing of VM events, we will find more signalling functionalities and processing on TES and NVE.

#### 3.2. VM Lifetime Events

Not every VM has to pass through all the listed VM lifetime events. Any VM can have at least two or a combination of the following events.

##### 3.2.1. VM Creation

VM Manager indicates the hypervisor to schedule resources on server for a particular VM, including CPU, Memory, Storage and Network resources. After the VM is created on the server, the VM has necessary resource and is ready to be launched. The creation of VM doesn't necessarily mean the VM is running. The VM can be created but not launched for some while as long as the manager would like. The VM can be created and launched at once. Launching a VM just like

startup a physical computer.

Though VM creation is a very important events for VM, but the attached NVE needn't be aware of this event.

### 3.2.2. VM Pre-associate with NVE

VM Manager can decide when to launch a VM and connect the VM to the network. Before VM connects to network, operator need to provision VM's network properties and policies to the NVE that the VM is attached to. The examples of network properties are VM MAC address, tenant virtual network identifier. The examples of policies are ACL and QoS. But these properties and policies are not immediately activated on NVE unless the VM Manager indicate the VM to connect to network. This is called Pre-association. Pre-association is optional event.

### 3.2.3. VM Associate with NVE

This event means the VM is going to connect to the network. NVE has to get VM's network properties and policies, assign resources and install these properties and policies. If there is Pre-association before Association, NVE can reduce the time for Association. While VM is associated, it can use network resources as a physical server does.

Association can happen with or without pre-association. If there is Pre-association before Association, NVE has already the network properties and policies restored, or even installed. If the network properties and policies in Association message is the same as the pre-association, NVE can activate the installed network properties and policies. If they are different, the old reserved resources should be released and the new network properties and policies are installed and activated.

### 3.2.4. VM Suspension

Creating and terminating VM may take a considerable amount of time. Instead of performing these operations, operators can suspend a virtual machine for the required time and quickly resume it later. Suspending a VM is similar to putting a real computer into the sleep mode. When suspending a VM, VM's current state (including the state of all applications and processes running in the VM) is stored. When the suspended virtual machine is resumed, it continues operating at the same point the virtual machine was at the time of its suspending.

### 3.2.5. VM Resume

To activate the suspended VM. The suspended applications will start again at the state the VM was suspended. It's not always predictable on when a suspended VM will be resumed.

### 3.2.6. VM Migration

Two kinds VM migration, i.e. hot migration (or live migration) and offline migration. The processing of offline migration is similar to terminating the VM on one server and creating it on another server. The running applications on the VM will be broken and then be restarted again on the new location. For live migration, VM is lively migrated from one location to another, and the running applications should not be visibly disrupted. There is no termination or creation during live migration, so it's highly important to let NVE be aware of the migration so that corresponding network properties and policies can be correctly obtained, installed and activated on new location, and removed from the old location. Otherwise, there might be security risk and will influence or even interrupted running applications.

There are two sub-type for VM migration: VM emigration and VM immigration.

- o VM Emigrating: VM is emigrating from this server. Hence, all the relevant resources on the server and attached NVE are disabled, but not removed right now, and is ready to be removed once VM is successfully migrated. If VM is failed to immigrate on the new location, VM has to be resumed on old location with the states and policies disabled by old NVE.
- o VM Immigrating: VM is immigrating to this server. The server and attached NVE has prepared the necessary resources and is ready to enable the VM's properties and policies once VM is successfully migrated.

### 3.2.7. VM Termination

All applications and processing on VM is terminated. All VM's resources on server, including CPU, Memory, Storage and network resources, are released. There is no such a VM any more.

### 3.2.8. VM Full Lifecycle Sketch

Not every VM has to pass through all the lifetime events emulated in above. A simplest VM life has only VM Creation, VM Associating with NVE and VM Termination. A most complex VM life has all the events

listed in above. In this section, we show a sketch for a VM's full lifecycle with all listed events. This is helpful for the signalling designation in the future.

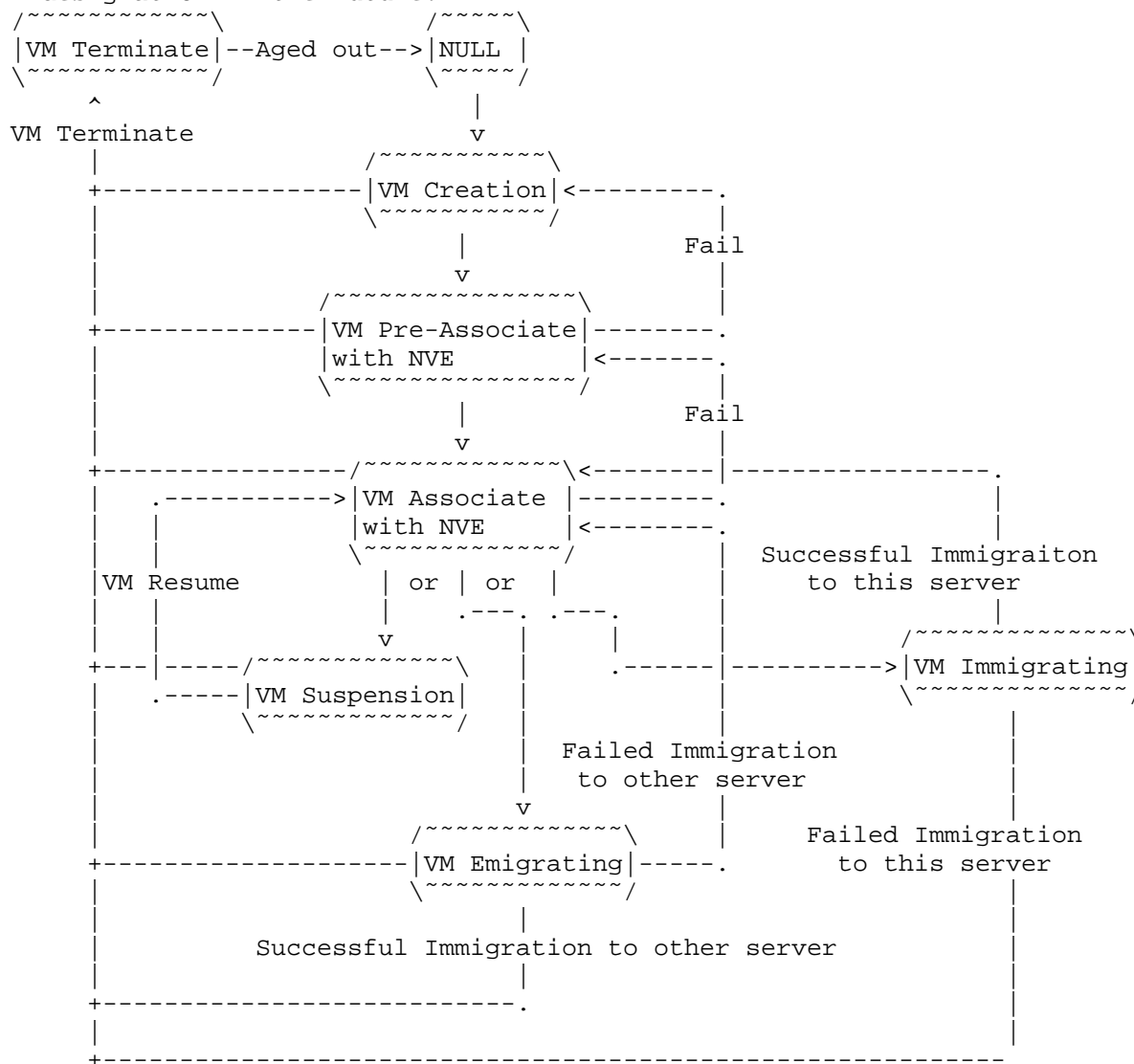


Figure 4: VM Full Lifecycle Sketch

### 3.3. Events, Interaction and Parameters

In this section, we will present description of interaction, parameters and special concerns for each VM events are provided. The interaction has strong relationship with VM lifetime events, but is not one-to-one mapping, for example, there is no interaction for VM Creation. For VM events, the interaction is initiated by hypervisor on behalf of a VM and sent to VNI on attached NVE. But this is not always the case, since NVE may also initiate interaction if there is some changes happen on NVE and those changes must be learned by particular VMs.

#### 3.3.1. VM Pre-association

- o Interaction: This event will trigger Hypervisor to compose a pre-association message, and then Hypervisor sends the message to NVE. While receives the pre-association message, NVE needs to authorize the VM and/or Hypervisor, obtain VM's network properties and policies, and install the properties and policies on NVE.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Pre-association.
  - \* VMID, a global unique ID in Data Center for a VM. A VM can have more than one MAC addresses and belongs to more than one VNID, so a VMID is necessary for NVE to accosicate the VNIDs and MACs with the particular VM.
  - \* VNID(s), a global unique ID in Data Center for a tenant's virtual network.
  - \* MAC addresses, a VM may have more than one MAC addresses. A VM may also belongs to more than one virtual network. So the MAC address(s) and VNID should be presented in a way that NVE can identify which MAC addresses belongs to which VNID.
  - \* Policies, including ACL, QoS, Priority and etc. In the case there are more than one VNID associated with the VM, Policies should be explicitly indicated to belong to which VNID.
- o Response: After NVE processes pre-association message, it repond to TES with processing result. The response can be SUCCESS or FAIL with such indicated reasons as FAILED AUTHORIZTION, CONFLICT POLICIES(e.g. the provisioned policies are conflict with other existed policies on NVE), NON-SUFFICIENT RESOURCES(e.g. the NVE has not enough resources to install the provisioned policies).



### 3.3.2. VM Association

- o Interaction: This event will trigger Hypervisor to compose an Association message, and then Hypervisor sends the message to NVE. Association can happen with or without a Pre-association message.
  - \* If there is a Pre-association message before Association, NVE needs to compare the information provided by Pre-association and Association. If they are same, NVE can activate the pre-installed resources. If they are different, NVE needs to do some additional work depending on what information has been changed from pre-association to association. For example, if policy or VNID is changed, NVE needs to update its memory.
  - \* If there is no Pre-association message before Association, NVE needs to do authorization, obtain VM's network properties and policies, and install and activate the properties and policies on NVE.
  - \* If there is another successful Association message before this Association, NVE needs to compare the information provided by previous provisioned Association and this Association. If all is the same, NVE do nothing except for update the VM's timer. If there is different in comparison, NVE needs to do some additional work, depends on what information is changed. For example, if policies or VNID is changed, NVE needs to update its memory.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Association.
  - \* VMID
  - \* VNID(s)
  - \* MAC addresses
  - \* Policies
- o Response: After NVE processes Association message, it repond to TES with processing result. The response can be SUCCESS or FAIL with such indicated reasons as FAILED AUTHORIZATION, CONFLICT POLICIES(e.g. the provisioned policies are conflict with other existed policies on NVE), NON-SUFFICIENT RESOURCES(e.g. the NVE has not enough resources to install the provisioned policies).

### 3.3.3. VM Suspension

- o Interaction: This event will trigger Hypervisor to compose an Suspension message or an Association message with Suspension indication, and then Hypervisor sends the message to NVE. Suspension must happen after Successful Association. On receiving a Suspension message, NVE inactivate, but not remove, the VM's resources and prepare for the next Resume message. In the state of suspension, NVE acts similar as it in Pre-association state. The FDB can be aged out during VM suspension.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Suspension or an Association message with Suspension indication
  - \* VMID
- o Response: After NVE processes Suspension message, it repond to TES with processing result. The response can be SUCCESS or FAIL . If it's FAIL, it may be because the NVE is too busy to process the message.

### 3.3.4. VM Resume

- o Interaction: This event will trigger Hypervisor to compose an Resume message or an Association message with Resume indication, and then Hypervisor sends the message to NVE. Resume is supposed to happen after a successful Suspension message, otherwise, it will be responded with a SUCCESS message and NVE will do nothing to the message.. On receiving a Resume message, NVE activates the VM's resources and prepare.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Resume or an Association message with Resume indication
  - \* VMID
- o Response: After NVE processes Resume message, it repond to TES with processing result. The response can be SUCCESS or FAIL. If it's FAIL, it may be because the NVE is too busy to process the message.

### 3.3.5. VM Emigration

- o Interaction: This event will trigger Hypervisor to compose an Emigration message or an Association message with Emigration indication, and then Hypervisor sends the message to NVE. Emigration can happen after Pre-association, Association, Suspension or Resume.
- o On receiving VM Emigration message or indication, NVE inactivate VM's resources. But NVE doesn't immediately remove VM's resources and states, because an emigration maybe fail if the immigration on the remote server or NVE is failed. In that case, the emigrating VM may need to continue its work on the current server. NVE will wait for a next Termination message to remove the VM's resources or states on NVE.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Association.
  - \* VMID
- o Response: After NVE processes VM Emigration, it repond to TES with processing result. The response can be SUCCESS or FAIL. If it's FAIL, it may be because the NVE is too busy to process the message.

### 3.3.6. VM Immigration

- o Interaction: This event will trigger Hypervisor to compose an Immigration message, or an Pre-association/Association message with Immigration indication, call them immigration(Pre-asso) and Immigration(Asso). NVE's reaction to VM Immigration is silimar to its reaction to Pre-association or Association. If the result of Immigration processing is FAIL, the VM will not migrate to the new location and continue its work on old server. VM Manger may have to find another new location for the VM to migrate to.
- o To distinguish Immigration from Pre-association and Association is meaningful, [statemigration-framework]shows the problem of VM's flow-coupled state migration in case of VM live migration. The Immigration message can be a indication or trigger for the flow-coupled state migration on middleboxes.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.

- \* Operation, i.e. Immigration or an (Pre-)Association message with Immigration indication.
  - \* VMID
  - \* VNID(s)
  - \* MAC addresses
  - \* Policies
- o Response: After NVE processes Immigration message, it repond to TES with processing result. The response can be SUCCESS or FAIL with such indicated reasons as FAILED AUTHORIZTION, CONFLICT POLICIES(e.g. the provisioned policies are conflict with other existed policies on NVE), NON-SUFFICIENT RESOURCES(e.g. the NVE has not enough resources to install the provisioned policies).

### 3.3.7. VM Termination

- o Interaction: This event will trigger Hypervisor to compose an Termination message. NVE' will release VM's resources on NVE and remove all state about this VM.
- o Parameters: The signalling from TES to NVE should at least include the following mandatory parameters.
  - \* Operation, i.e. Termination
  - \* VMID
- o Response: After NVE processes Termination message, it repond to TES with processing result. The response can be SUCCESS or FAIL. If it's FAIL, it maybe because NVE is too busy to process the Termination message, however the VM can be terminated on the server anyway.

### 3.3.8. Keep-alive

This is not a VM lifetime events. Since the resources on NVE is precious, if a associated, pre-associated or suspended VM keeps idle for a pre-defined time, NVE will remove the VM's resources, so that NVE can serve other active VMs. In order to keep VM's resource on NVE, Hypervisor has to create keep-alive message, or an Pre-association/Association message with Keep-alive indication, NVE will update VM's timer upon the Keep-alive message.

Parameters: The signalling from TES to NVE should at least include

the following mandatory parameters.

- o Operation, i.e. Keep-alive or an (Pre-)Association message with Keep-alive indication.
- o VMID

### 3.3.9. NVE Local Changes

While VM associate with a VNID on NVE, NVE will generate local significant indicators for the VM and VNIDs, e.g. VID. If the indicators are sent to Hypervisor in previous response, and the indicators change later on, NVE need to create an Associate or a dedicated message with the changed indicators and send to Hypervisor, and Hypervisor will respond with processing result.

Note: Although we use the VM Lifetime events names as the names of messages in this section, it does mean that there should be a dedicated message for each event in the future signalling. Some of the events can be carried in one signalled message with different operation type. For example, an Association message with Immigration indication or an Association message with Suspension indication.

## 3.4. Signalling Design Considerations

### 3.4.1. General Requirements

#### 3.4.1.1. Basic Requirements

REQUIREMENT-1: The TNS (TES to NVE Signalling) MUST support TES to notify NVE about the VM's events, including but not limited to Pre-Association, Association, Emigration, Immigration and Termination.

REQUIREMENT-2: The TNS MUST support TES to notify NVE about the VM's VNID, which can be one identifier or a combination of several identifier.

REQUIREMENT-3: The TNS MUST support TES to notify NVE about the VM's address. The address MUST include one or both of MAC address of VM's virtual NIC and VM's IP address. And it SHOULD be extensible to carry new address type.

REQUIREMENT-4: The TNS MUST support NVE to notify TES about the VM's local tag. The local Tag type supported by TNP MUST include IEEE 802.1Q tag. And it SHOULD be extensible to carry other type of local tag.

#### 3.4.1.2. Extension Requirements

REQUIREMENT-5: The TNS SHOULD support NVE to notify TES about the VM's traffic PCP value.

In typical DC, where physical server connects to adjacent bridge, the data frame from server can be tagged with PCP or untagged. If a data frame is untagged, it can be tagged with PCP on adjacent bridge. While in virtualized DC, the adjacent bridge is Hypervisor. There are two options to deal with PCP tag, 1) data frame is tagged with PCP by VM, 2) data frame is tagged with PCP by Hypervisor and 3) data frame is tagged with PCP by NVE.

In cloud service, the VM can be anybody and it may want a higher priority than it should have. The VM can tag its data frame with higher PCP value and get better service. Based on the assumption that PCP provided by VM is not reliable, it's more reasonable to let the network to define the PCP value based on VM's priority, and enable bridges to tag the PCP value, as 2) or 3).

This problem is similar to local VID, which can be tagged either by Hypervisor or by NVE. The benefit to tag PCP by Hypervisor is to reduce the load on NVE.

#### 3.4.2. Consideration

To be added.

#### 3.4.3. Signalling States Machine

The interaction should be stateful. Both Hypervisor and NVE need to record the state of their signalling state. The main states are Pre-association, Association, Suspension, and Termination. The following diagram shows a the state machine of TES to NVE signalling. Only reasonable situations are listed in the diagram. In the future, more situation will be added to the state machine.

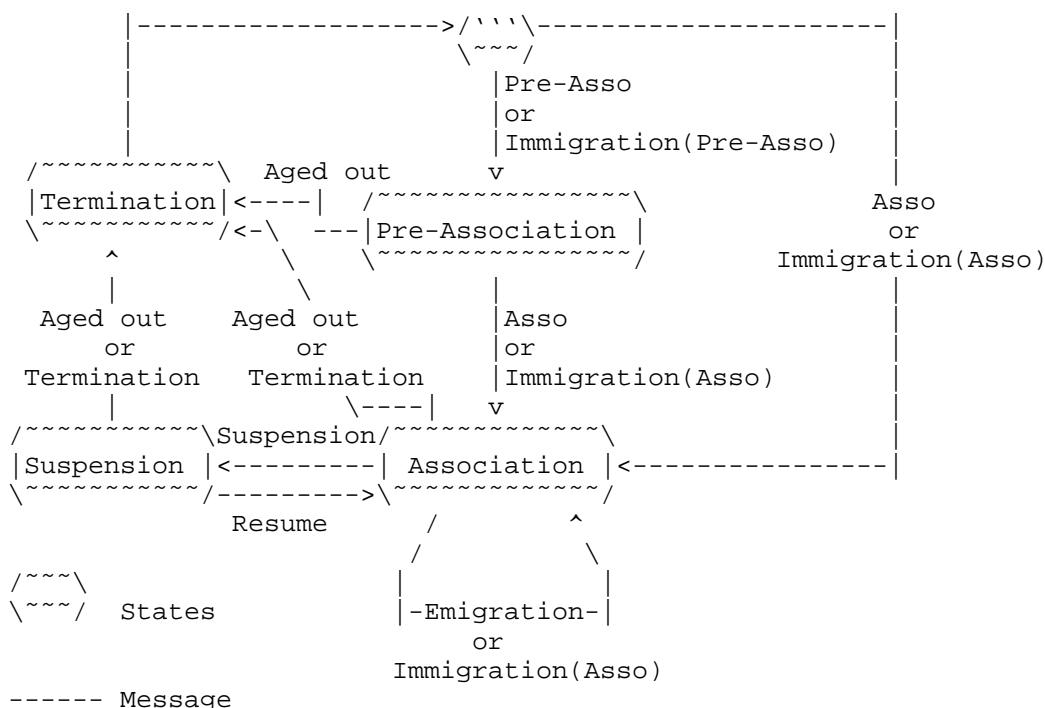


Figure 5: TES to NVE signalling State Machine

#### 4. Security Considerations

There are some considerations on security in [overlay-cp]. Most of the considerations are about mechanism between NVE and external controller, and the attack on underlying networks, which can not be resolved only by the mechanism between TES and NVE. One security issue related to the mechanism between TES and NVE is about the authentication of VM who announces to associate with a particular VN. There is a hypervisor between VMs and NVEs, and both VMs and hypervisor are not always reliable. For example, a poisoned hypervisor may modify the VN Name, or identification for similar intention, in order to associate with a VN that it doesn't belong to.

#### 5. Appendix 1: Mechanism Analysis

##### 5.1. IEEE 802.1Qbg

## 5.1.1.1. Brief Introduction

VDP has four basic TLV types.

- o Pre-Associate: Pre-Associate is used to pre-associate a VSI instance with a bridge port. The bridge validates the request and returns a failure Status in case of errors. Successful pre-association does not imply that the indicated VSI Type will be applied to any traffic flowing through the VSI. The pre-associate enables faster response to an associate, by allowing the bridge to obtain the VSI Type prior to an association.
- o Pre-Associate with resource reservation: Pre-Associate with Resource Reservation involves the same steps as Pre-Associate, but on successful pre-association also reserves resources in the Bridge to prepare for a subsequent Associate request.
- o Associate: The Associate TLV Type creates and activates an association between a VSI instance and a bridge port. The Bridge allocates any required bridge resources for the referenced VSI. The Bridge activates the configuration for the VSI Type ID. This association is then applied to the traffic flow to/from the VSI instance.
- o Deassociate: The de-associate TLV Type is used to remove an association between a VSI instance and a bridge port. Pre-Associated and Associated VSIs can be de-associated. De-associate releases any resources that were reserved as a result of prior Associate or Pre-Associate operations for that VSI instance.

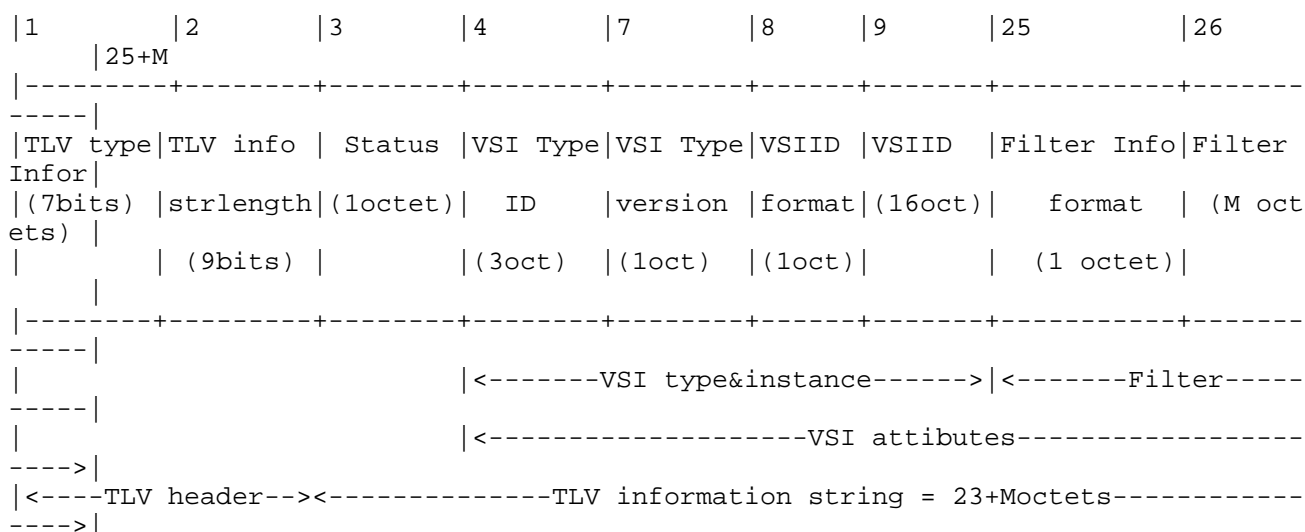


Figure 6: VDP TLV definitions

Some important flag values in VDP request:

- o M-bit (Bit 5): Indicates that the user of the VSI (e.g., the VM) is migrating (M-bit = 1) or provides no guidance on the migration of the user of the VSI (M-bit = 0). The M-bit is used as an indicator relative to the VSI that the user is migrating to.





- o S-bit (Bit 6): Indicates that the VSI user (e.g., the VM) is suspended (S-bit = 1) or provides no guidance as to whether the user of the VSI is suspended (S-bit = 0). A keep-alive Associate request with S-bit = 1 can be sent when the VSI user is suspended. The S-bit is used as an indicator relative to the VSI that the user is migrating from.

The filter information field supports the following format:

- o VID

#of entries (2octets)	PS (1bit)	PCP (3bits)	VID (12bits)
<---Repeated per entry-->			

Figure 7

- o MAC/VID

#of entries (2octets)	MAC address (6 octets)	PS (1bit)	PCP (3bits)	VID (12bits)
<-----Repeated per entry----->				

Figure 8

- o GroupID/VID

#of entries (2octets)	GroupID (4 octets)	PS (1bit)	PCP (3bits)	VID (12bits)
<-----Repeated per entry----->				

Figure 9

- o GroupID/MAC/VID

#of entries (2octets)	GroupID (4 octets)	MAC address (6 octets)	PS (1bit)	PCP (3bits)	VID (12bits)
<-----Repeated per entry----->					

Figure 10

In each format, the null VID can be used in the VDP Request. In this case, the Bridge is expected to supply the corresponding local VID value in the VDP Response.

The VSIID in VDP request that identify a VM can be one of the following format: IPV4 address, IPV6 address, MAC address, UUID or locally defined.

VDP features	Requirements Matching
Pre-Associate/ Pre-Associate with resource reservation/ Associate/ Deassociate	Requirement-1
M-bit/S-bit	Requirement-1
VSI type&instance in VDP request	Requirement-2
Filter Infor	Requirement-3
VID infor in VDP response	Requirement-4
PCP in VDP response	Requirement-5

#### VDP TLV types

### 5.2. BGP

gives a brief analysis on how BGP can be reused for TES and NVE signalling. Please refer to it for more information. [server2nve]

### 5.3. External Controller

## 6. References

### 6.1. Normative Reference

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

[Qbg] "IEEE P802.1Qbg Edge Virtual Bridging".

### 6.2. Informative Reference

[framework]  
 Marc Lasserre, Marc., Balus, Florin., Morin, Thomas., Bitar, Nabil., and Yakov. Rekhter,  
 "draft-ietf-nvo3-framework-00", September 2012.

[overlay-cp]

Kreeger, L., Dutt, D., Narten, T., Black, D., and M.  
Sridharan, "draft-kreeger-nvo3-overlay-cp-00", Jan 2012.

[server2nve]

Kompella, K.,  
"draft-dunbar-nvo3-overlay-mobility-issues-00", July 2012.

[statemigration-framework]

Gu, Y., Shore, M., and S. Sivakumar, "A Framework and  
Problem Statement for Flow-associated Middlebox State  
Migration", October 2012.

#### Authors' Addresses

Gu Yingjie  
Huawei  
No. 101 Software Avenue  
Nanjing, Jiangsu Province 210001  
P.R.China

Phone: +86-25-56625392  
Email: guyingjie@huawei.com

Yizhou Li  
Huawei  
No. 101 Software Avenue  
Nanjing, Jiangsu Province 210001  
P.R.China

Phone:  
Email: liyizhou@huawei.com



Internet Engineering Task Force  
Internet Draft  
Intended status: Informational  
Expires: Jan 2015

Marc Lasserre  
Florin Balus  
Alcatel-Lucent

Thomas Morin  
France Telecom Orange

Nabil Bitar  
Verizon

Yakov Rekhter  
Juniper

July 4, 2014

Framework for DC Network Virtualization  
draft-ietf-nvo3-framework-09.txt

Abstract

This document provides a framework for Data Center (DC) Network Virtualization Overlays (NVO3) and it defines a reference model along with logical components required to design a solution.

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 Jan 4, 2015.

## Copyright Notice

Copyright (c) 2014 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
1.1. General terminology.....	3
1.2. DC network architecture.....	6
2. Reference Models.....	8
2.1. Generic Reference Model.....	8
2.2. NVE Reference Model.....	10
2.3. NVE Service Types.....	10
2.3.1. L2 NVE providing Ethernet LAN-like service.....	11
2.3.2. L3 NVE providing IP/VRF-like service.....	11
2.4. Operational Management Considerations.....	11
3. Functional components.....	12
3.1. Service Virtualization Components.....	12
3.1.1. Virtual Access Points (VAPs).....	12
3.1.2. Virtual Network Instance (VNI).....	12
3.1.3. Overlay Modules and VN Context.....	12
3.1.4. Tunnel Overlays and Encapsulation options.....	13
3.1.5. Control Plane Components.....	14
3.1.5.1. Distributed vs Centralized Control Plane.....	14
3.1.5.2. Auto-provisioning/Service discovery.....	14
3.1.5.3. Address advertisement and tunnel mapping.....	15
3.1.5.4. Overlay Tunneling.....	15
3.2. Multi-homing.....	16
3.3. VM Mobility.....	17
4. Key aspects of overlay networks.....	17
4.1. Pros & Cons.....	17
4.2. Overlay issues to consider.....	19
4.2.1. Data plane vs Control plane driven.....	19
4.2.2. Coordination between data plane and control plane..	19

4.2.3. Handling Broadcast, Unknown Unicast and Multicast (BUM) traffic.....	19
4.2.4. Path MTU.....	20
4.2.5. NVE location trade-offs.....	21
4.2.6. Interaction between network overlays and underlays.....	22
5. Security Considerations.....	22
6. IANA Considerations.....	23
7. References.....	23
7.1. Informative References.....	23
8. Acknowledgments.....	25

## 1. Introduction

This document provides a framework for Data Center (DC) Network Virtualization over Layer3 (L3) tunnels. This framework is intended to aid in standardizing protocols and mechanisms to support large-scale network virtualization for data centers.

[NVOPS] defines the rationale for using overlay networks in order to build large multi-tenant data center networks. Compute, storage and network virtualization are often used in these large data centers to support a large number of communication domains and end systems.

This document provides reference models and functional components of data center overlay networks as well as a discussion of technical issues that have to be addressed.

### 1.1. General terminology

This document uses the following terminology:

**NVO3 Network:** An overlay network that provides a Layer2 (L2) or Layer3 (L3) service to Tenant Systems over an L3 underlay network using the architecture and protocols as defined by the NVO3 Working Group.

**Network Virtualization Edge (NVE).** An NVE is the network entity that sits at the edge of an underlay network and implements L2 and/or L3 network virtualization functions. The network-facing side of the NVE uses the underlying L3 network to tunnel tenant frames to and from other NVEs. The tenant-facing side of the NVE sends and receives Ethernet frames to and from individual Tenant Systems. An NVE could be implemented as part of a virtual switch within a hypervisor, a physical switch or router, a Network Service Appliance, or be split across multiple devices.



**Virtual Network (VN):** A VN is a logical abstraction of a physical network that provides L2 or L3 network services to a set of Tenant Systems. A VN is also known as a Closed User Group (CUG).

**Virtual Network Instance (VNI):** A specific instance of a VN from the perspective of an NVE.

**Virtual Network Context (VN Context) Identifier:** Field in overlay encapsulation header that identifies the specific VN the packet belongs to. The egress NVE uses the VN Context identifier to deliver the packet to the correct Tenant System. The VN Context identifier can be a locally significant identifier or a globally unique identifier.

**Underlay or Underlying Network:** The network that provides the connectivity among NVEs and over which NVO3 packets are tunneled, where an NVO3 packet carries an NVO3 overlay header followed by a tenant packet. The Underlay Network does not need to be aware that it is carrying NVO3 packets. Addresses on the Underlay Network appear as "outer addresses" in encapsulated NVO3 packets. In general, the Underlay Network can use a completely different protocol (and address family) from that of the overlay. In the case of NVO3, the underlay network is IP.

**Data Center (DC):** A physical complex housing physical servers, network switches and routers, network service appliances and networked storage. The purpose of a Data Center is to provide application, compute and/or storage services. One such service is virtualized infrastructure data center services, also known as Infrastructure as a Service.

**Virtual Data Center (Virtual DC):** A container for virtualized compute, storage and network services. A Virtual DC is associated with a single tenant, and can contain multiple VNs and Tenant Systems connected to one or more of these VNs.

**Virtual machine (VM):** A software implementation of a physical machine that runs programs as if they were executing on a physical, non-virtualized machine. Applications (generally) do not know they are running on a VM as opposed to running on a "bare metal" host or server, though some systems provide a para-virtualization environment that allows an operating system or application to be aware of the presence of virtualization for optimization purposes.

**Hypervisor:** Software running on a server that allows multiple VMs to run on the same physical server. The hypervisor manages and provides

shared compute/memory/storage and network connectivity to the VMs that it hosts. Hypervisors often embed a Virtual Switch (see below).

Server: A physical end host machine that runs user applications. A standalone (or "bare metal") server runs a conventional operating system hosting a single-tenant application. A virtualized server runs a hypervisor supporting one or more VMs.

Virtual Switch (vSwitch): A function within a Hypervisor (typically implemented in software) that provides similar forwarding services to a physical Ethernet switch. A vSwitch forwards Ethernet frames between VMs running on the same server, or between a VM and a physical NIC card connecting the server to a physical Ethernet switch or router. A vSwitch also enforces network isolation between VMs that by policy are not permitted to communicate with each other (e.g., by honoring VLANs). A vSwitch may be bypassed when an NVE is enabled on the host server.

Tenant: The customer using a virtual network and any associated resources (e.g., compute, storage and network). A tenant could be an enterprise, or a department/organization within an enterprise.

Tenant System: 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.

Tenant Separation: Tenant Separation refers to isolating traffic of different tenants such that traffic from one tenant is not visible to or delivered to another tenant, except when allowed by policy. Tenant Separation also refers to address space separation, whereby different tenants can use the same address space without conflict.

Virtual Access Points (VAPs): A logical connection point on the NVE for connecting a Tenant System to a virtual network. Tenant Systems connect to VNIs at an NVE through VAPs. VAPs can be physical ports or virtual ports identified through logical interface identifiers (e.g., VLAN ID, internal vSwitch Interface ID connected to a VM).

End Device: A physical device that connects directly to the DC Underlay Network. This is in contrast to a Tenant System, which connects to a corresponding tenant VN. An End Device is administered by the DC operator rather than a tenant, and is part of the DC infrastructure. An End Device may implement NVO3 technology in support of NVO3 functions. Examples of an End Device include hosts (e.g., server or server blade), storage systems (e.g., file servers,

iSCSI storage systems), and network devices (e.g., firewall, load-balancer, IPSec gateway).

Network Virtualization Authority (NVA): Entity that provides reachability and forwarding information to NVEs.

## 1.2. DC network architecture

A generic architecture for Data Centers is depicted in Figure 1:

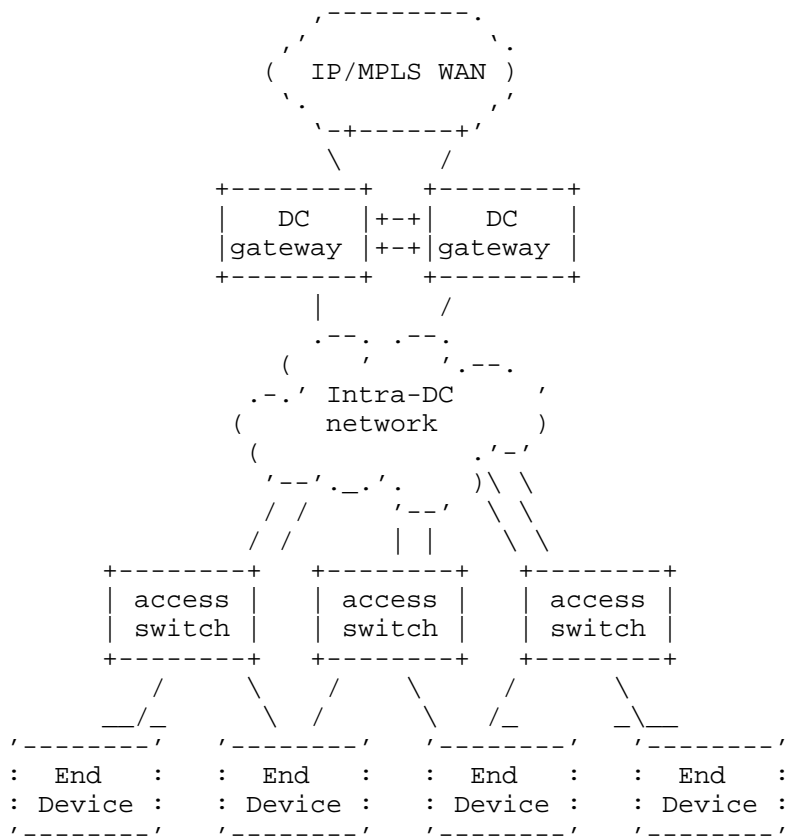


Figure 1 : A Generic Architecture for Data Centers

An example of multi-tier DC network architecture is presented in Figure 1. It provides a view of physical components inside a DC.

A DC network is usually composed of intra-DC networks and network services, and inter-DC network and network connectivity services.

DC networking elements can act as strict L2 switches and/or provide IP routing capabilities, including network service virtualization.

In some DC architectures, some tier layers could provide L2 and/or L3 services. In addition, some tier layers may be collapsed, and Internet connectivity, inter-DC connectivity and VPN support may be handled by a smaller number of nodes. Nevertheless, one can assume that the network functional blocks in a DC fit in the architecture depicted in Figure 1.

The following components can be present in a DC:

- Access switch: Hardware-based Ethernet switch aggregating all Ethernet links from the End Devices in a rack representing the entry point in the physical DC network for the hosts. It may also provide routing functionality, virtual IP network connectivity, or Layer2 tunneling over IP for instance. Access switches are usually multi-homed to aggregation switches in the Intra-DC network. A typical example of an access switch is a Top of Rack (ToR) switch. Other deployment scenarios may use an intermediate Blade Switch before the ToR, or an EoR (End of Row) switch, to provide similar functions to a ToR.
- Intra-DC Network: Network composed of high capacity core nodes (Ethernet switches/routers). Core nodes may provide virtual Ethernet bridging and/or IP routing services.
- DC Gateway (DC GW): Gateway to the outside world providing DC Interconnect and connectivity to Internet and VPN customers. In the current DC network model, this may be simply a router connected to the Internet and/or an IP Virtual Private Network (VPN)/L2VPN PE. Some network implementations may dedicate DC GWs for different connectivity types (e.g., a DC GW for Internet, and another for VPN).

Note that End Devices may be single or multi-homed to access switches.

## 2. Reference Models

### 2.1. Generic Reference Model

Figure 2 depicts a DC reference model for network virtualization overlay where NVEs provide a logical interconnect between Tenant Systems that belong to a specific VN.

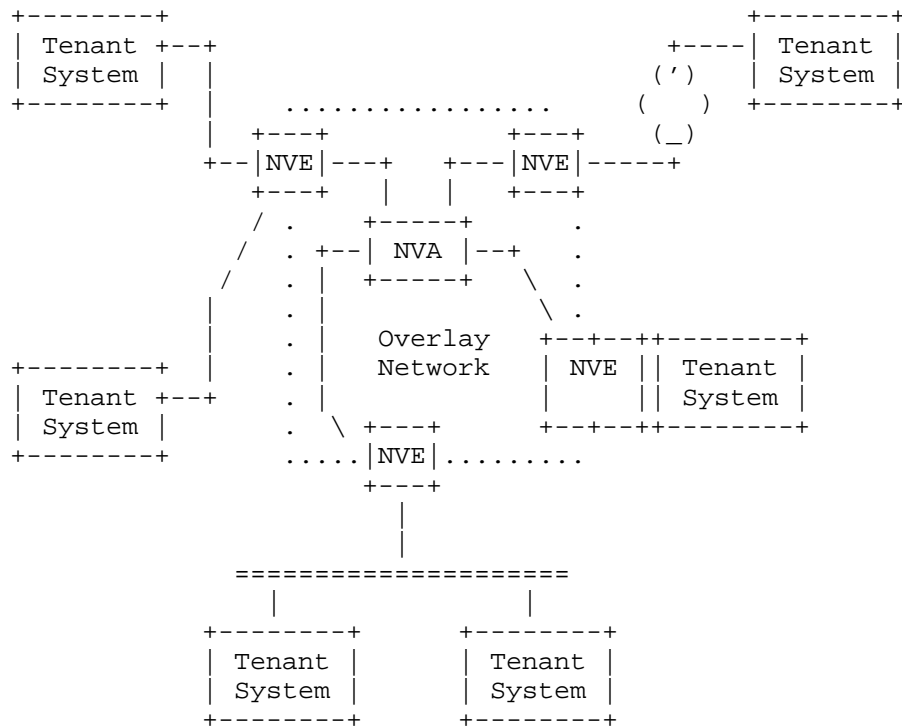


Figure 2 : Generic reference model for DC network virtualization overlay

In order to obtain reachability information, NVEs may exchange information directly between themselves via a control plane protocol. In this case, a control plane module resides in every NVE.

It is also possible for NVEs to communicate with an external Network Virtualization Authority (NVA) to obtain reachability and forwarding information. In this case, a protocol is used between NVEs and NVA(s) to exchange information.

It should be noted that NVAs may be organized in clusters for redundancy and scalability and can appear as one logically centralized controller. In this case, inter-NVA communication is necessary to synchronize state among nodes within a cluster or share information across clusters. The information exchanged between NVAs of the same cluster could be different from the information exchanged across clusters.

A Tenant System can be attached to an NVE in several ways:

- locally, by being co-located in the same End Device
- remotely, via a point-to-point connection or a switched network

When an NVE is co-located with a Tenant System, the state of the Tenant System can be determined without protocol assistance. For instance, the operational status of a VM can be communicated via a local API. When an NVE is remotely connected to a Tenant System, the state of the Tenant System or NVE needs to be exchanged directly or via a management entity, using a control plane protocol or API, or directly via a dataplane protocol.

The functional components in Figure 2 do not necessarily map directly to the physical components described in Figure 1. For example, an End Device can be a server blade with VMs and a virtual switch. A VM can be a Tenant System and the NVE functions may be performed by the host server. In this case, the Tenant System and NVE function are co-located. Another example is the case where the End Device is the Tenant System, and the NVE function can be implemented by the connected ToR. In this case, the Tenant System and NVE function are not co-located.

Underlay nodes utilize L3 technologies to interconnect NVE nodes. These nodes perform forwarding based on outer L3 header information, and generally do not maintain per tenant-service state albeit some applications (e.g., multicast) may require control plane or forwarding plane information that pertain to a tenant, group of

tenants, tenant service or a set of services that belong to one or more tenants. Mechanisms to control the amount of state maintained in the underlay may be needed.

## 2.2. NVE Reference Model

Figure 3 depicts the NVE reference model. One or more VNIs can be instantiated on an NVE. A Tenant System interfaces with a corresponding VNI via a VAP. An overlay module provides tunneling overlay functions (e.g., encapsulation and decapsulation of tenant traffic, tenant identification and mapping, etc.).

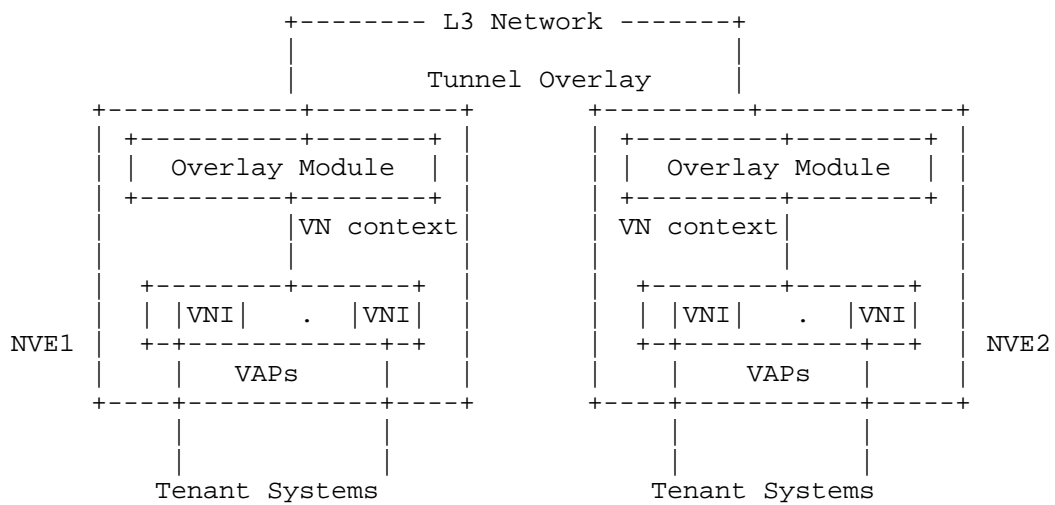


Figure 3 : Generic NVE reference model

Note that some NVE functions (e.g., data plane and control plane functions) may reside in one device or may be implemented separately in different devices.

## 2.3. NVE Service Types

An NVE provides different types of virtualized network services to multiple tenants, i.e. an L2 service or an L3 service. Note that an NVE may be capable of providing both L2 and L3 services for a

tenant. This section defines the service types and associated attributes.

#### 2.3.1. L2 NVE providing Ethernet LAN-like service

An L2 NVE implements Ethernet LAN emulation, an Ethernet based multipoint service similar to an IETF VPLS [RFC4761][RFC4762] or EVPN [EVPN] service, where the Tenant Systems appear to be interconnected by a LAN environment over an L3 overlay. As such, an L2 NVE provides per-tenant virtual switching instance (L2 VNI), and L3 (IP/MPLS) tunneling encapsulation of tenant MAC frames across the underlay. Note that the control plane for an L2 NVE could be implemented locally on the NVE or in a separate control entity.

#### 2.3.2. L3 NVE providing IP/VRF-like service

An L3 NVE provides Virtualized IP forwarding service, similar to IETF IP VPN (e.g., BGP/MPLS IPVPN [RFC4364]) from a service definition perspective. That is, an L3 NVE provides per-tenant forwarding and routing instance (L3 VNI), and L3 (IP/MPLS) tunneling encapsulation of tenant IP packets across the underlay. Note that routing could be performed locally on the NVE or in a separate control entity.

#### 2.4. Operational Management Considerations

NVO3 services are overlay services over an IP underlay.

As far as the IP underlay is concerned, existing IP OAM facilities are used.

With regards to the NVO3 overlay, both L2 and L3 services can be offered. it is expected that existing fault and performance OAM facilities will be used. Sections 4.1. and 4.2.6. below provide further discussion of additional fault and performance management issues to consider.

As far as configuration is concerned, the DC environment is driven by the need to bring new services up rapidly and is typically very dynamic specifically in the context of virtualized services. It is therefore critical to automate the configuration of NVO3 services.



### 3. Functional components

This section decomposes the Network Virtualization architecture into functional components described in Figure 3 to make it easier to discuss solution options for these components.

#### 3.1. Service Virtualization Components

##### 3.1.1. Virtual Access Points (VAPs)

Tenant Systems are connected to VNIs through Virtual Access Points (VAPs).

VAPs can be physical ports or virtual ports identified through logical interface identifiers (e.g., VLAN ID, internal vSwitch Interface ID connected to a VM).

##### 3.1.2. Virtual Network Instance (VNI)

A VNI is a specific VN instance on an NVE. Each VNI defines a forwarding context that contains reachability information and policies.

##### 3.1.3. Overlay Modules and VN Context

Mechanisms for identifying each tenant service are required to allow the simultaneous overlay of multiple tenant services over the same underlay L3 network topology. In the data plane, each NVE, upon sending a tenant packet, must be able to encode the VN Context for the destination NVE in addition to the L3 tunneling information (e.g., source IP address identifying the source NVE and the destination IP address identifying the destination NVE, or MPLS label). This allows the destination NVE to identify the tenant service instance and therefore appropriately process and forward the tenant packet.

The Overlay module provides tunneling overlay functions: tunnel initiation/termination as in the case of stateful tunnels (see Section 3.1.4), and/or simply encapsulation/decapsulation of frames from VAPs/L3 underlay.

In a multi-tenant context, tunneling aggregates frames from/to different VNIs. Tenant identification and traffic demultiplexing are based on the VN Context identifier.

The following approaches can be considered:

- VN Context identifier per Tenant: Globally unique (on a per-DC administrative domain) VN identifier used to identify the corresponding VNI. Examples of such identifiers in existing technologies are IEEE VLAN IDs and ISID tags that identify virtual L2 domains when using IEEE 802.1aq and IEEE 802.1ah, respectively. Note that multiple VN identifiers can belong to a tenant.
- One VN Context identifier per VNI: Each VNI value is automatically generated by the egress NVE, or a control plane associated with that NVE, and usually distributed by a control plane protocol to all the related NVEs. An example of this approach is the use of per VRF MPLS labels in IP VPN [RFC4364]. The VNI value is therefore locally significant to the egress NVE.
- One VN Context identifier per VAP: A value locally significant to an NVE is assigned and usually distributed by a control plane protocol to identify a VAP. An example of this approach is the use of per CE-PE MPLS labels in IP VPN [RFC4364].

Note that when using one VN Context per VNI or per VAP, an additional global identifier (e.g., a VN identifier or name) may be used by the control plane to identify the Tenant context.

#### 3.1.4. Tunnel Overlays and Encapsulation options

Once the VN context identifier is added to the frame, an L3 Tunnel encapsulation is used to transport the frame to the destination NVE.

Different IP tunneling options (e.g., GRE, L2TP, IPSec) and MPLS tunneling can be used. Tunneling could be stateless or stateful. Stateless tunneling simply entails the encapsulation of a tenant packet with another header necessary for forwarding the packet across the underlay (e.g., IP tunneling over an IP underlay). Stateful tunneling on the other hand entails maintaining tunneling state at the tunnel endpoints (i.e., NVEs). Tenant packets on an ingress NVE can then be transmitted over such tunnels to a destination (egress) NVE by encapsulating the packets with a corresponding tunneling header. The tunneling state at the endpoints may be configured or dynamically established. Solutions should specify the tunneling technology used, whether it is stateful or stateless. In this document, however, tunneling and tunneling encapsulation are used interchangeably to simply mean the encapsulation of a tenant packet with a tunneling header necessary to carry the packet between an ingress NVE and an egress NVE across the underlay. It should be noted that stateful tunneling, especially when configuration is involved, does impose management overhead and

scale constraints. When confidentiality is required, the use of opportunistic security [OPPSEC] can be used as a stateless tunneling solution.

### 3.1.5. Control Plane Components

#### 3.1.5.1. Distributed vs Centralized Control Plane

A control/management plane entity can be centralized or distributed. Both approaches have been used extensively in the past. The routing model of the Internet is a good example of a distributed approach. Transport networks have usually used a centralized approach to manage transport paths.

It is also possible to combine the two approaches, i.e., using a hybrid model. A global view of network state can have many benefits but it does not preclude the use of distributed protocols within the network. Centralized models provide a facility to maintain global state, and distribute that state to the network. When used in combination with distributed protocols, greater network efficiencies, improved reliability and robustness can be achieved. Domain and/or deployment specific constraints define the balance between centralized and distributed approaches.

#### 3.1.5.2. Auto-provisioning/Service discovery

NVEs must be able to identify the appropriate VNI for each Tenant System. This is based on state information that is often provided by external entities. For example, in an environment where a VM is a Tenant System, this information is provided by VM orchestration systems, since these are the only entities that have visibility of which VM belongs to which tenant.

A mechanism for communicating this information to the NVE is required. VAPs have to be created and mapped to the appropriate VNI. Depending upon the implementation, this control interface can be implemented using an auto-discovery protocol between Tenant Systems and their local NVE or through management entities. In either case, appropriate security and authentication mechanisms to verify that Tenant System information is not spoofed or altered are required. This is one critical aspect for providing integrity and tenant isolation in the system.

NVEs may learn reachability information to VNIs on other NVEs via a control protocol that exchanges such information among NVEs, or via a management control entity.

### 3.1.5.3. Address advertisement and tunnel mapping

As traffic reaches an ingress NVE on a VAP, a lookup is performed to determine which NVE or local VAP the packet needs to be sent to. If the packet is to be sent to another NVE, the packet is encapsulated with a tunnel header containing the destination information (destination IP address or MPLS label) of the egress NVE. Intermediate nodes (between the ingress and egress NVEs) switch or route traffic based upon the tunnel destination information.

A key step in the above process consists of identifying the destination NVE the packet is to be tunneled to. NVEs are responsible for maintaining a set of forwarding or mapping tables that hold the bindings between destination VM and egress NVE addresses. Several ways of populating these tables are possible: control plane driven, management plane driven, or data plane driven.

When a control plane protocol is used to distribute address reachability and tunneling information, the auto-provisioning/Service discovery could be accomplished by the same protocol. In this scenario, the auto-provisioning/Service discovery could be combined with (be inferred from) the address advertisement and associated tunnel mapping. Furthermore, a control plane protocol that carries both MAC and IP addresses eliminates the need for ARP, and hence addresses one of the issues with explosive ARP handling as discussed in [RFC6820].

### 3.1.5.4. Overlay Tunneling

For overlay tunneling, and dependent upon the tunneling technology used for encapsulating the Tenant System packets, it may be sufficient to have one or more local NVE addresses assigned and used in the source and destination fields of a tunneling encapsulation header. Other information that is part of the tunneling encapsulation header may also need to be configured. In certain cases, local NVE configuration may be sufficient while in other cases, some tunneling related information may need to be shared among NVEs. The information that needs to be shared will be technology dependent. For instance, potential information could include tunnel identity, encapsulation type, and/or tunnel resources. In certain cases, such as when using IP multicast in the underlay, tunnels which interconnect NVEs may need to be established. When tunneling information needs to be exchanged or shared among NVEs, a control plane protocol may be required. For instance, it may be necessary to provide active/standby status

information between NVEs, up/down status information, pruning/grafting information for multicast tunnels, etc.

In addition, a control plane may be required to setup the tunnel path for some tunneling technologies. This applies to both unicast and multicast tunneling.

### 3.2. Multi-homing

Multi-homing techniques can 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.

The NVO3 underlay nodes (i.e. from NVEs to DC GWs) rely on IP routing as the means to re-route traffic upon failures techniques or on MPLS re-rerouting capabilities.

When a Tenant System is co-located with the NVE, the Tenant System is effectively single homed to the NVE via a virtual port. When the Tenant System and the NVE are separated, the Tenant System is connected to the NVE via a logical Layer2 (L2) construct such as a VLAN and it can be multi-homed to various NVEs. An NVE may provide an L2 service to the end system or an L3 service. An NVE may be multi-homed to a next layer in the DC at Layer2 (L2) or Layer3 (L3). When an NVE provides an L2 service and is not co-located with the end system, loop avoidance techniques must be used. Similarly, when the NVE provides L3 service, similar dual-homing techniques can be used. When the NVE provides a L3 service to the end system, it is possible that no dynamic routing protocol is enabled between the end system and the NVE. The end system can be multi-homed to multiple physically-separated L3 NVEs over multiple interfaces. When one of the links connected to an NVE fails, the other interfaces can be used to reach the end system.

External connectivity from a DC can be handled by two or more DC gateways. Each gateway provides access to external networks such as VPNs or the Internet. A gateway may be connected to two or more edge nodes in the external network for redundancy. When a connection to an upstream node is lost, the alternative connection is used and the failed route withdrawn.

### 3.3. VM Mobility

In DC environments utilizing VM technologies, an important feature is that VMs can move from one server to another server in the same or different L2 physical domains (within or across DCs) in a seamless manner.

A VM can be moved from one server to another in stopped or suspended state ("cold" VM mobility) or in running/active state ("hot" VM mobility). With "hot" mobility, VM L2 and L3 addresses need to be preserved. With "cold" mobility, it may be desired to preserve at least VM L3 addresses.

Solutions to maintain connectivity while a VM is moved are necessary in the case of "hot" mobility. This implies that connectivity among VMs is preserved. For instance, for L2 VNs, ARP caches are updated accordingly.

Upon VM mobility, NVE policies that define connectivity among VMs must be maintained.

During VM mobility, it is expected that the path to the VM's default gateway assures adequate QoS to VM applications, i.e. QoS that matches the expected service level agreement for these applications.

## 4. Key aspects of overlay networks

The intent of this section is to highlight specific issues that proposed overlay solutions need to address.

### 4.1. Pros & Cons

An overlay network is a layer of virtual network topology on top of the physical network.

Overlay networks offer the following key advantages:

- Unicast tunneling state management and association of Tenant Systems reachability are handled at the edge of the network (at the NVE). Intermediate transport nodes are unaware of such state. Note that when multicast is enabled in the underlay network to build multicast trees for tenant VNs, there would be more state related to tenants in the underlay core network.
- Tunneling is used to aggregate traffic and hide tenant addresses from the underlay network, and hence offer the

advantage of minimizing the amount of forwarding state required within the underlay network

- Decoupling of the overlay addresses (MAC and IP) used by VMs from the underlay network for tenant separation and separation of the tenant address spaces from the underlay address space.
- Support of a large number of virtual network identifiers

Overlay networks also create several challenges:

- Overlay networks have typically no control of underlay networks and lack underlay network information (e.g. underlay utilization):
  - Overlay networks and/or their associated management entities typically probe the network to measure link or path properties, such as available bandwidth or packet loss rate. It is difficult to accurately evaluate network properties. It might be preferable for the underlay network to expose usage and performance information.
  - Miscommunication or lack of coordination between overlay and underlay networks can lead to an inefficient usage of network resources.
  - When multiple overlays co-exist on top of a common underlay network, the lack of coordination between overlays can lead to performance issues and/or resource usage inefficiencies.
- Traffic carried over an overlay might fail to traverse firewalls and NAT devices.
- Multicast service scalability: Multicast support may be required in the underlay network to address tenant flood containment or efficient multicast handling. The underlay may also be required to maintain multicast state on a per-tenant basis, or even on a per-individual multicast flow of a given tenant. Ingress replication at the NVE eliminates that additional multicast state in the underlay core, but depending on the multicast traffic volume, it may cause inefficient use of bandwidth.

## 4.2. Overlay issues to consider

### 4.2.1. Data plane vs Control plane driven

In the case of an L2 NVE, it is possible to dynamically learn MAC addresses against VAPs. It is also possible that such addresses be known and controlled via management or a control protocol for both L2 NVEs and L3 NVEs. Dynamic data plane learning implies that flooding of unknown destinations be supported and hence implies that broadcast and/or multicast be supported or that ingress replication be used as described in section 4.2.3. Multicasting in the underlay network for dynamic learning may lead to significant scalability limitations. Specific forwarding rules must be enforced to prevent loops from happening. This can be achieved using a spanning tree, a shortest path tree, or a split-horizon mesh.

It should be noted that the amount of state to be distributed is dependent upon network topology and the number of virtual machines. Different forms of caching can also be utilized to minimize state distribution between the various elements. The control plane should not require an NVE to maintain the locations of all the Tenant Systems whose VNs are not present on the NVE. The use of a control plane does not imply that the data plane on NVEs has to maintain all the forwarding state in the control plane.

### 4.2.2. Coordination between data plane and control plane

For an L2 NVE, the NVE needs to be able to determine MAC addresses of the Tenant Systems connected via a VAP. This can be achieved via dataplane learning or a control plane. For an L3 NVE, the NVE needs to be able to determine IP addresses of the Tenant Systems connected via a VAP.

In both cases, coordination with the NVE control protocol is needed such that when the NVE determines that the set of addresses behind a VAP has changed, it triggers the NVE control plane to distribute this information to its peers.

### 4.2.3. Handling Broadcast, Unknown Unicast and Multicast (BUM) traffic

There are several options to support packet replication needed for broadcast, unknown unicast and multicast. Typical methods include:

- Ingress replication



- Use of underlay multicast trees

There is a bandwidth vs state trade-off between the two approaches. Depending upon the degree of replication required (i.e. the number of hosts per group) and the amount of multicast state to maintain, trading bandwidth for state should be considered.

When the number of hosts per group is large, the use of underlay multicast trees may be more appropriate. When the number of hosts is small (e.g. 2-3) and/or the amount of multicast traffic is small, ingress replication may not be an issue.

Depending upon the size of the data center network and hence the number of (S,G) entries, and also the duration of multicast flows, the use of underlay multicast trees can be a challenge.

When flows are well known, it is possible to pre-provision such multicast trees. However, it is often difficult to predict application flows ahead of time, and hence programming of (S,G) entries for short-lived flows could be impractical.

A possible trade-off is to use in the underlay shared multicast trees as opposed to dedicated multicast trees.

#### 4.2.4. Path MTU

When using overlay tunneling, an outer header is added to the original frame. This can cause the MTU of the path to the egress tunnel endpoint to be exceeded.

It is usually not desirable to rely on IP fragmentation for performance reasons. Ideally, the interface MTU as seen by a Tenant System is adjusted such that no fragmentation is needed.

It is possible for the MTU to be configured manually or to be discovered dynamically. Various Path MTU discovery techniques exist in order to determine the proper MTU size to use:

- Classical ICMP-based MTU Path Discovery [RFC1191] [RFC1981]
  - Tenant Systems rely on ICMP messages to discover the MTU of the end-to-end path to its destination. This method is not always possible, such as when traversing middle boxes (e.g. firewalls) which disable ICMP for security reasons

- Extended MTU Path Discovery techniques such as defined in [RFC4821]
- Tenant Systems send probe packets of different sizes, and rely on confirmation of receipt or lack thereof from receivers to allow a sender to discover the MTU of the end-to-end paths.

While it could also be possible to rely on the NVE to perform segmentation and reassembly operations without relying on the Tenant Systems to know about the end-to-end MTU, this would lead to undesired performance and congestion issues as well as significantly increase the complexity of hardware NVEs required for buffering and reassembly logic.

Preferably, the underlay network should be designed in such a way that the MTU can accommodate the extra tunneling and possibly additional NVO3 header encapsulation overhead.

#### 4.2.5. 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 virtual switch or ToR switch and then by a DC gateway. The NVE function can be embedded within any of these elements.

There are several criteria to consider when deciding where the NVE function should happen:

- Processing and memory requirements
  - Datapath (e.g. lookups, filtering, encapsulation/decapsulation)
  - Control plane processing (e.g. routing, signaling, OAM) and where specific control plane functions should be enabled
- FIB/RIB size
- Multicast support
  - Routing/signaling protocols
  - Packet replication capability
  - Multicast FIB
- Fragmentation support

- QoS support (e.g. marking, policing, queuing)
- Resiliency

#### 4.2.6. Interaction between network overlays and underlays

When multiple overlays co-exist on top of a common underlay network, resources (e.g., bandwidth) should be provisioned to ensure that traffic from overlays can be accommodated and QoS objectives can be met. Overlays can have partially overlapping paths (nodes and links).

Each overlay is selfish by nature. It sends traffic so as to optimize its own performance without considering the impact on other overlays, unless the underlay paths are traffic engineered on a per overlay basis to avoid congestion of underlay resources.

Better visibility between overlays and underlays, or generally coordination in placing overlay demand on an underlay network, may be achieved by providing mechanisms to exchange performance and liveness information between the underlay and overlay(s) or the use of such information by a coordination system. Such information may include:

- Performance metrics (throughput, delay, loss, jitter) such as defined in [RFC3148], [RFC2679], [RFC2680], and [RFC3393].
- Cost metrics

### 5. Security Considerations

There are three points-of-view when considering security for NVO3. First, the service offered by a service provider via NVO3 technology to a tenant must meet the mutually agreed security requirements. Second, a network implementing NVO3 must be able to trust the virtual network identity associated with packets received from a tenant. Third, an NVO3 network must consider the security associated with running as an overlay across the underlaying network.

To meet a tenant's security requirements, the NVO3 service must deliver packets from the tenant to the indicated destination(s) in the overlay network and external networks. The NVO3 service provides data confidentiality through data separation. The use of both VNIs and tunneling of tenant traffic by NVEs ensures that NVO3 data is kept in a separate context and thus separated from other tenant traffic. The infrastructure supporting an NVO3 service (e.g.

management systems, NVEs, NVAs, and intermediate underlay networks) should be limited to authorized access so that data integrity can be expected. If a tenant requires that its data be confidential, then the tenant system may choose to encrypt its data before transmission into the NVO3 service.

An NVO3 service must be able to verify the VNI received on a packet from the tenant. To ensure this, not only tenant data but also NVO3 control data must be secured (e.g. control traffic between NVAs and NVEs, between NVAs and between NVEs). Since NVEs and NVAs play a central role in NVO3, it is critical that a secure access to NVEs and NVAs be ensured such that no unauthorized access is possible. As discussed in section 3.1.5.2. , Tenant Systems identification is based upon state that is often provided by management systems (e.g. a VM orchestration system in a virtualized environment). Secure access to such management systems must also be ensured. When an NVE receives data from a Tenant System, the tenant identity needs to be verified in order to guarantee that it is authorized to access the corresponding VN. This can be achieved by identifying incoming packets against specific VAPs in some cases. In other circumstances, authentication may be necessary. Once this verification is done, the packet is allowed into the NVO3 overlay and no integrity protection is provided on the overlay packet encapsulation (e.g. the VNI, destination VNE, etc.).

Since an NVO3 service can run across diverse underlay networks, when the underlay network is not trusted to provide at least data integrity, data encryption is needed to assure correct packet delivery.

It is also desirable to restrict the types of information (e.g. topology information, such as discussed in Section 4.2.6) that can be exchanged between an NVO3 service and underlaying networks based upon their agreed security requirements.

## 6. IANA Considerations

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

## 7. References

### 7.1. Informative References

[EVPN] Sajassi, A. et al, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn (work in progress)

- [NVOPS] Narten, T. et al, "Problem Statement : Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement (work in progress)
- [OPPSEC] Dukhovni, V. "Opportunistic Security: some protection most of the time", draft-dukhovni-opportunistic-security (work in progress)
- [RFC1191] Mogul, J. "Path MTU Discovery", RFC1191, November 1990
- [RFC1981] McCann, J. et al, "Path MTU Discovery for IPv6", RFC1981, August 1996
- [RFC2679] Almes, G. et al, "A One-way Delay Metric for IPPM", RFC2679, September 1999
- [RFC2680] Almes, G. et al, "A One-way Packet Loss Metric for IPPM", RFC2680, September 1999
- [RFC3148] Mathis, M. et al, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", RFC3148, July 2001
- [RFC3393] Demichelis, C. and Chimeto, P., "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC3393, November 2002
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC4761] Kompella, K. et al, "Virtual Private LAN Service (VPLS) Using BGP for auto-discovery and Signaling", RFC4761, January 2007
- [RFC4762] Lasserre, M. et al, "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC4762, January 2007
- [RFC4821] Mathis, M. et al, "Packetization Layer Path MTU Discovery", RFC4821, March 2007
- [RFC6820] Narten, T. et al, "Address Resolution Problems in Large Data Center Networks", RFC6820, January 2013

## 8. Acknowledgments

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

Dimitrios Stiliadis, Rotem Salomonovitch, Lucy Yong, Thomas Narten, Larry Kreeger, David Black.

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

## Authors' Addresses

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

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

Yakov Rekhter  
Juniper  
Email: yakov@juniper.net



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: February 01, 2014

T. Narten, Ed.  
IBM  
E. Gray, Ed.  
Ericsson  
D. Black  
EMC  
L. Fang  
L. Kreeger  
Cisco  
M. Napierala  
AT&T  
July 31, 2013

Problem Statement: Overlays for Network Virtualization  
draft-ietf-nvo3-overlay-problem-statement-04

Abstract

This document describes issues associated with providing multi-tenancy in large data center networks and how these issues may be addressed using an overlay-based network virtualization approach. A key multi-tenancy requirement is traffic isolation, so that one tenant's traffic is not visible to any other tenant. Another requirement is address space isolation, so that different tenants can use the same address space within different virtual networks. Traffic and address space isolation is achieved by assigning one or more virtual networks to each tenant, where traffic within a virtual network can only cross into another virtual network in a controlled fashion (e.g., via a configured router and/or a security gateway). Additional functionality is required to provision virtual networks, associating a virtual machine's network interface(s) with the appropriate virtual network, and maintaining that association as the virtual machine is activated, migrated and/or deactivated. Use of an overlay-based approach enables scalable deployment on large network infrastructures.

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 February 01, 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 . . . . .	5
3. Problem Areas . . . . .	6
3.1. Need For Dynamic Provisioning . . . . .	6
3.2. Virtual Machine Mobility Limitations . . . . .	6
3.3. Inadequate Forwarding Table Sizes . . . . .	7
3.4. Need to Decouple Logical and Physical Configuration . . . . .	7
3.5. Need For Address Separation Between Virtual Networks . . . . .	7
3.6. Need For Address Separation Between Virtual Networks and Infrastructure . . . . .	8
3.7. Optimal Forwarding . . . . .	8
4. Using Network Overlays to Provide Virtual Networks . . . . .	9
4.1. Overview of Network Overlays . . . . .	9
4.2. Communication Between Virtual and Non-virtualized Networks . . . . .	11
4.3. Communication Between Virtual Networks . . . . .	12
4.4. Overlay Design Characteristics . . . . .	12
4.5. Control Plane Overlay Networking Work Areas . . . . .	13
4.6. Data Plane Work Areas . . . . .	14
5. Related IETF and IEEE Work . . . . .	15
5.1. BGP/MPLS IP VPNs . . . . .	15
5.2. BGP/MPLS Ethernet VPNs . . . . .	15
5.3. 802.1 VLANs . . . . .	16
5.4. IEEE 802.1aq - Shortest Path Bridging . . . . .	16

5.5.	VDP . . . . .	17
5.6.	ARMD . . . . .	17
5.7.	TRILL . . . . .	17
5.8.	L2VPNs . . . . .	17
5.9.	Proxy Mobile IP . . . . .	18
5.10.	LISP . . . . .	18
6.	Summary . . . . .	18
7.	Acknowledgments . . . . .	18
8.	Contributors . . . . .	19
9.	IANA Considerations . . . . .	19
10.	Security Considerations . . . . .	19
11.	References . . . . .	20
11.1.	Informative References . . . . .	20
11.2.	Normative References . . . . .	21
Appendix A.	Change Log . . . . .	21
A.1.	Changes From -03 to -04 . . . . .	21
A.2.	Changes From -02 to -03 . . . . .	22
A.3.	Changes From -01 to -02 . . . . .	22
A.4.	Changes From -00 to -01 . . . . .	22
A.5.	Changes from draft-narten-nvo3-overlay-problem- statement-04.txt . . . . .	23
Authors' Addresses	. . . . .	23

## 1. Introduction

Data Centers are increasingly being consolidated and outsourced in an effort to improve the deployment time of applications and reduce operational costs. This coincides with an increasing demand for compute, storage, and network resources from applications. In order to scale compute, storage, and network resources, physical resources are being abstracted from their logical representation, in what is referred to as server, storage, and network virtualization. Virtualization can be implemented in various layers of computer systems or networks.

The demand for server virtualization is increasing in data centers. With server virtualization, each physical server supports multiple virtual machines (VMs), each running its own operating system, middleware and applications. Virtualization is a key enabler of workload agility, i.e., allowing any server to host any application and providing the flexibility of adding, shrinking, or moving services within the physical infrastructure. Server virtualization provides numerous benefits, including higher utilization, increased security, reduced user downtime, reduced power usage, etc.

Multi-tenant data centers are taking advantage of the benefits of server virtualization to provide a new kind of hosting, a virtual hosted data center. Multi-tenant data centers are ones where

individual tenants could belong to a different company (in the case of a public provider) or a different department (in the case of an internal company data center). Each tenant has the expectation of a level of security and privacy separating their resources from those of other tenants. For example, one tenant's traffic must never be exposed to another tenant, except through carefully controlled interfaces, such as a security gateway (e.g., a firewall).

To a tenant, virtual data centers are similar to their physical counterparts, consisting of end stations attached to a network, complete with services such as load balancers and firewalls. But unlike a physical data center, tenant systems connect to a virtual network. To tenant systems, a virtual network looks like a normal network (e.g., providing an ethernet or L3 service), except that the only end stations connected to the virtual network are those belonging to a tenant's specific virtual network.

A tenant is the administrative entity on whose behalf one or more specific virtual network instances and their associated services (whether virtual or physical) are managed. In a cloud environment, a tenant would correspond to the customer that is using a particular virtual network. However, a tenant may also find it useful to create multiple different virtual network instances. Hence, there is a one-to-many mapping between tenants and virtual network instances. A single tenant may operate multiple individual virtual network instances, each associated with a different service.

How a virtual network is implemented does not generally matter to the tenant; what matters is that the service provided (L2 or L3) has the right semantics, performance, etc. It could be implemented via a pure routed network, a pure bridged network or a combination of bridged and routed networks. A key requirement is that each individual virtual network instance be isolated from other virtual network instances, with traffic crossing from one virtual network to another only when allowed by policy.

For data center virtualization, two key issues must be addressed. First, address space separation between tenants must be supported. Second, it must be possible to place (and migrate) VMs anywhere in the data center, without restricting VM addressing to match the subnet boundaries of the underlying data center network.

The document outlines problems encountered in scaling the number of isolated virtual networks in a data center. Furthermore, the document presents issues associated with managing those virtual networks, in relation to operations, such as virtual network creation /deletion and end-node membership change. Finally, the document makes the case that an overlay based approach has a number of

advantages over traditional, non-overlay approaches. The purpose of this document is to identify the set of issues that any solution has to address in building multi-tenant data centers. With this approach, the goal is to allow the construction of standardized, interoperable implementations to allow the construction of multi-tenant data centers.

This document is the problem statement for the "Network Virtualization over L3" (NVO3) Working Group. NVO3 is focused on the construction of overlay networks that operate over an IP (L3) underlay transport network. NVO3 expects to provide both L2 service and IP service to end devices (though perhaps as two different solutions). Some deployments require an L2 service, others an L3 service, and some may require both.

Section 2 gives terminology. Section 3 describes the problem space details. Section 4 describes overlay networks in more detail. Sections 5 and 6 review related and further work, while Section 7 closes with a summary.

## 2. Terminology

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

**Overlay Network:** A Virtual Network in which the separation of tenants is hidden from the underlying physical infrastructure. That is, the underlying transport network does not need to know about tenancy separation to correctly forward traffic. IEEE 802.1 Provider Backbone Bridging (PBB) [IEEE-802.1Q]. is an example of an L2 Overlay Network. PBB uses MAC-in-MAC encapsulation and the underlying transport network forwards traffic using only the B-MAC and B-VID in the outer header. The underlay transport network is unaware of the tenancy separation provided by, for example, a 24-bit I-SID.

**C-VLAN:** This document refers to C-VLANs as implemented by many routers, i.e., an L2 virtual network identified by a C-VID. An end station (e.g., a VM) in this context that is part of an L2 virtual network will effectively belong to a C-VLAN. Within an IEEE 802.1Q-2011 network, other tags may be used as well, but such usage is generally not visible to the end station. Section 5.3 provides more details on VLANs defined by [IEEE-802.1Q].

This document uses the phrase "virtual network instance" with its ordinary meaning to represent an instance of a virtual network. Its usage may differ from the VNI acronym defined in the framework document [I-D.ietf-nvo3-framework]. The VNI acronym is not used in this document.

### 3. Problem Areas

The following subsections describe aspects of multi-tenant data center networking that pose problems for network infrastructure. Different problem aspects may arise based on the network architecture and scale.

#### 3.1. Need For Dynamic Provisioning

Some service providers offer services to multiple customers whereby services are dynamic and the resources assigned to support them must be able to change quickly as demand changes. In current systems, it can be difficult to provision resources for individual tenants (e.g., QoS) in such a way that provisioned properties migrate automatically when services are dynamically moved around within the data center to optimize workloads.

#### 3.2. Virtual Machine Mobility Limitations

A key benefit of server virtualization is virtual machine (VM) mobility. A VM can be migrated from one server to another, live, i.e., while continuing to run and without needing to shut it down and restart it at the new location. A key requirement for live migration is that a VM retain critical network state at its new location, including its IP and MAC address(es). Preservation of MAC addresses may be necessary, for example, when software licenses are bound to MAC addresses. More generally, any change in the VM's MAC addresses resulting from a move would be visible to the VM and thus potentially result in unexpected disruptions. Retaining IP addresses after a move is necessary to prevent existing transport connections (e.g., TCP) from breaking and needing to be restarted.

In data center networks, servers are typically assigned IP addresses based on their physical location, for example based on the Top of Rack (ToR) switch for the server rack or the C-VLAN configured to the server. Servers can only move to other locations within the same IP subnet. This constraint is not problematic for physical servers, which move infrequently, but it restricts the placement and movement of VMs within the data center. Any solution for a scalable multi-tenant data center must allow a VM to be placed (or moved) anywhere within the data center, without being constrained by the subnet boundary concerns of the host servers.

### 3.3. Inadequate Forwarding Table Sizes

Today's virtualized environments place additional demands on the forwarding tables of forwarding nodes in the physical infrastructure. The core problem is that location independence results in specific end state information being propagated into the forwarding system (e.g., /32 host routes in L3 networks, or MAC addresses in L2 networks). In L2 networks, for instance, instead of just one address per server, the network infrastructure may have to learn addresses of the individual VMs (which could range in the 100s per server). This increases the demand on a forwarding node's table capacity compared to non-virtualized environments.

### 3.4. Need to Decouple Logical and Physical Configuration

Data center operators must be able to achieve high utilization of server and network capacity. For efficient and flexible allocation, operators should be able to spread a virtual network instance across servers in any rack in the data center. It should also be possible to migrate compute workloads to any server anywhere in the network while retaining the workload's addresses.

In networks of many types (e.g., IP subnets, MPLS VPNs, VLANs, etc.) moving servers elsewhere in the network may require expanding the scope of a portion of the network (e.g., subnet, VPN, VLAN, etc.) beyond its original boundaries. While this can be done, it requires potentially complex network configuration changes and may (in some cases - e.g., a VLAN or L2VPN) conflict with the desire to bound the size of broadcast domains. In addition, when VMs migrate, the physical network (e.g., access lists) may need to be reconfigured which can be time consuming and error prone.

An important use case is cross-pod expansion. A pod typically consists of one or more racks of servers with associated network and storage connectivity. A tenant's virtual network may start off on a pod and, due to expansion, require servers/VMs on other pods, especially the case when other pods are not fully utilizing all their resources. This use case requires that virtual networks span multiple pods in order to provide connectivity to all of its tenant's servers/VMs. Such expansion can be difficult to achieve when tenant addressing is tied to the addressing used by the underlay network or when the expansion requires that the scope of the underlying C-VLAN expand beyond its original pod boundary.

### 3.5. Need For Address Separation Between Virtual Networks

Individual tenants need control over the addresses they use within a virtual network. But it can be problematic when different tenants

want to use the same addresses, or even if the same tenant wants to reuse the same addresses in different virtual networks. Consequently, virtual networks must allow tenants to use whatever addresses they want without concern for what addresses are being used by other tenants or other virtual networks.

### 3.6. Need For Address Separation Between Virtual Networks and Infrastructure

As in the previous case, a tenant needs to be able to use whatever addresses it wants in a virtual network independent of what addresses the underlying data center network is using. Tenants (and the underlay infrastructure provider) should be able use whatever addresses make sense for them, without having to worry about address collisions between addresses used by tenants and those used by the underlay data center network.

### 3.7. Optimal Forwarding

Another problem area relates to the optimal forwarding of traffic between peers that are not connected to the same virtual network. Such forwarding happens when a host on a virtual network communicates with a host not on any virtual network (e.g., an Internet host) as well as when a host on a virtual network communicates with a host on a different virtual network. A virtual network may have two (or more) gateways for forwarding traffic onto and off of the virtual network and the optimal choice of which gateway to use may depend on the set of available paths between the communicating peers. The set of available gateways may not be equally "close" to a given destination. The issue appears both when a VM is initially instantiated on a virtual network or when a VM migrates or is moved to a different location. After a migration, for instance, a VM's best-choice gateway for such traffic may change, i.e., the VM may get better service by switching to the "closer" gateway, and this may improve the utilization of network resources.

IP implementations in network endpoints typically do not distinguish between multiple routers on the same subnet - there may only be a single default gateway in use, and any use of multiple routers usually considers all of them to be one-hop away. Routing protocol functionality is constrained by the requirement to cope with these endpoint limitations - for example VRRP has one router serve as the master to handle all outbound traffic. This problem can be particularly acute when the virtual network spans multiple data centers, as a VM is likely to receive significantly better service when forwarding external traffic through a local router by comparison to using a router at a remote data center.

The optimal forwarding problem applies to both outbound and inbound traffic. For outbound traffic, the choice of outbound router determines the path of outgoing traffic from the VM, which may be sub-optimal after a VM move. For inbound traffic, the location of the VM within the IP subnet for the VM is not visible to the routers beyond the virtual network. Thus, the routing infrastructure will have no information as to which of the two externally visible gateways leading into the virtual network would be the better choice for reaching a particular VM.

The issue is further complicated when middleboxes (e.g., load-balancers, firewalls, etc.) must be traversed. Middle boxes may have session state that must be preserved for ongoing communication, and traffic must continue to flow through the middle box, regardless of which router is "closest".

#### 4. Using Network Overlays to Provide Virtual Networks

Virtual Networks are used to isolate a tenant's traffic from that of other tenants (or even traffic within the same tenant network that requires isolation). There are two main characteristics of virtual networks:

1. Virtual networks isolate the address space used in one virtual network from the address space used by another virtual network. The same network addresses may be used in different virtual networks at the same time. In addition, the address space used by a virtual network is independent from that used by the underlying physical network.
2. Virtual Networks limit the scope of packets sent on the virtual network. Packets sent by Tenant Systems attached to a virtual network are delivered as expected to other Tenant Systems on that virtual network and may exit a virtual network only through controlled exit points such as a security gateway. Likewise, packets sourced from outside of the virtual network may enter the virtual network only through controlled entry points, such as a security gateway.

##### 4.1. Overview of Network Overlays

To address the problems described in Section 3, a network overlay approach can be used.

The idea behind an overlay is quite straightforward. Each virtual network instance is implemented as an overlay. The original packet is encapsulated by the first-hop network device, called a Network Virtualization Edge (NVE), and tunneled to a remote NVE. The



encapsulation identifies the destination of the device that will perform the decapsulation (i.e., the egress NVE for the tunneled packet) before delivering the original packet to the endpoint. The rest of the network forwards the packet based on the encapsulation header and can be oblivious to the payload that is carried inside.

Overlays are based on what is commonly known as a "map-and-encap" architecture. When processing and forwarding packets, three distinct and logically separable steps take place:

1. The first-hop overlay device implements a mapping operation that determines where the encapsulated packet should be sent to reach its intended destination VM. Specifically, the mapping function maps the destination address (either L2 or L3) of a packet received from a VM into the corresponding destination address of the egress NVE device. The destination address will be the underlay address of the NVE device doing the decapsulation and is an IP address.
2. Once the mapping has been determined, the ingress overlay NVE device encapsulates the received packet within an overlay header.
3. The final step is to actually forward the (now encapsulated) packet to its destination. The packet is forwarded by the underlay (i.e., the IP network) based entirely on its outer address. Upon receipt at the destination, the egress overlay NVE device decapsulates the original packet and delivers it to the intended recipient VM.

Each of the above steps is logically distinct, though an implementation might combine them for efficiency or other reasons. It should be noted that in L3 BGP/VPN terminology, the above steps are commonly known as "forwarding" or "virtual forwarding".

The first hop network NVE device can be a traditional switch or router or the virtual switch residing inside a hypervisor. Furthermore, the endpoint can be a VM or it can be a physical server. Examples of architectures based on network overlays include BGP/MPLS VPNs [RFC4364], TRILL [RFC6325], LISP [RFC6830], and Shortest Path Bridging (SPB) [SPB].

In the data plane, an overlay header provides a place to carry either the virtual network identifier, or an identifier that is locally-significant to the edge device. In both cases, the identifier in the overlay header specifies which specific virtual network the data packet belongs to. Since both routed and bridged semantics can be supported by a virtual data center, the original packet carried within the overlay header can be an Ethernet frame or just the IP packet.

A key aspect of overlays is the decoupling of the "virtual" MAC and/or IP addresses used by VMs from the physical network infrastructure and the infrastructure IP addresses used by the data center. If a VM changes location, the overlay edge devices simply update their mapping tables to reflect the new location of the VM within the data center's infrastructure space. Because an overlay network is used, a VM can now be located anywhere in the data center that the overlay reaches without regards to traditional constraints imposed by the underlay network such as the C-VLAN scope, or the IP subnet scope.

Multi-tenancy is supported by isolating the traffic of one virtual network instance from traffic of another. Traffic from one virtual network instance cannot be delivered to another instance without (conceptually) exiting the instance and entering the other instance via an entity (e.g., a gateway) that has connectivity to both virtual network instances. Without the existence of a gateway entity, tenant traffic remains isolated within each individual virtual network instance.

Overlays are designed to allow a set of VMs to be placed within a single virtual network instance, whether that virtual network provides a bridged network or a routed network.

#### 4.2. Communication Between Virtual and Non-virtualized Networks

Not all communication will be between devices connected to virtualized networks. Devices using overlays will continue to access devices and make use of services on non-virtualized networks, whether in the data center, the public Internet, or at remote/branch campuses. Any virtual network solution must be capable of interoperating with existing routers, VPN services, load balancers, intrusion detection services, firewalls, etc. on external networks.

Communication between devices attached to a virtual network and devices connected to non-virtualized networks is handled architecturally by having specialized gateway devices that receive packets from a virtualized network, decapsulate them, process them as regular (i.e., non-virtualized) traffic, and finally forward them on to their appropriate destination (and vice versa).

A wide range of implementation approaches are possible. Overlay gateway functionality could be combined with other network functionality into a network device that implements the overlay functionality, and then forwards traffic between other internal components that implement functionality such as full router service, load balancing, firewall support, VPN gateway, etc.

#### 4.3. Communication Between Virtual Networks

Communication between devices on different virtual networks is handled architecturally by adding specialized interconnect functionality among the otherwise isolated virtual networks. For a virtual network providing an L2 service, such interconnect functionality could be IP forwarding configured as part of the "default gateway" for each virtual network. For a virtual network providing L3 service, the interconnect functionality could be IP forwarding configured as part of routing between IP subnets or it can be based on configured inter-virtual-network traffic policies. In both cases, the implementation of the interconnect functionality could be distributed across the NVEs and could be combined with other network functionality (e.g., load balancing, firewall support) that is applied to traffic forwarded between virtual networks.

#### 4.4. Overlay Design Characteristics

Below are some of the characteristics of environments that must be taken into account by the overlay technology.

1. Highly distributed systems: The overlay should work in an environment where there could be many thousands of access switches (e.g., residing within the hypervisors) and many more Tenant Systems (e.g., VMs) connected to them. This leads to a distributed mapping system that puts a low overhead on the overlay tunnel endpoints.
2. Many highly distributed virtual networks with sparse membership: Each virtual network could be highly dispersed inside the data center. Also, along with expectation of many virtual networks, the number of end systems connected to any one virtual network is expected to be relatively low; Therefore, the percentage of NVEs participating in any given virtual network would also be expected to be low. For this reason, efficient delivery of multi-destination traffic within a virtual network instance should be taken into consideration.

3. Highly dynamic Tenant Systems: Tenant Systems connected to virtual networks can be very dynamic, both in terms of creation/deletion/power-on/off and in terms of mobility from one access device to another.
4. Be incrementally deployable, without necessarily requiring major upgrade of the entire network: The first hop device (or end system) that adds and removes the overlay header may require new software and may require new hardware (e.g., for improved performance). But the rest of the network should not need to change just to enable the use of overlays.
5. Work with existing data center network deployments without requiring major changes in operational or other practices: For example, some data centers have not enabled multicast beyond link-local scope. Overlays should be capable of leveraging underlay multicast support where appropriate, but not require its enablement in order to use an overlay solution.
6. Network infrastructure administered by a single administrative domain: This is consistent with operation within a data center, and not across the Internet.

#### 4.5. Control Plane Overlay Networking Work Areas

There are three specific and separate potential work areas in the area of control plane protocols needed to realize an overlay solution. The areas correspond to different possible "on-the-wire" protocols, where distinct entities interact with each other.

One area of work concerns the address dissemination protocol an NVE uses to build and maintain the mapping tables it uses to deliver encapsulated packets to their proper destination. One approach is to build mapping tables entirely via learning (as is done in 802.1 networks). Another approach is to use a specialized control plane protocol. While there are some advantages to using or leveraging an existing protocol for maintaining mapping tables, the fact that large numbers of NVE's will likely reside in hypervisors places constraints on the resources (cpu and memory) that can be dedicated to such functions.

From an architectural perspective, one can view the address mapping dissemination problem as having two distinct and separable components. The first component consists of a back-end Network Virtualization Authority (NVA) that is responsible for distributing and maintaining the mapping information for the entire overlay system. For this document, we use the term NVA to refer to an entity that supplies answers, without regard to how it knows the answers it

is providing. The second component consists of the on-the-wire protocols an NVE uses when interacting with the NVA.

The back-end NVA could provide high performance, high resiliency, failover, etc. and could be implemented in significantly different ways. For example, one model uses a traditional, centralized "directory-based" database, using replicated instances for reliability and failover. A second model involves using and possibly extending an existing routing protocol (e.g., BGP, IS-IS, etc.). To support different architectural models, it is useful to have one standard protocol for the NVE-NVA interaction while allowing different protocols and architectural approaches for the NVA itself. Separating the two allows NVEs to transparently interact with different types of NVAs, i.e., either of the two architectural models described above. Having separate protocols could also allow for a simplified NVE that only interacts with the NVA for the mapping table entries it needs and allows the NVA (and its associated protocols) to evolve independently over time with minimal impact to the NVEs.

A third work area considers the attachment and detachment of VMs (or Tenant Systems [I-D.ietf-nvo3-framework] more generally) from a specific virtual network instance. When a VM attaches, the NVE associates the VM with a specific overlay for the purposes of tunneling traffic sourced from or destined to the VM. When a VM disconnects, the NVE should notify the NVA that the Tenant System to NVE address mapping is no longer valid. In addition, if this VM was the last remaining member of the virtual network, then the NVE can also terminate any tunnels used to deliver tenant multi-destination packets within the VN to the NVE. In the case where an NVE and hypervisor are on separate physical devices separated by an access network, a standardized protocol may be needed.

In summary, there are three areas of potential work. The first area concerns the implementation of the NVA function itself and any protocols it needs (e.g., if implemented in a distributed fashion). A second area concerns the interaction between the NVA and NVEs. The third work area concerns protocols associated with attaching and detaching a VM from a particular virtual network instance. All three work areas are important to the development of scalable, interoperable solutions.

#### 4.6. Data Plane Work Areas

The data plane carries encapsulated packets for Tenant Systems. The data plane encapsulation header carries a VN Context identifier [I-D.ietf-nvo3-framework] for the virtual network to which the data packet belongs. Numerous encapsulation or tunneling protocols already exist that can be leveraged. In the absence of strong and

compelling justification, it would not seem necessary or helpful to develop yet another encapsulation format just for NVO3.

## 5. Related IETF and IEEE Work

The following subsections discuss related IETF and IEEE work. The items are not meant to provide complete coverage of all IETF and IEEE data center related work, nor should the descriptions be considered comprehensive. Each area aims to address particular limitations of today's data center networks. In all areas, scaling is a common theme as are multi-tenancy and VM mobility. Comparing and evaluating the work result and progress of each work area listed is out of scope of this document. The intent of this section is to provide a reference to the interested readers. Note that NVO3 is scoped to running over an IP/L3 underlay network.

### 5.1. BGP/MPLS IP VPNs

BGP/MPLS IP VPNs [RFC4364] support multi-tenancy, VPN traffic isolation, address overlapping and address separation between tenants and network infrastructure. The BGP/MPLS control plane is used to distribute the VPN labels and the tenant IP addresses that identify the tenants (or to be more specific, the particular VPN/virtual network) and tenant IP addresses. Deployment of enterprise L3 VPNs has been shown to scale to thousands of VPNs and millions of VPN prefixes. BGP/MPLS IP VPNs are currently deployed in some large enterprise data centers. The potential limitation for deploying BGP/MPLS IP VPNs in data center environments is the practicality of using BGP in the data center, especially reaching into the servers or hypervisors. There may be computing work force skill set issues, equipment support issues, and potential new scaling challenges. A combination of BGP and lighter weight IP signaling protocols, e.g., XMPP, have been proposed to extend the solutions into DC environment [I-D.ietf-l3vpn-end-system], while taking advantage of built-in VPN features with its rich policy support; it is especially useful for inter-tenant connectivity.

### 5.2. BGP/MPLS Ethernet VPNs

Ethernet Virtual Private Networks (E-VPNs) [I-D.ietf-l2vpn-evpn] provide an emulated L2 service in which each tenant has its own Ethernet network over a common IP or MPLS infrastructure. A BGP/MPLS control plane is used to distribute the tenant MAC addresses and the MPLS labels that identify the tenants and tenant MAC addresses. Within the BGP/MPLS control plane a 32-bit Ethernet Tag is used to identify the broadcast domains (VLANs) associated with a given L2 VLAN service instance and these Ethernet tags are mapped to VLAN IDs understood by the tenant at the service edges. This means that any

customer site VLAN based limitation is associated with an individual tenant service edge, enabling a much higher level of scalability. Interconnection between tenants is also allowed in a controlled fashion.

VM Mobility [I-D.raggarwa-data-center-mobility] introduces the concept of a combined L2/L3 VPN service in order to support the mobility of individual Virtual Machines (VMs) between Data Centers connected over a common IP or MPLS infrastructure.

### 5.3. 802.1 VLANs

VLANs are a well understood construct in the networking industry, providing an L2 service via a physical network in which tenant forwarding information is part of the physical network infrastructure. A VLAN is an L2 bridging construct that provides the semantics of virtual networks mentioned above: a MAC address can be kept unique within a VLAN, but it is not necessarily unique across VLANs. Traffic scoped within a VLAN (including broadcast and multicast traffic) can be kept within the VLAN it originates from. Traffic forwarded from one VLAN to another typically involves router (L3) processing. The forwarding table look up operation may be keyed on {VLAN, MAC address} tuples.

VLANs are a pure L2 bridging construct and VLAN identifiers are carried along with data frames to allow each forwarding point to know what VLAN the frame belongs to. Various types of VLANs are available today, which can be used for network virtualization even together. The C-VLAN, S-VLAN and B-VLAN IDs [IEEE-802.1Q] are 12 bits. The 24-bit I-SID [SPB] allows the support of more than 16 million virtual networks.

### 5.4. IEEE 802.1aq - Shortest Path Bridging

Shortest Path Bridging (SPB) [SPB] is an IS-IS based overlay that operates over L2 Ethernets. SPB supports multi-pathing and addresses a number of shortcomings in the original Ethernet Spanning Tree Protocol. Shortest Path Bridging Mac (SPBM) uses IEEE 802.1ah PBB (MAC-in-MAC) encapsulation and supports a 24-bit I-SID, which can be used to identify virtual network instances. SPBM provides multi-pathing and supports easy virtual network creation or update.

SPBM extends IS-IS in order to perform link-state routing among core SPBM nodes, obviating the need for learning for communication among core SPBM nodes. Learning is still used to build and maintain the mapping tables of edge nodes to encapsulate Tenant System traffic for transport across the SPBM core.

SPB is compatible with all other 802.1 standards thus allows leveraging of other features, e.g., VSI Discovery Protocol (VDP), OAM or scalability solutions.

#### 5.5. VDP

VDP is the Virtual Station Interface (VSI) Discovery and Configuration Protocol specified by IEEE P802.1Qbg [IEEE-802.1Qbg]. VDP is a protocol that supports the association of a VSI with a port. VDP is run between the end system (e.g., a hypervisor) and its adjacent switch, i.e., the device on the edge of the network. VDP is used for example to communicate to the switch that a Virtual Machine (Virtual Station) is moving, i.e., designed for VM migration.

#### 5.6. ARMD

The Address Resolution for Massive numbers of hosts in the Data center (ARMD) WG examined data center scaling issues with a focus on address resolution and developed a problem statement document [RFC6820]. While an overlay-based approach may address some of the "pain points" that were raised in ARMD (e.g., better support for multi-tenancy), analysis will be needed to understand the scaling tradeoffs of an overlay based approach compared with existing approaches. On the other hand, existing IP-based approaches such as proxy ARP may help mitigate some concerns.

#### 5.7. TRILL

TRILL is a network protocol that provides an Ethernet L2 service to end systems and is designed to operate over any L2 link type. TRILL establishes forwarding paths using IS-IS routing and encapsulates traffic within its own TRILL header. TRILL as originally defined, supports only the standard (and limited) 12-bit C-VID identifier. Work to extend TRILL to support more than 4094 VLANs has recently completed and is defined in [I-D.ietf-trill-fine-labeling]

#### 5.8. L2VPNs

The IETF has specified a number of approaches for connecting L2 domains together as part of the L2VPN Working Group. That group, however has historically been focused on Provider-provisioned L2 VPNs, where the service provider participates in management and provisioning of the VPN. In addition, much of the target environment for such deployments involves carrying L2 traffic over WANs. Overlay approaches as discussed in this document are intended be used within data centers where the overlay network is managed by the data center operator, rather than by an outside party. While overlays can run across the Internet as well, they will extend well into the data



center itself (e.g., up to and including hypervisors) and include large numbers of machines within the data center itself.

Other L2VPN approaches, such as L2TP [RFC3931] require significant tunnel state at the encapsulating and decapsulating end points. Overlays require less tunnel state than other approaches, which is important to allow overlays to scale to hundreds of thousands of end points. It is assumed that smaller switches (i.e., virtual switches in hypervisors or the adjacent devices to which VMs connect) will be part of the overlay network and be responsible for encapsulating and decapsulating packets.

#### 5.9. Proxy Mobile IP

Proxy Mobile IP [RFC5213] [RFC5844] makes use of the GRE Key Field [RFC5845] [RFC6245], but not in a way that supports multi-tenancy.

#### 5.10. LISP

LISP[RFC6830] essentially provides an IP over IP overlay where the internal addresses are end station Identifiers and the outer IP addresses represent the location of the end station within the core IP network topology. The LISP overlay header uses a 24-bit Instance ID used to support overlapping inner IP addresses.

### 6. Summary

This document has argued that network virtualization using overlays addresses a number of issues being faced as data centers scale in size. In addition, careful study of current data center problems is needed for development of proper requirements and standard solutions.

This document identified three potential control protocol work areas. The first involves a backend Network Virtualization Authority and how it learns and distributes the mapping information NVEs use when processing tenant traffic. A second involves the protocol an NVE would use to communicate with the backend NVA to obtain the mapping information. The third potential work concerns the interactions that take place when a VM attaches or detaches from a specific virtual network instance.

There are a number of approaches that provide some if not all of the desired semantics of virtual networks. Each approach needs to be analyzed in detail to assess how well it satisfies the requirements.

### 7. Acknowledgments

Helpful comments and improvements to this document have come from Lou Berger, John Drake, Ilango Ganga, Ariel Hendel, Vinit Jain, Petr Lapukhov, Thomas Morin, Benson Schliesser, Qin Wu, Xiaohu Xu, Lucy Yong and many others on the NVO3 mailing list.

Special thanks to Janos Farkas for his persistence and numerous detailed comments related to the lack of precision in the text relating to IEEE 802.1 technologies.

#### 8. Contributors

Dinesh Dutt and Murari Sridharin were original co-authors of the Internet-Draft that led to the BoF that formed the NVO3 WG. That original draft eventually became the basis for the WG Problem Statement document.

#### 9. IANA Considerations

This memo includes no request to IANA.

#### 10. Security Considerations

Because this document describes the problem space associated with the need for virtualization of networks in complex, large-scale, data-center networks, it does not itself introduce any security risks. However, it is clear that security concerns need to be a consideration of any solutions proposed to address this problem space.

Solutions will need to address both data plane and control plane security concerns.

In the data plane, isolation of virtual network traffic from other virtual networks is a primary concern - for NVO3, this isolation may be based on VN identifiers that are not involved in underlay network packet forwarding between overlay edges (NVEs). This reduces the underlay network's role in isolating virtual networks by comparison to approaches where VN identifiers are involved in packet forwarding (e.g. - 802.1 VLANs - see Section 5.3).

In addition to isolation, assurances against spoofing, snooping, transit modification and denial of service are examples of other important data plane considerations. Some limited environments may even require confidentiality.

In the control plane, the primary security concern is ensuring that an unauthorized party does not compromise the control plane protocol in ways that improperly impact the data plane. Some environments may also be concerned about confidentiality of the control plane.

More generally, denial of service concerns may also be an consideration. For example, a tenant on one virtual network could consume excessive network resources in a way that degrades services for other tenants on other virtual networks.

## 11. References

### 11.1. Informative References

[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-l3vpn-end-system]

Marques, P., Fang, L., Pan, P., Shukla, A., Napierala, M., and N. Bitar, "BGP-signaled end-system IP/VPNs.", draft-ietf-l3vpn-end-system-01 (work in progress), April 2013.

[I-D.ietf-trill-fine-labeling]

Eastlake, D., Zhang, M., Agarwal, P., Perlman, R., and D. Dutt, "TRILL (Transparent Interconnection of Lots of Links): Fine-Grained Labeling", draft-ietf-trill-fine-labeling-07 (work in progress), May 2013.

[I-D.raggarwa-data-center-mobility]

Aggarwal, R., Rekhter, Y., Henderickx, W., Shekhar, R., Fang, L., and A. Sajassi, "Data Center Mobility based on E-VPN, BGP/MPLS IP VPN, IP Routing and NHRP", draft-raggarwa-data-center-mobility-05 (work in progress), June 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.

[IEEE-802.1Qbg]

IEEE 802.1Qbg-2012, ., "IEEE standard for local and metropolitan area networks: Media access control (MAC) bridges and virtual bridged local area networks -- Amendment 21: Edge virtual bridging, ", July 2012.

- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.
- [RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, May 2010.
- [RFC5845] Muhanna, A., Khalil, M., Gundavelli, S., and K. Leung, "Generic Routing Encapsulation (GRE) Key Option for Proxy Mobile IPv6", RFC 5845, June 2010.
- [RFC6245] Yegani, P., Leung, K., Lior, A., Chowdhury, K., and J. Navali, "Generic Routing Encapsulation (GRE) Key Extension for Mobile IPv4", RFC 6245, May 2011.
- [RFC6325] Perlman, R., Eastlake, D., Dutt, D., Gai, S., and A. Ghanwani, "Routing Bridges (RBridges): Base Protocol Specification", RFC 6325, July 2011.
- [RFC6820] Narten, T., Karir, M., and I. Foo, "Address Resolution Problems in Large Data Center Networks", RFC 6820, January 2013.
- [RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.
- [SPB] IEEE 802.1aq, ., "IEEE standard for local and metropolitan area networks: Media access control (MAC) bridges and virtual bridged local area networks -- Amendment 20: Shortest path bridging, ", June 2012.

## 11.2. Normative References

- [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.

## Appendix A. Change Log

### A.1. Changes From -03 to -04

Changes in response to IESG review; use rcsdiff to see changes.

A.2. Changes From -02 to -03

1. Comments from Janos Farkas, including:
  - \* Defined C-VLAN and changed VLAN -> C-VLAN where appropriate.
  - \* Improved references to IEEE work.
  - \* Removed Section "Further Work".
2. Improved first paragraph in "Optimal Forwarding" Section (per Qin Wu).
3. Replaced "oracle" term with Network Virtualization Authority, to match terminology discussion on list.
4. Reduced number of authors to 6. Still above the usual guideline of 5, but chairs will ask for exception in this case.

A.3. Changes From -01 to -02

1. Security Considerations changes (Lou Berger)
2. Changes to section on Optimal Forwarding (Xuxiaohu)
3. More wording improvements in L2 details (Janos Farkas)
4. References to ARMD and LISP documents are now RFCs.

A.4. Changes From -00 to -01

1. Numerous editorial and clarity improvements.
2. Picked up updated terminology from the framework document (e.g., Tenant System).
3. Significant changes regarding IEEE 802.1 Ethernets and VLANs. All text moved to the Related Work section, where the technology is summarized.
4. Removed section on Forwarding Table Size limitations. This issue only occurs in some deployments with L2 bridging, and is not considered a motivating factor for the NVO3 work.

5. Added paragraph in Introduction that makes clear that NVO3 is focused on providing both L2 and L3 service to end systems, and that IP is assumed as the underlay transport in the data center.
  6. Added new section (2.6) on Optimal Forwarding.
  7. Added a section on Data Plane issues.
  8. Significant improvement to Section describing SPBM.
  9. Added sub-section on VDP in "Related Work"
- A.5. Changes from draft-narten-nvo3-overlay-problem-statement-04.txt
1. This document has only one substantive change relative to draft-narten-nvo3-overlay-problem-statement-04.txt. Two sentences were removed per the discussion that led to WG adoption of this document.

#### Authors' Addresses

Thomas Narten (editor)  
IBM

Email: narten@us.ibm.com

Eric Gray (editor)  
Ericsson

Email: eric.gray@ericsson.com

David Black  
EMC

Email: david.black@emc.com

Luyuan Fang  
Cisco  
111 Wood Avenue South  
Iselin, NJ 08830  
USA

Email: lufang@cisco.com

Lawrence Kreeger  
Cisco

Email: kreeger@cisco.com

Maria Napierala  
AT&T  
200 Laurel Avenue  
Middletown, NJ 07748  
USA

Email: mnapierala@att.com

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: July 3, 2013

Bhumip Khasnabish  
ZTE USA, Inc.  
Bin Liu  
ZTE Corporation  
Baohua Lei  
Feng Wang  
China Telecom  
Dec 30, 2012

Mobility and Interconnection of Virtual Machines and Virtual Network  
Elements  
draft-khasnabish-vmmi-problems-03.txt

Abstract

In this draft, we discuss the challenges and requirements related to the migration, mobility, and interconnection of Virtual Machines (VMs) and Virtual Network Elements (VNEs). VM migration scheme across IP subnets is needed to implement virtual computing resources sharing across multiple network administrative domains. Many technologies are involved in the VM migration across DCs. These technologies are classified and discussed according to their different locations in the inter-DC and intra-DC network. For the seamless online migration in various scenarios, many problems need to be resolved in the control plane. The VM migration process should be adapted to these aspects. We also describe the limitations of various types of virtual local area (VLAN) networking technologies and virtual private networking (VPN) technologies that are traditionally expected to support such migration, mobility, and interconnections.

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 July 3, 2013.



## Copyright Notice

Copyright (c) 2012 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 . . . . .	4
1.1. Conventions Used in this Document . . . . .	4
2. Terminology and Concepts . . . . .	4
3. Control & Mobility Related Problem Specifications . . . . .	6
3.1. Summarization of Mobility in Virtualized Environments . . . . .	7
3.2. VM Migration Mobility Problems across IP Subnets/WAN . . . . .	7
3.2.1. IP Tunnel Problems . . . . .	9
3.2.2. IP Allocation Strategy Problems . . . . .	10
3.2.3. Routing Synchronization Strategy Problems . . . . .	12
3.2.4. The Migration Protocol State Machine of VM Online Migration across Subnets . . . . .	12
3.2.5. Resource Gateway Problems . . . . .	13
3.2.6. Optimized Location of Default Gateway . . . . .	13
3.2.7. Other Problems . . . . .	13
3.3. VM Mobility Problems Implemented on the VR Device . . . . .	13
3.4. Security and Authentication of VMMI . . . . .	13
3.5. The Virtual Network Model . . . . .	14
3.6. The Processing Flow . . . . .	14
3.7. The NVE/OBP Location Problems . . . . .	15
3.7.1. NVE/OBP on the Server . . . . .	16
3.7.2. NVE/OBP on the ToR . . . . .	17
3.7.3. Hybrid Scenario . . . . .	19
4. Technology Problems involved in the VM Migration . . . . .	19
4.1. TRILL . . . . .	20
4.2. SPB . . . . .	20
4.3. Data Center Interconnection Fabric Related Problems . . . . .	20
4.4. Types and Applications of VPN Interconnections between DCs which Provide DCI . . . . .	21
4.4.1. Types of VPNs . . . . .	21
4.4.2. Applications of L2VPN in DCs . . . . .	21

4.4.3. Applications of L3VPN in DCs . . . . .	22
4.5. The Actual Number of Available Isolated Domains . . . . .	22
4.6. Problems of the Number of Management Devices/Management Domains of DC in the NVO3 Network . . . . .	22
4.7. Limitation of TCAM capacity . . . . .	23
4.8. SDN . . . . .	23
4.9. LISP . . . . .	23
5. Others Problems Related with IETF and IEEE . . . . .	23
5.1. Review of VXLAN, NVGRE, and NVO3 . . . . .	23
5.2. The East-West Traffic Problem . . . . .	25
5.3. The MAC, IP, and ARP Explosion Problems . . . . .	26
5.4. Suppressing Flooding within a VLAN . . . . .	27
5.5. Packet Encapsulation Problems . . . . .	27
6. Acknowledgement . . . . .	27
7. References . . . . .	27
8. Security Considerations . . . . .	28
9. IANA Consideration . . . . .	28
10. Normative References . . . . .	28
Authors' Addresses . . . . .	28

## 1. Introduction

There are many challenges related to the VM migration and their interconnections among two or more data centers (DCs). The technologies that can be used for VM migration and DC interconnection should support the required level of performance, security, scalability, along with simplicity and cost-effective management, operations and maintenance.

In this draft, the issues and requirements for moving the virtual machines are summarized with reference to the necessary conditions for migration, business needs, state classification, security, and efficiency.

In this draft, the requirements for VMMI technologies that are useful on large-scale Layer-2 network and on segmented IP network/WAN are discussed. VM migration scheme across IP subnets/WAN is therefore needed to implement virtual computing resources sharing across multiple network administrative domains. This will make a wider range of VM migration possible, and allow for migration of VMs to different types of DCs. It can be adapted to different types of physical networks, different topological networks, and various protocols. For the seamless online migration in these scenarios, a very intelligent seamless VM migration orchestration is needed in the control plane. We summarize the requirements of virtual networks for VM migration, virtual networking, and operations in DCI/overlay modes.

### 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 [RFC2119].

## 2. Terminology and Concepts

- o ARP: Address Resolution Protocol
- o DC: Data Center
- o DC GW: Data Center Gateway
- o DCI: Data Center Interconnection
- o DCS: Data Center Switch

- o FDB: Forwarding DataBase
- o HPC: High-Performance Computing
- o IDC: Internet Data Center
- o IP: Internet Protocol
- o IP VPN: Layer 3 VPN, defined in L3VPN working group
- o LISP: Locator ID Separation Protocol
- o NVO3: Network Virtualization Overlays (Over Layer-3)
- o OBP: Overlay network boundary point
- o OTV: Overlay Transport Virtualization
- o PBB: Provider Backbone Bridge
- o PM: Physical Machine
- o QoS: Quality of Service
- o STP: Spanning Tree Protocol
- o TNI: Tenant Network Identifier
- o ToR: Top of the Rack
- o TRILL: Transparent Interconnection of Lots of Links
- o VLAN: Virtual Local Area Networking
- o VM: Virtual Machine
- o VMMI: Virtual Machine Mobility and Interconnection
- o VN: Virtual Network
- o VNI: Virtual Network Identifier
- o VNE: Virtual Network Entity.(a virtualized layer-3/network entity with associated virtualized port and virtualized processing capabilities)
- o VPN: Virtual Private Network

- o VPLS: Virtual Private LAN Service
- o VR: Virtual Router(a logical device which realizes simulation capabilities of the physical router on the software using hardware layer's resources and capabilities)
- o VRRP: Virtual Router Redundancy Protocol
- o VSE: Virtual Switching Entity (a virtualized layer-2/switch entity with associated virtualized port and virtualized processing capabilities)
- o VSw: Virtual Switch
- o WAN: Wide Area Network
- o communication agent: NVO3 communication agent(an entity which forwards traffic between NVO3 and non-NVO3 environments)

### 3. Control & Mobility Related Problem Specifications

Overall, the requirements of VM migration bring in the following challenges in the forefront of data center operations and management:

(A)How can the existing technologies be compatible with each other, including various multi-tenant network technologies, network interface technologies between VMs and ToRs, Access layer network technologies, networking technologies within the DC and across DCs? So that these technologies can work seamlessly with overlay network technologies through virtualization technologies. It may accomplish the accommodation of a large number of individual tenant networks in the DC and the communication between the isolated domains. It makes the VMs can be seamlessly online migrated in the management domain.

(B)When the VMs are migrated from one DC to another within one administrative domain, (i) how to ensure that the necessary conditions of migration are satisfied, (ii) how to ensure that a successful migration occurs without service disruption, and (iii) how to ensure successful rollback when any unforeseen problem occurs in the migration process.

(C)When the VMs are migrated from one administrative domain to another, how to solve the problems of seamless communications between the domains. There are several different solutions(such as VXLAN, NVGRE,etc) to the current Layer-2 (L2) based DC interconnection technology, and each can solve different problems in different scenarios. If the unification of packet encapsulation mapping rules in different solutions can be achieved, it is bound to promote

seamless migration of VMs among DCs along with the desired integration in cloud computing and networking.

(D) How to utilize IP based technologies to accomplish the migration of VMs over layer-3 (L3) network? For example, VPN technology can be used to carry L2 and L3 traffic across the IP/MPLS core network.

We discuss the above in more details in the following sections. A related draft [DCN Ops Req] discusses data center network and operations requirements.

### 3.1. Summarization of Mobility in Virtualized Environments

Mobility refers to the movement of a VM from one server to another server within one DC or across DCs, while maintaining the VM's original IP and MAC address throughout the process. When a VM is migrated to a new location, it should maintain the existing client sessions, and the state of the VM sessions should be copied to the new location. VM mobility does not change the VLAN/subnet connection to the VM, and it requires that the serving VLAN is extended to the new location of the VM.

In order to support VM mobility, it is required to allow VMs to be migrated easily and repeatedly -- that is as often as needed by the applications and services -- among a large (more than two) number of DCs. Seamless migration of VMs in mixed IPv4 and IPv6 VPN environments should be supported by using appropriate DC GWs.

Some widely used VM migration tools require that management programs on the source server and destination server are directly connected via an L2 network. The objective is to facilitate the implementation of smooth VM migration.

The participating source server and destination server in the VM migration process may be located in different DCs. It may be required to extend the Layer-2 network beyond what is covered by the L2 network of the source DC. This may create islands of the same VLAN in different (geographically dispersed) DCs.

Besides, the optimal forwarding in a VLAN that supports VM mobility may involve traffic management over multiple DCs. The support of seamless mobility of VM across DCs may not necessarily always achieve optimal intra-VLAN forwarding and routing.

### 3.2. VM Migration Mobility Problems across IP Subnets/WAN

There are many existing implementable solutions for migrating VM within a traditional LAN(non-large-scale Layer-2 network). These

solutions include Xen, KVM, and VMWare, which all implement VM image file sharing based on NFS, and only CPU and memory status are migrated. These are layer-2 VM migration technologies. The advantage of the implementation is that VM's IP addresses don't need to be changed after the VM migration. With the development and popularization of the DCs and virtualization technologies, the number of servers and network environment in a single LAN will limit the scalability of the virtual computing environment.

In addition, when re-configuring the VLAN in the traditional DC network, STP (MSTP) will lead to the VLAN isolation. It is a very serious problem in the DC network, especially in the storage network, because the storage network is very demanding for uninterrupted service.

With the evolution of new technologies, such as various virtualization technologies, the large-scale Layer-2 technology, DCI technology and overlay network technology, the network environment in the VM migration will become more complex. In order to realize virtual computing resource sharing and online VM migration across multiple management domains using these technologies as the foundation, VM migration scheme based on Nvo3 network technology is needed to adapt to the complex network environment. This will make a wider range of VM migration possible, and can allow for migration of VMs to different types of DCs. It can be adapted to different types of physical networks, different topological networks, and various protocols.

For example, in the process of VM migration in DC, there are scenarios that VM in the traditional three-tier topological network is migrated through WAN to Fat-Tree topological network, or to a variety of other topological networks. For the seamless online migration in these scenarios, a very intelligent seamless VM online migration is needed to be implemented in the control plane.

If VM migration is only implemented in the L2 domain, people are concerned about the expansion of the number of VLANs or isolated domains, such as the 16,000,000 isolated domains in PBB.

Now the limitless and seamless VM online migration across overlay based IP subnets means that the following issues need to be addressed, in order to achieve our goal: to create a true virtual network environment that is separated from the physical network.

Isolation domain mapping rules.

Migration across IP subnets.

VM migration in the overlay network needs to be adapted to the heterogeneous network topology.

How the source network environment(i.e. the network between the VM or its server and the connected ToR) adapts its configuration to the destination network environment?

The network redirection technology, IP-in-IP technology and dynamic IP tunnel configuration will be used to allow online VM migration across subnets.

A module which allocates IP addresses to VMs is needed, which can manage each IP allocation in the virtual network. The IP allocation should not conflict with each other, and make the path cost of routing forwarding as small as possible. It is necessary to know the DC network topology, its routing protocols, and real-time results of the path cost to realize minimum path cost. We know that the network topologies of different DCs are not necessarily the same. For example, the network topologies and routing protocols of traditional DC and Fat-Tree network DC are different. The addition of related protocol processing in the control plane is needed for seamless VM migration between them. Otherwise, online VM migration cannot be implemented across DCs or across IP subnets. The scheme of IPinIP tunneling resolves the contradiction between unchanged IP addresses during the VM migration and changed IP addresses when VMs are migrated across IP subnets. Therefore, the VM's mobility problem can be solved only after the above mentioned problems have been solved.

Service providers can implement VM migration by upgrading its software to support new protocols, and the hardware devices don't need to be upgraded.

These problems are described as below:

### 3.2.1. IP Tunnel Problems

During the VM migration, it is required to establish the IP-in-IP tunnels . The purpose is to make the user/application have no perception of the migration process, and their IP addresses on the related layer should be the same. The scheme of IPinIP tunneling resolves the contradiction between unchanged VM IP addresses during the VM migration and changed server IP addresses when VMs migrate across IP subnets. OBP is involved in setting up IP tunnels. According to nvo3 control plane protocols, there are two positions for OBP (NVE / VTEP) in the DC: on the Server and on ToR. Placing OBP on the server can minimize its correlation with network elements in the specific network topology. It will face more problems if OBP



is placed on ToR. NVE is preferred to be placed on the Server (unless there are other stronger reasons). It will create a virtual network for VM communications. The traffic between VMs will not directly be exposed on the wire and switches.

However, OBP on the Server can reduce the weak coupling with DC topology to a certain extent, but they cannot be completely unrelated.

The disadvantage of network connection solutions for online VM migration across different subnets is that the network configuration of VM needs to be changed after the migration, and the migration process is opaque. So the transparent migration of VM needs to be implemented, and network connection redirection technology needs to be considered.

Since users cannot utilize the VMs due to changes of the network access point during the online migration of VMs across subnets, the scheme of network connection redirection system based on Proxy Mobile IP (PM IP) can be used. VM which is migrated to the external subnet is regarded as a mobile node and the IP address isn't changed. All the packets to/from the VM is transmitted through the bi-directional tunnel between the external network and the home network, in order to implement online transparent migration across subnets. After the necessary data has been migrated, the tunnel directly connected to the location of the VM's new server is re-established at the preferably switching speed.

The source VM and the destination VM need to be activated simultaneously and must be dynamically configured with IP tunnel. In order to make the VM migration process completely transparent (including transparent to the VMs' applications and the outside users), the migration environment of the VMs should be regarded as a mobile network environment, and the migrated VM is regarded as a mobile node. The mobile agent function of the host should be taken full advantage to communicate with the external network.

### 3.2.2. IP Allocation Strategy Problems

In the encapsulation of packets described as above, the IP address of the VM is a critical entity. Its allocation is based on DHCP in the small network. With the expansion of the network scale, IP address conflict is more likely to occur. When the VM is migrated to another network, its IP address may possibly conflict with IP address of the VM or physical host in the destination network. For example, the duplicate IP addresses will causes confusion and migration failure.

Therefore, a management module allocating IP addresses to VMs is

needed, which can manage each IP allocation in the virtual network. The IP allocation should not conflict with each other, and should make the path cost of routing and forwarding as small as possible.

When allocating IP, it should not conflict with the currently assigned IP network segments of the VM clusters. In addition, it should not conflict with the IP network segments where the physical hosts are located. It also should not conflict with the destination IP network segments after the migration. So the synchronization of IP address allocation information needs to be done. Of course, the synchronization in the whole network is not necessary as long as there are ways to ensure no conflict exists. Moreover, the allocation method needs to consider the introduction of network overhead as small as possible, as well as insufficient IP issues in the destination network segments.

When allocating IP addresses to the hosts based on DHCP protocol, the IP addresses in the IP address pool are allocated from small to large. The insufficient number of addresses in the pool may lead to conflict with assigned VMs' IP addresses, which hinders VM migration. The IP address allocation from small to large makes assigned VMs' IP addresses may affect routing protocol to choose the better path.

Especially for the specific architectures like Fat-Tree, specific network topology and the protocol architecture of specific routing strategy (such as OSPF) should be utilized. The VM migration process must be adapted to these aspects, and cannot be copied for purely Layer-2 migration approach. So VM migration is inherently related to network topologies and network routing protocols.

In the Fat-tree topology, IP addressing and IP allocation methods of network servers and switches are related to the routing protocols. Two routing methods can be chosen: OSPF protocol (OSPF domain cannot be too large), and the fixed routing configuration.

As VM in the destination DC is needed to assign an IP, in order to prevent IP conflict, routing protocols used in the destination DC need to be known. For example, OSPF routing protocol (in this case, the new added network node is assigned IP address by using DHCP), or the fixed configuration IP routing protocol is used in the Fat-tree topology. If the former, the number and distribution of reserved IP addresses in the IP address pool are different from the latter. Therefore a scheme is required to know the adopted network topology and address allocation strategy, IP usage for each segment, the remaining number of IP addresses, etc. This information cannot be acquired purely by the existing DHCP protocol.

Different routing strategies have different routing management

mechanism for VM migration across the DCs for the following reasons: (a) It involves the uniqueness problem of the IP address assignment and IP tunnel establishment, and (b) it involves the global unified management issues. These problems will be discussed later.

In the addressing method of the fixed routing protocol, the IP address assigned to the device located within DC actually contains the location information. The type of the corresponding device can easily be determined through IP, and the location of the device in the topology can also be visually judged.

So these addresses must be avoided in the automated allocation of IP addresses to the VMs.

The function of DHCP protocol needs to be greatly enhanced, or the protocols and tools of IP address allocation need to be re-designed.

Moreover, negotiation is needed before migration. Because it may be required to migrate the VMs back after the migration process has been finished, in order to make the migration process smoothly, the IP addresses of source /destination communications agents should be considered for reservation. The reserved IP address relates to network topology and IP address allocation strategy in the source/destination network environment.

For the control plane protocols in the nvo3 network, reasonable allocation of IP address is used according to the adopted network topology and routing protocols in the source and destination DC, in order to achieve the seamless VM migration and the optimal path as much as possible.

In addition, the above mentioned problems are also involved in PortLand network topology, similar to Fat Tree network topology.

Future server-centric network topologies, such as Dcell/Bcube network topology, also need to achieve compatibility in the control plane.

### 3.2.3. Routing Synchronization Strategy Problems

In order to ensure the normal data forwarding after the VM migration, the routing synchronization between the source network and destination network is needed.

### 3.2.4. The Migration Protocol State Machine of VM Online Migration across Subnets

As for the routing strategy discussed earlier, compared to the migration in the same IP subnet, the IP allocation strategy and

routing synchronization strategy will be changed. So the state and handling of routing updates must be included in the state machine of VM migration across subnets at the preparation phase before the VM migration.

Therefore, if the VM is allowed to span cross subnets, the network redirection technology should be used. For IP-in-IP technology, the advantage of it is good compatibility with network equipments, as long as upgrading their software.

#### 3.2.5. Resource Gateway Problems

A resource gateway is needed to record IP address resources that have been used, and IP network segments which the used IP addresses belong to.

#### 3.2.6. Optimized Location of Default Gateway

The VM's default gateway should be in a close topological proximity to the ToR that is connected to the server presently hosting that VM.

#### 3.2.7. Other Problems

Migration across domains has proposed new requirements for network protocols, for example, the ARP response packet mechanism is no longer applicable in the WAN. In addition, some packets will be lost during the migration, which does not apply to parallel computing. There are also problems such as computing resources sharing across multiple administrative domains, etc.

#### 3.3. VM Mobility Problems Implemented on the VR Device

Each virtual router (VR) has logically independent routing table and forwarding table. It supports the overlay of private IP addresses and public IP addresses. Multiple VRs can be logically formed on one physical router. Each VR individually runs its own instances of the routing protocols, and has its dedicated I/O ports, the cache, the address space, routing and forwarding tables and network management software. But it isn't in a way that supports multi-tenancy. Therefore it does not support the live migration of VM.

#### 3.4. Security and Authentication of VMMI

During the VM migration process, it is required to give proper considerations to the security related matters; this includes solving traffic roundabout issues, ensuring that the firewall functionalities are appropriately enacted, and so on.

Therefore, in addition to authorization and authentication, appropriate policies and measures to check/enforce the security level must be in place while migrating VMs from one DC to another, especially from a private DC to a public DC in the Cloud [NIST 800-145, Cloud/DataCenter SDO Survey].

For example, when a VM is migrated to the destination DC network, the corresponding switch port connected to the VM and its host server should utilize the port strategy of the source switch. The end time of the VM migration and the issue time of the strategy must be synchronized. If the former is earlier than the latter, the services may not get a timely response, and if the former is later than the latter, it may not have exact level of network security for a time period.

What may be helpful in such environment is the creation and maintenance of a reasonable interactive state machine.

### 3.5. The Virtual Network Model

Based on the above problems, two requirements will be added on the virtual network model: Firstly, the routing information is adjusted automatically according to the physical location of VM after the VM is migrated to a new subnet; Secondly, a logical entity, namely "virtual network communications agent", is added, which is responsible for data routing, storage and forwarding in the inter-subnets communications. The agent can be dynamically created and revoked, and can be running on each server.

The communication nodes in the overlay layer are called overlay entities, which are composed by all VMs and communication agents. Each VN on the overlay layer can be customized as required. The VN is composed by the specified VMs and communication agents. VMs and communication agents may come from different physical networks. They are connected through private tunnels established by the communication agents. The positions of communication agents in the physical network can be divided into two categories: on the server or on the network device (such as on the ToR).

### 3.6. The Processing Flow

During the process, VM migration messages will trigger the topology updates of the VMs' clusters in the source virtual network and destination virtual network. It is therefore required to acquire the network topology, the routing protocols, and the IP address assignment rules for each other on both ends, so the VM can be assigned an unique IP address. The routing information of the communications agents is updated. The communications agent captures the corresponding VM's packets, encapsulates them into the data section of the packets, and adds the necessary control information

(such as self-defined forwarding rules). After the encapsulation, these packets are transferred to the destination network through the tunnels between the communications agents. The communications agent in the destination network de-capsulates the packets and processes the information, and then delivers the packets to the destination network. The data transfer process across subnets is now completed.

The modules which need to be modified are as follows:

According to the above processing flow, the modules can be divided by function as follows: Routing management, MAC capture, Tunnel packet encapsulation, Tunnel forwarding, Tunnel packet de-capsulation, and Forwarding in the destination network.

### 3.7. The NVE/OBP Location Problems

VMs communicate with each other through the interconnected network either within the same domain, or between different domains. According to various NVE / OBP position, the processing is different.

As it is transparent to network topology and L2/L3 protocol, NVE / OBP on the server should be the default configuration mode.

Assume that a set of VMs and the network that interconnects them are allowed to communicate with each other, MAC source and destination addresses in the Ethernet header of the packets exchanged among these VMs are preserved. This is L2-based VM communication within a LAN. Any VM should have its own IP. If a VM belongs to more than one domain, this VM will have multiple IP addresses and multiple logical interfaces, which is similar to the model of L3 switches.

Different VM clusters are distinguished by VLAN mechanism in the same L2 physical domain. In the case of VM communications across IP subnets, the packets are encapsulated in NVE, and directly delivered to the peer NVE, and then transferred to the destination VM.

Once migration across L3 network occurs, some scenarios will cause the MAC source address to be modified.

It is also possible that a VM may belong to different VM cluster networks at the same time, and the two VM clusters are distinguished by the VLANs(VN ID).

In the above case, from the perspective of the overlay network, VLAN is mapped to VNI. Different VNI domains are isolated from each other. If you want to communicate between different VNI domains, the packets should be routed according to the outer layer addresses, and then the outer headers are stripped. The packets are looked up

according to the inner layer addresses.

As NVEs may belong to different domains, if a NVE communicates with the other NVE in the same domain, the VLAN-ID of packets exchanged should be the same. In order to simplify the process, the VLAN-IDs are allowed to be removed. But once a NVE communicates with the other NVE in the different domain, the VLAN-ID of packets exchanged may be different.

Note that 'Two VMs' in the following scenarios refers to 'Two VMs which are communicating with each other'.

### 3.7.1. NVE/OBP on the Server

Scenario classification:

Note that 'belonging to the same VN ID' refers to 'within the same subnet'.

(1) Two VMs are on the same server and belong to the same VN ID. In this case, the packets are forwarded with the Layer-2 mechanism.

VM migration processing scenario: this scenario is not in the scope of VM migration.

(2) Two VMs are on the same server, but belong to different VN IDs. In this case, the packets are forwarded with the Layer-3 mechanism.

VM migration processing scenario: this scenario is not in the scope of VM migration.

(3) Two VMs are on different servers, but belong to the same VN ID. In this case, the packets are encapsulated on the local NVE, and forwarded according to the outer layer addresses with the L2 mechanism. After they are delivered to the peer NVE, the outer layer addresses are stripped, and then the packets are forwarded according to the inner layer addresses.

VM migration processing scenario: it can be processed in the way as across IP subnets, and IP-in-IP tunnel is needed.

(4) Two VMs are on different servers, and belong to different VN IDs. In this case, the packets are encapsulated on the local NVE, and forwarded according to the outer layer addresses with the L3 mechanism. After they are delivered to the peer NVE, the outer layer addresses are stripped, and then the packets are forwarded according to the inner layer addresses.

VM migration processing scenario: it is similar with the third scenario.

In the above cases, the processing of the outer layer information of the packets, including L2 information( such as destination MAC, source MAC, VLAN ID) and L3 information, follows the existing mechanism, because the underlying network is transparent to the overlay layer.

In the case of NVE on the server, the routing protocols (unicast and multicast) may not be changed on the underlying network, but the routing design of the overlay layer is subjected to considerable restriction.

### 3.7.2. NVE/OBP on the ToR

In case of NVE on the ToR, NVE needs to handle the VLANs of various packets. Once the VM is migrated, the rules of source network also need to be migrated, causing physical network configuration changes. Therefore, it is required to develop a set of rules to deal with it.

In this case, the source of the VLANs used by the VM please refer to Section 4.5 (The Actual Number of Available Isolated Domains). Various rules and usage range of the VLANs are required to be set. The VLAN-ID used by a given VM refers to the VLAN-ID carried by the traffic that is originated by that VM and within the same L2 physical domain.

When a VM is communicating with the VM in a different VLAN, there are two ways to implement before their communication messages enter into the processing module in the overlay layer. Firstly, the implementation is on Layer-2. One possible solution is that the port on which the server hosting the VM is connected to ToR belongs to two or more VLANs. Secondly, the implementation is on Layer-3. The VM has multiple IP addresses and logical interfaces. The number of addresses and interfaces are related to the number of communication parties. These methods are similar to the approaches of conventional Layer-2 forwarding and Layer-3 routing. After the processing, the packets enter into the processing module in the overlay layer, and then the header information is processed.

Scenario classification:

(1)Two VMs are on the servers connected to the ports on the same ToR and these ports belong to the same VLAN.  
In this case, the packets are forwarded according to the underlying network address with the L2 mechanism.



VM migration processing scenario: VM is migrated in the way as within the same VLAN.

(2) Two VMs are on the servers connected to the ports on the same ToR and these ports belong to different VLANs. In this case, the packets are forwarded according to the underlying network address with the L3 mechanism.

VM migration processing scenario: it can be processed in the way as across IP subnets, and IP-in-IP tunnel is needed.

(3) Two VMs are on the servers connected to the ports on different ToRs and these ports belong to the same VLAN. In this case, the packets are encapsulated on the ToRs, then enter into the processing in the underlying network. If the ToRs are directly L2 connected, the packets are forwarded according to the outer layer addresses with the L2 mechanism; if the ToRs are not directly L2 connected, the packets are routed according to the outer layer addresses with the L3 mechanism. After they are delivered to the peer NVE, the outer layer addresses are stripped, and then the packets are forwarded according to the inner layer addresses.

VM migration processing scenario: it is similar with the second scenario.

(4) Two VMs are on the servers connected to the ports on different ToRs and these ports belong to different VLANs. In this case, the packets are encapsulated on the ToRs, and processed in the underlying network. Since the ToRs are not directly L2 connected, the packets are routed according to the outer layer addresses with the L3 mechanism. After they are delivered to the peer NVE, the outer layer addresses are stripped, and then the packets are forwarded according to the inner layer addresses.

VM migration processing scenario: it is similar with the second scenario.

When a VM is communicating with the VM in a different VLAN, there are two implementations before their communication messages enter into the processing module in the overlay layer. Firstly, the implementation is on Layer-2. One possible solution is that the port belongs to two or more VLANs, on which the server hosting the VM is connected to ToR. Secondly, the implementation is on Layer-3. The VM has multiple IP addresses and logical interfaces. The number of addresses and interfaces are related to the number of communication parties. These methods are similar to the approaches of conventional Layer-2 forwarding and Layer-3 routing. After the processing, the

packets enter into the processing module in the overlay layer, and then the header information is processed.

### 3.7.3. Hybrid Scenario

The hybrid scenario should be considered in the communication model of NVO3 network. This may include the situation when NVE is on the Server in some part of the network, and NVE is on ToR in other part of the network. The normal communication between two parts of the network should be covered under the hybrid scenario discussion.

## 4. Technology Problems involved in the VM Migration

As mentioned above, when VMs are migrated from one server to another, the source and destination server may be within the same DC, or in the DCs at different geographic locations. The remote data centers can be interconnected through different DCI technologies. Overall, the technologies involved in the VM migration across DCs include those within the DC. The key technologies involved in the VM migration are broadly categorized according to different locations of the elements in DCs as follows:

- 1)a) VM/Network-aware VM migration technologies, which makes the network automatically perceive the network behavior of VM/Host (such as QoS / VLAN, etc.). The network behavior is mainly perceived by the switch which the server hosting VM is directly connected to. Such technologies include IEEE EVB / VDP, PortExtender (802.1Qbh), and so on.
- b) Service-aware VM migration technologies, e.g., a distributed cluster of VMs may be used for Unified Communications Services (UCS), and when any UCS related VMs need to be migrated, a set of VMs from the clusters of the same category can be used.
- 2) The network interconnection driven technologies within the DC, including IETF TRILL, IEEE 802.1Q/SPBV, and PBB / SPBM.
- 3) The internal exit device interconnection based technologies across the DCs, including IETF VPLS, PBB-EVPN.
- 4) The ToR / Host interconnection technologies within the DC as well as across the DCs, including the NVO3 which IETF is currently standardizing.

The above-mentioned classification is according to the locations of various elements in DC. The technologies in the second class or third class can logically become one of the foundations of the

technologies in the fourth class.

The above-mentioned VM migration technologies may be considered to be within the scope of NVO3 WG.

#### 4.1. TRILL

The large-scale Layer-2 and multi-path interconnection in the DC network can be implemented through TRILL. The transparent TRILL network can be used to ensure that the MAC and IP remain unchanged in the VM migration. However, in order to guarantee service continuity during the VM migration, the control plane of NOV3 exchanges messages with the control plane of TRILL, so that the position state can be quickly updated after the VM migration. Business traffic can be quickly pointed to the new network address of the VM after the migration and tenants may not have any perception of the VM migration.

#### 4.2. SPB

As with TRILL, the large-scale Layer-2 and multi-path interconnection in the DC network can be implemented through SPB. The transparent SPB network can be used to ensure that the MAC and IP remain unchanged in the VM migration. However, in order to guarantee service continuity during the VM migration, the control plane of NOV3 exchanges messages with the control plane of SPB, so that the position state can be quickly updated after the VM migration. Business traffic can be quickly pointed to the new network address of the VM after the migration and tenants may not have any perception of the VM migration.

#### 4.3. Data Center Interconnection Fabric Related Problems

One of the most important factors that directly impact the VMMI is connectivity among the relevant data centers. There are many features that determine this required connectivity. These features of connectivity include bandwidth, security, quality of service, load balancing capability, etc. These are frequently utilized to make decision on whether a VM can join a host in real-time or it needs to join VRF in certain unit of VM.

The requirements related to the above are as follows:

- o The negative impact of ARP, MAC and IP entry explosion on the individual network which contains a large number of tenants should be minimized by DC and DC-interconnection technologies.
- o The link capacity of both intra-DC and inter-DC network should be effectively utilized. Efficient utilization of the link capacity

requires traffic forwarding on the shortest path between two VMs both within the DC and across DCs. Therefore, Traffic should be forwarded on the shortest path between two VMs within the DC or across DCs.

- o Support of east-west traffic between tenants' applications located in different DCs.

Many mature VPN technologies can be utilized to provide connectivity between DCs. The extension of VLAN and virtual domain between DCs may also be utilized for this purpose.

#### 4.4. Types and Applications of VPN Interconnections between DCs which Provide DCI

##### 4.4.1. Types of VPNs

Related technologies of layer3 VPN: BGP / MPLS IP Virtual Private Networks (VPNs), RFC 4364, etc.

Related technologies of layer2 VPN: PBB + L2VPN, TRILL + L2VPN, VLAN + L2VPN, NVGRE [draft-sridharan-virtualization-nvgre-00], PBB VPLS, E-VPN, PBB-EVPN, VPLS, VPWS, etc.

##### 4.4.2. Applications of L2VPN in DCs

It is a very common practice to use L2 interconnection technologies for DC interconnection across geographical regions. Note that VPN technology is also used to carry L2 and L3 traffic across the IP/MPLS core network. This technology can be used in the same DC to support scalability or interconnection across L3 domains. VPLS is commonly used for IP/MPLS connection over WAN and it supports transparent LAN services. IP VPN, including BGP / MPLS IP VPN and IPsec VPN, has been used in a common IP/MPLS core network to provide virtual IP routing instances.

The implementation of PBB plus L2-VPN can take advantage of some of the existing technologies. It is flexible to use VPN network in the cloud computing environment and can support a sufficient number of VPN connections/sessions (networking resources), which is much larger than the 4K VLAN mode of L2VPN. Therefore, the resulting effect is similar to that of VXLAN.

Note that PBB can not only support access to more than 16M virtual LAN instances, it can also separate the tenants and provide different domains through isolated MAC address spaces.

The use of PBB encapsulation has one major advantage. Note that since VM's MAC address will not be processed by ToRs and Core SWs, MAC table size of ToRs and Core SWs may be reduced by two orders of magnitude; the specific number is related with the number of VMs in each server and VMs' virtual interfaces.

One solution to solve problems in DC is to deploy other technologies in the existing DC network. A service provider can separate its domains of VLAN into different VLAN islands, in this way each island can support up to 4K VLANs. Domains of VLAN can be interconnected via VPLS, at the same time, DC GWs can be used as VPLS PEs. If retaining the existing VLAN-based solutions only in VSw, while the number of tenants in some VLAN islands is more than 4K, the service provider needs to deploy VPLS deeper in the DC network. This is equivalent to supporting L2VPN from the ToRs, and using the existing VPLS solutions to enable MPLS for the ToR and core DC elements.

#### 4.4.3. Applications of L3VPN in DCs

IP VPN technology can also be used for DC network virtualization. For example, multi-tenant L3 virtualization can be achieved by assigning a different IP VPN instance to each tenant who needs L3 virtualization in a DC network.

There are many advantages of using IP VPN as an L3 virtualization solution within DC compared to using existing virtual routing technology. Some of the advantages are as mentioned below:

- (1) It supports many VRF-to-VRF tunneling options containing different operational models: BGP/MPLS IP VPN, IP or L3 VPN GRE, etc.
- (2) The connections of IP VPN instances used in Cloud services below the WAN can be IP VPN that is directly involved in the WAN.

#### 4.5. The Actual Number of Available Isolated Domains

The isolation of VMs is achieved through VNI. One way to acquire the value of the VNI is through tag mapping in the underlying network. VNI may be derived from the 12-bit VLAN ID, or from other related 24-bit information (e.g. 24-bit PBB I-SID tag mapping, or other information which can be mapped to VNI), It relates to the technical solutions adopted by the Provider Edge Bridge and the degree of chip's support. If VNI is from 12-bit VLAN ID, the actual number of available VN IDs is 4096 in the application; if it is from 24-bit VLAN ID, the actual number of available VN IDs is 16M in the application. This is the compatibility issue between the isolated domains in the overlay layer and the isolated domains in the underlying network.

#### 4.6. Problems of the Number of Management Devices/Management Domains of DC in the NVO3 Network

The settings of management domains should be done in concert with the control plane of NVO3 network. It relates to the management domain of VM. The range of VM's management domain can be considered from two aspects.

Firstly, it relates to whether only one management center (e.g.

vCenter) is required. With VCenter as an example, a single vCenter under the 32-bit OS can control 200 ESX Hosts and 2,000 VMs. If the number of the ESX Hosts and VMs in the DC exceeds the quantity, there should be multiple vCenters in place. Secondly, it relates to the institutional settings. If the company needs operations across the country or across the WAN, a single vCenter can manage through the connections across the WAN, However, multiple vCenters may be needed for hierarchical settings of global organizational structure or the required security settings.

#### 4.7. Limitation of TCAM capacity

Regardless of the locations of communication agents in the overlay network, the number of 16M isolated domains is clearly beyond the maximum TCAM capacity which hardware currently can support, so it is necessary to consider the hierarchical virtual network.

#### 4.8. SDN

If SDN technology is used for VM migration in Nvo3 network, it also has to face the problem of limited TCAM capacity which hardware can support, and the problem is more prominent.

#### 4.9. LISP

If LISP is used an option in Nvo3 control plane, it will face the problem of notify the ITR of fast switching to new Locator IP after the migration. For example, a VM has just migrated from site A to site B. The ITR do not know, and still use the old Locator IP to send the packets. The applications are certainly interrupted and won't be restored until the ITR acquires the new Locator IP. However, the length of the interruption time has a great influence on the time-sensitive operations.

### 5. Others Problems Related with IETF and IEEE

#### 5.1. Review of VXLAN, NVGRE, and NVO3

In order to solve the problem of insufficient number of VLANs in DC, the technologies like VXLAN and NVGRE have adopted two major strategies; one of the strategies is the encapsulation and the other is tunneling.

Both VXLAN and NVGRE use encapsulation and tunneling to create a number of VLAN subnets, which can be extended to the Layer-2 and Layer-3 networks. This solves the problem of limitation of the number of VLAN as defined by IEEE802.1Q, and helps achieve shared

load-balancing in multi-tenant environment in both public and private networks.

The VXLAN technology is introduced in 2011, and it is designed to address the number restrictions of 802.1Q VLAN. The technologies like MAC in MAC, MAC in GRE also extend the number of VLANs. However, VXLAN attempts to address the issues related to inadequate utilization of link resources, monitoring of packets after re-encapsulation of header more effectively.

The frame format of VXLAN is the same as that of OTV and LISP, although these three solutions solve different problems of DC Interconnection and VM migration. Also, in VXLAN, the packet is encapsulated in MAC in UDP, and addressing is extended to 24-bit, which is the effective solution to the restrictions of VLAN number. UDP encapsulation enables the logical virtual network extension to different subnets. It also supports the migration of VMs across subnets. The change of the frame's structure increases the field for extending the VLAN.

Note that VXLAN solves different problem compared to OTV. OTV solves the problem of DC interconnection, which builds an IP tunnel between different data centers through MAC in IP. VXLAN mainly solves the problem of limitation of VLAN resources in DCs due to the increase in the number of tenants. The key is the expansion of the VNI field to increase the number of VLANs. Both technologies can be applied to VM migration, since the two packet formats are almost the same and completely compatible.

NVGRE specifies the 24-bit Tenant Network Identifier (TNI) and resolves some issues related to supporting multiple tenants in DC network. It uses GRE to create an independent virtual Layer-2 network, and limits physical Layer-2 network to expand across subnet borders. Terminals supporting NVGRE insert the TNI indicators in the GRE headers to separate the TNIs.

NVGRE and VXLAN solve the same problem. The two technologies were proposed almost at the same time. However, there are some differences between them:

VXLAN not only increases VXLAN header(VNI), but also increases the outer UDP encapsulation on the packet, which facilitates live migration of VMs across subnets. In addition, differentiated services can be supported to the tenants in the same subnet because of the use of UDP. Both proposals are built on the assumption that load-balancing is the necessary condition to achieve efficient operation. VXLAN randomly assigns port number to achieve load-balancing, while NVGRE uses the retained 8-bit in the key GRE field. However, there may be opportunity to improve the capability of the

control plane for both mechanisms in future.

## 5.2. The East-West Traffic Problem

Let us discuss the background of East-West traffic problem first. There are a variety of applications in the DC, such as distributed computing, distributed storage, and distributed search. These applications and services need frequent exchanges of transactions between the business servers across the DCs. According to the traditional three-tier network model, the data stream first flows north-south and then finally flows east-west. In order to improve the forwarding efficiency of the data stream, it is necessary to update the existing network model and network forwarding technology. Among others, the Layer-2 multi-path technology being studied is one of the directions to solve this problem.

Distributed computing is the basis of transformation of the existing IT services. This allows scalable and efficient use of sometimes underutilized computing and storage resources scattered across the data centers.

In typical data centers, the average server utilization is often low in the existing network. The concept of virtualization and distributed computing can perfectly solve the problem of capacity limitation of a single server in demanding environments in certain DCs via on-demand utilization of resources and without impacting the performance. This revolutionary technology of distributed computing and services using resources in the DCs also produces several horizontal flows of traffic. The application of distributed computing technology on the servers produces a large number of interactive traffic streams between servers. In addition, the types of DC would influence the traffic model both within and across data centers.

The first type of DC is telecom operators who usually not only operate DCs, but also supply bandwidth for the Level-2 ISP providers. The second type is the traditional ISP companies with strong power. The third type is some IT enterprises which invest in the construction of DCs. The fourth type is high-performance computing (HPC) centers that are built by universities, research institutes and organizations. Note that in these types of DCs, the south-north traffic flow is significantly smaller compared to the horizontal flow, and this brings greatest challenges to the network design and installation. In addition to the normal flow of traffic due to the distributed computing, storage, communications, and management, hot backup and live VM migration produce a sudden lateral flow of traffic and associated challenges.



There are two potential solutions to the distributed horizontal flow of traffic, as described below.

A. The first one is to solve the problem of east-west traffic within the server clusters by exploiting representative technologies such as vswitch, Dcell, B-cube, and DCTCP .

B. The second solution is the network-based solution. The tree structure of the traditional DC network is not inherently efficient for horizontal flow of traffic. The problems can be solved in two ways: (i) in the direction of radical changes: radical deformations in changing the tree structure to multi-path, and (ii) in the direction of mild improvement: changing L2 big trees to L2 small trees and meeting the requirements by expanding the interconnection capacity of the upper node, clustering/stacking system, and link trunking.

The requirements related to the above are as follows: Stacking technology across the data center requires specialized interfaces, and the length of feasible transmission distance is limited.

The problems related to the above statement include the following:

(a) although TRILL resolves the multi-path problem of Layer-2 protocol, it negatively impacts the multi-path properties of Layer-3 protocol. This is because only one active default router supports Virtual Router Redundancy Protocol (VRRP), and this means that the multi-path characteristics cannot be fully utilized in Layer-3 protocol. (b) TRILL does not define how to deal with the problem of overlapping namespace.

### 5.3. The MAC, IP, and ARP Explosion Problems

Network devices within data centers encounter many problems for supporting conventional communication framework because they need to accommodate a huge number of IP, MAC addresses and ARP.

Each blade server in a network device usually supports at least 16-40 VMs, and each VM has its own MAC address and IP address. The entities like Disk, memory, FDB table, MAC table, etc, cause an increase in convergence time. In order to accommodate this large number of the servers, different options for the network topology, for example, fat tree topology or a conventional network topology may be considered.

The number of ARP packets grows not only with the number of virtual L2 domains or ELANs, which is instantiated on server, but also with the number of VMs in that domain. Therefore, scenarios like overload of ARP entries on the servers/hypervisors, exhaustion of ARP entries on the routers/PEs, and processing overload of L3 service appliances, must be efficiently resolved. Otherwise, these problems will easily

propagate throughout the layer-2 switching network.

Consequently, what are needed to resolve these problems include (a) automated management of MAC/IP/ARP in DCs, and (b) network deployment that will reduce the explosion in MAC number requirements in DCs.

#### 5.4. Suppressing Flooding within a VLAN

Efficient operations of DCs in nvo3 network require that flooding of broadcast, multicast and unknown unicast frames within a VLAN (that may be caused by the improper configuration) should be reduced.

#### 5.5. Packet Encapsulation Problems

In order to achieve seamless migration of VMs across DCs that support different VLAN expansion mechanisms, unification of packet encapsulation methods is required.

### 6. Acknowledgement

The following experts have provided valuable comments on the earlier version of this draft: Thomas Narten, Christopher LILJENSTOLPE, Steven Blake, Ashish Dalela, Melinda Shore, David Black, Joel M. Halpern, Vishwas Manral, Lizhong Jin, Juergen Schoenwaelder, Donald Eastlake, and Truman Boyes. We express our sincere thanks to them, and expect that they will continue to provide suggestions in future.

### 7. References

[PBB-VPLS] Balus, F. et al. "Extensions to VPLS PE model for Provider Backbone Bridging", draft-ietf-l2vpn-pbb-vpls-pe-model-04.txt (work in progress), October 2011.

[DCN Ops Req] A. Dalela. "Datacenter Network and Operations Requirements", draft-dalela-dc-requirements-00.txt, December 30, 2011

[VPN Applicability] Nabil Bitar. "Cloud Networking: Framework and VPN Applicability", draft-bitar-datacenter-vpn-applicability-01.txt, October 2011

[VXLAN] M.Mahalingam. "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-01.txt, February 24, 2012

[NIST 800-145] NIST Special Publication 800-145, Peter Mell and Timothy Grance, The NIST definition of cloud computing, <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>, September 2011

[Cloud/DataCenter SDO Survey] B. Khasnabish and C. JunSheng. "Cloud/DataCenter SDO Activities Survey and Analysis", draft-khasnabish-cloud-sdo-survey-02.txt, December 28, 2011

[NVGRE] M. Sridharan. "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-00.txt, September 2011

[NVO3] Thomas Narten. " NVO3: Network Virtualization", 12vpn-9.pdf, November 2011

## 8. Security Considerations

To be added later, on as-needed basis.

## 9. IANA Consideration

The extensions that are discussed in this draft are related to DC operations environment.

## 10. Normative References

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

## Authors' Addresses

Bhumip Khasnabish  
ZTE USA, Inc.  
55 Madison Avenue, Suite 160 Morristown, NJ 07960  
USA

Phone: +001-781-752-8003  
Email: [vumi1@gmail.com](mailto:vumi1@gmail.com), [bhumip.khasnabish@zteusa.com](mailto:bhumip.khasnabish@zteusa.com)

Bin Liu  
ZTE Corporation  
15F, ZTE Plaza, No.19 East Huayuan Road, Haidian District  
Beijing 100191  
P.R.China

Phone: +86-10-59932098  
Email: richard.bohan.liu@gmail.com, liu.bin21@zte.com.cn

Baohua Lei  
China Telecom  
118, St. Xizhimennei, Office 709, Xicheng District  
Beijing  
P.R.China

Phone: +86-10-58552124  
Email: leibh@ctbri.com.cn

Feng Wang  
China Telecom  
118, St. Xizhimennei, Office 709, Xicheng District  
Beijing  
P.R.China

Phone: +86-10-58552866  
Email: wangfeng@ctbri.com.cn



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: October 31, 2013

K. Kompella  
Y. Rekhter  
Juniper Networks  
T. Morin  
France Telecom - Orange Labs  
D. Black  
EMC Corporation  
April 29, 2013

Signaling Virtual Machine Activity to the Network Virtualization Edge  
draft-kompella-nvo3-server2nve-02

Abstract

This document proposes a simplified approach for provisioning network parameters related to Virtual Machine creation, migration and termination on servers. The idea is to provision the server, then have the server signal the requisite parameters to the relevant network device(s). Such an approach reduces the workload on the provisioning system and simplifies the data model that the provisioning system needs to maintain. It is also more resilient to topology changes in server-network connectivity, for example, reconnecting a server to a different network port or switch.

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 October 31, 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.

## Table of Contents

1. Introduction . . . . .	2
1.1. VM Creation . . . . .	3
1.2. VM Live Migration . . . . .	4
1.3. VM Termination . . . . .	5
2. Acronyms Used . . . . .	6
3. Virtual Networks . . . . .	7
3.1. Current Mode of Operation . . . . .	8
3.2. Future Mode of Operation . . . . .	8
4. Provisioning DCVPNs . . . . .	9
5. Signaling . . . . .	9
5.1. Preliminaries . . . . .	9
5.2. VM Operations . . . . .	10
5.2.1. Network Parameters . . . . .	10
5.2.2. Creating a VM . . . . .	12
5.2.3. Terminating a VM . . . . .	14
5.2.4. Migrating a VM . . . . .	15
5.3. Signaling Protocols . . . . .	16
6. Interfacing with DCVPN Control Planes . . . . .	16
7. Security Considerations . . . . .	16
8. IANA Considerations . . . . .	17
9. Acknowledgments . . . . .	17
10. Informative References . . . . .	17
Authors' Addresses . . . . .	18

## 1. Introduction

To create a Virtual Machine (VM) on a server in a data center, one must specify parameters for the compute, storage, network and appliance aspects of the VM. At a minimum, this requires provisioning the server that will host the VM, and the Network Virtualization Edge (NVE) that will implement the virtual network for the VM in addition to the VM's storage. Similar considerations apply to live migration and terminating VMs. This document proposes mechanisms whereby a server can be provisioned with all of the parameters for the VM, and the server in turn signals the networking aspects to the NVE. The NVE may be located on the server or in an

external network switch that may be directly connected to the server or accessed via an L2 (Ethernet) LAN or VLAN. The following sections capture the abstract sequence of steps for VM creation, live migration and deletion.

While much of the material in this draft may apply to virtual entities other than virtual machines that exist on physical entities other than servers, this draft is written in terms of virtual machines and servers for clarity.

### 1.1. VM Creation

This section describes an abstract sequence of steps involved in creating a VM and making it operational (the latter is also known as "powering on" the VM). The following steps are intended as an illustrative example, not as prescriptive text; the goal is to capture sufficient detail to set a context for the signaling described in Section 5.

Creating a VM requires:

1. gathering the compute, network, storage, and appliance parameters required for the VM;
2. deciding which server, network, storage and network appliance devices best match the VM requirements in the current state of the data center;
3. provisioning the server with the VM parameters;
4. provisioning the network element(s) to which the server is connected with the network-related parameters of the VM;
5. informing the network element(s) to which the server is connected about the VM's peer VMs, storage devices and other network appliances with which the VM needs to communicate;
6. informing the network element(s) to which a VM's peer VMs are connected about the new VM and its addresses;
7. provisioning storage with the storage-related parameters; and
8. provisioning necessary network appliances (firewalls, load balancers and "middle boxes").

Steps 1 and 2 are primarily information gathering. For Steps 3 to 8, the provisioning system talks actively to servers, network switches, storage and appliances, and must know the details of the physical



server, network, storage and appliance connectivity topologies. Step 4 is typically done using just provisioning, whereas Steps 5 and 6 may be a combination of provisioning and other techniques that may defer discovery of the relevant information. Steps 4 to 6 accomplish the task of provisioning the network for a VM, the result of which is a Data Center Virtual Private Network (DCVPN) overlaid on the physical network.

While shown as a numbered sequence above, some of these steps may be concurrent (e.g., server, storage and network provisioning for the new VM may be done concurrently), and the two "informing" steps for the network (5 and 6) may be partially or fully lazily evaluated based on network traffic that the VM sends or receives after it becomes operational.

This document focuses on the case where the network elements in Step 4 are not co-resident with the server, and describes how the provisioning in Step 4 can be replaced by signaling between server and network, using information from Step 3.

## 1.2. VM Live Migration

This subsection describes an abstract sequence of steps involved in live migration of a VM. Live migration is sometimes referred to as "hot" migration, in that from an external viewpoint, the VM appears to continue to run while being migrated to another server (e.g., TCP connections generally survive this class of migration). In contrast, suspend/resume (or "cold") migration consists of suspending VM execution on one server and resuming it on another. The following live migration steps are intended as an illustrative example, not as prescriptive text; the goal is to capture sufficient detail to provide context for the signaling described in Section 5.

For simplicity, this set of abstract steps assumes shared storage, so that the VM's storage is accessible to the source and destination servers. Live migration of a VM requires:

1. deciding which server should be the destination of the migration based on the VM's requirements, data center state and reason for the migration;
2. provisioning the destination server with the VM parameters and creating a VM to receive the live migration;
3. provisioning the network element(s) to which the destination server is connected with the network-related parameters of the VM;

4. transferring the VM's memory image between the source and destination servers;
5. actually moving the VM: pausing the VM's execution on the source server, transferring the VM's execution state and any remaining memory state to the destination server and continuing the VM's execution on the destination server;
6. informing the network element(s) to which the destination server is connected about the VM's peer VMs, storage devices and other network appliances with which the VM needs to communicate;
7. informing the network element(s) to which a VM's peer VMs are connected about the VM's new location;
8. activating the VM's network parameters at the destination server;
9. deprovisioning the VM from the network element(s) to which the source server is connected; and
10. deleting the VM from the source server.

Step 1 is primarily information gathering. For Steps 2, 3, 9 and 10, the provisioning system talks actively to servers, network switches and appliances, and must know the details of the physical server, network and appliance connectivity topologies. Steps 4 and 5 are usually handled directly by the servers involved. Steps 6 to 9 may be handled by the servers (e.g., one or more "gratuitous" ARPs or RARPs from the destination server may accomplish all four steps) or other techniques. For steps 6 and 7, the other techniques may involve discovery of the relevant information after the VM has been migrated.

While shown as a numbered sequence above, some of these steps may be concurrent (e.g., moving the VM and associated network changes), and the two "informing" steps (6 and 7) may be partially or fully lazily evaluated based on network traffic that the VM sends and/or receives after it is migrated to the destination server.

This document focuses on the case where the network elements are not co-resident with the server, and shows how the provisioning in Step 3 and the deprovisioning in Step 9 can be replaced by signaling between server and network, using information from Step 3.

### 1.3. VM Termination

This subsection describes an abstract sequence of steps involved in termination of a VM, also referred to as "powering off" a VM. The following termination steps are intended as an illustrative example, not as prescriptive text; the goal is to capture sufficient detail to set a context for the signaling described in Section 5.

Termination of a VM requires:

1. ensuring that the VM is no longer executing;
2. deprovisioning the VM from the network element(s) to which the server is connected; and
3. deleting the VM from the server (the VM's image may remain on storage for reuse).

Steps 1 and 3 are handled by the server, based on instructions from the provisioning system. For Step 2, the provisioning system talks actively to servers, network switches, storage and appliances, and must know the details of the physical server, network, storage and appliance connectivity topologies.

While shown as a numbered sequence above, some of these steps may be concurrent (e.g., network deprovisioning and VM deletion).

This document focuses on the case where the network elements in Step 2 are not co-resident with the server, and shows how the deprovisioning in Step 3 can be replaced by signaling between server and network.

## 2. Acronyms Used

The following acronyms are used:

DCVPN: Data Center Virtual Private Network -- a virtual connectivity topology overlaid on physical devices to provide virtual devices with the connectivity they need and isolation from other DCVPNs. This corresponds to the concept of a Virtual Network Instance (VNI) in [I-D.ietf-nvo3-framework].

NVE: Network Virtualization Edge -- the entities that realize private communication among VMs in a DCVPN

l-NVE: local NVE: wrt a VM, NVE elements to which it is directly connected

r-NVE: remote NVE: wrt a VM, NVE elements to which the VM's peer VMs are connected

NVGRE: Network Virtualization using Generic Routing Encapsulation

VDP: VSI Discovery and Configuration Protocol

VID: 12-bit VLAN tag or identifier used locally between a server and its l-NVE

VLAN: Virtual Local Area Network

VM: Virtual Machine (same as Virtual Station)

Peer VM: wrt a VM, other VMs in the VM's DCVPN

VNID: DCVPN Identifier

VSI: Virtual Station Interface

VXLAN: Virtual eXtensible Local Area Network

### 3. Virtual Networks

The goal of provisioning a network for VMs is to create an "isolation domain" wherein a group of VMs can talk freely to each other, but communication to and from VMs outside that group is restricted (either prohibited, or mediated via a router, firewall or other network gateway). Such an isolation domain, sometimes called a Closed User Group, here will be called a Data Center Virtual Private Network (DCVPN). The network elements on the outer border or edge of the overlay portion of a Virtual Network are called Network Virtualization Edges (NVEs).

A DCVPN is assigned a global "name" that identifies it in the management plane; this name is unique in the scope of the data center, but may be unique across several cooperating data centers. A DCVPN is also assigned an identifier unique in the scope of the data center, the Virtual Network Group ID (VNID). The VNID is a control plane entity. A data plane tag is also needed to distinguish different DCVPNs' traffic; more on this later.

For a given VM, the NVE can be classified into two parts: the network elements to which the VM's server is directly connected (the local NVE or l-NVE), and those to which peer VMs are connected (the remote NVE or r-NVE). In some cases, the l-NVE is co-resident with the server hosting the VM; in other cases, the l-NVE is separate (distributed l-NVE). The latter case is the one of primary interest in this document.

A created VM is added to a DCVPN through Steps 4 to 6 in section Section 1.1 which can be recast as follows. In Step 4, the l-NVE(s) are informed about the VM's VNID, network addresses and policies, and the l-NVE and server agree on how to distinguish traffic for different DCVPNs from and to the server. In Step 5 the relevant r-NVE elements and the addresses of their VMs are discovered, and in Step 6, the r-NVE(s) are informed of the presence of the new VM and obtain or discover its addresses; for both steps 5 and 6, the discovery may be lazily evaluated so that it occurs after the VM begins sending and receiving DCVPN traffic.

Once a DCVPN is created, the next steps for network provisioning are to create and apply policies such as for QoS or access control. These occur in three flavors: policies for all VMs in the group, policies for individual VMs, and policies for communication across DCVPN boundaries.

### 3.1. Current Mode of Operation

DCVPNs are often realized as Ethernet VLAN segments. A VLAN segment satisfies the communication properties of a DCVPN. A VLAN also has data plane mechanisms for discovering network elements (Layer 2 switches, aka bridges) and VM addresses. When a DCVPN is realized as a VLAN, Step 4 in section Section 1.1 requires provisioning both the server and l-NVE with the VLAN tag that identifies the DCVPN. Step 6 requires provisioning all involved network elements with the same VLAN tag. Address learning is done by flooding, and the announcement of a new VM or the new location of a migrated VM is often via a "gratuitous" ARP or RARP.

While VLANs are familiar and well-understood, they have scaling challenges because they are Layer 2 infrastructure. The number of independent VLANs in a Layer 2 domain is limited by the 12-bit size of the VLAN tag. In addition, data plane techniques (flooding and broadcast) are another source of scaling concerns as the overall size of the network grows.

### 3.2. Future Mode of Operation

There are multiple scalable realizations of DCVPNs that address the isolation requirements of DCVPNs as well as the need for a scalable substrate for DCVPNs and the need for scalable mechanisms for NVE and VM address discovery. While describing these approaches beyond the scope of this document, a secondary goal of this document is to show how the signaling that replaces Step 4 in section Section 1.1 can seamlessly interact with realizations of DCVPNs.

VLAN tags (VIDs) will be used as the data plane tag to distinguish traffic for different DCVPNs' between a server and its l-NVE. Note that, as used here, VIDs only have local significance between server and NVE, and should not be confused with data-center-wide usage of VLANs. If VLAN tags are used for traffic between NVEs, that tag usage depends on the encapsulation mechanism among the NVEs and is orthogonal to VLAN tag usage between servers and l-NVEs.

#### 4. Provisioning DCVPNs

For VM creation as described in section Section 1.1, Step 3 provisions the server; Steps 4 and 5 provision the l-NVE elements; Step 6 provisions the r-NVE elements.

In some cases, the l-NVE is located within the server (e.g., a software-implemented switch within a hypervisor); in this case, Steps 3 and 4 are "single-touch" in that the provisioning system need only talk to the server, as both compute and network parameters are applied by the server. However, in other cases, the l-NVE is separate from the server, requiring that the provisioning system talk to both the server and l-NVE. This scenario, which we call "distributed local NVE", is the one considered in this document. This draft's goal is to describe how "single-touch" provisioning can be achieved in the distributed l-NVE case.

The overall approach is to provision the server, and have the server signal the requisite parameters to the l-NVE. This approach reduces the workload on the provisioning system, allowing it to scale both in the number of elements it can manage, as well as the rate at which it can process changes. It also simplifies the data model of the network that is used by the provisioning system, because a complete, up-to-date map of server to network connectivity is not required. This approach is also more resilient to server-network connectivity/topology changes that have not yet been transmitted to the provisioning system. For example, if a server is reconnected to a different port or a different l-NVE to recover from a malfunctioning port, the server can contact the new l-NVE over the new port without the provisioning system needing to immediately be aware of the change.

While this draft focuses on provisioning networking parameters via signaling, extensions may address the provisioning of storage and network appliance parameters in a similar fashion.

#### 5. Signaling

##### 5.1. Preliminaries

This draft considers three common VM operations in a virtualized data center: creating a VM; migrating a VM from one physical server to another; and terminating a VM. Creating a VM requires "associating" it with its DCVPN and "activating" that association; decommissioning a VM requires "dissociating" the VM from its DCVPN. Moving a VM consists of associating it with its DCVPN in its new location, activating that association, and dissociating the VM from its old location.

## 5.2. VM Operations

### 5.2.1. Network Parameters

For each VM association or dissociation operation, a subset of the following information is needed from server to l-NVE:

operation: one of associate or dissociate.

authentication: proof that this operation was authorized by the provisioning system

VNID: identifier of DCVPN to which VM belongs

VID: tag to use between server and l-NVE to distinguish DCVPN traffic; the value zero in an associate operation is a request that the l-NVE to assign an unused VID. This approach provides extensibility by allowing the VID to be a VLAN-id, although other local means of multiplexing traffic between the server and the NVE could be used instead of VIDs.

encapsulation type: type of encapsulation used by the DCVPN for traffic exchanged between NVEs (see below).

network addresses: network addresses for VM on the server (e.g., MACs)

policy: VM-specific and/or network-address-specific network policies, such as access control lists and/or QoS policies

hold time: time (in milliseconds) to keep a VM's addresses after it migrates away from this l-NVE. This is usually set to zero when a VM is terminated.

per-address-VID-allocation: boolean flag which can optionally be set to "yes", resulting in the VID allocated to the this address being distinct from the VID allocated to other addresses (for the same VM or other VMs) connected to the same DCVPN on a same NVE port; this behavior will result in traffic always transiting through the

NVE, even to/from other addresses for the same DCVPN on the same server.

The "activate" operation is a dataplane operations that references a previously established association via the address and VID; all other parameters are obtained at the NVE by mapping the source address, VID and port involved to obtain information established by a prior associate operation.

Realizations of DCVPNs include, E-VPNs ([I-D.ietf-l2vpn-evpn]), IP VPNs ([RFC4364]), NVGRE ([I-D.sridharan-virtualization-nvgre], VPLS ([RFC4761], [RFC4762]), and VXLAN ([I-D.mahalingam-dutt-dcops-vxlan]). The encapsulation type determines whether forwarding at the NVE for the DCVPN is based on Layer 2 or Layer 3 service.

Typically, for the associate messages, all of the above information except hold time would be needed. Similarly, for the dissociate message, all of the above information except VID and encapsulation type would typically be needed.

These operations are stateful in that their results remain in place until superseded by another operation. For example, on receiving an associate message, an NVE is expected to create and maintain the DCVPN information for the addresses until the NVE receives a dissociate message to remove that information. A separate liveness protocol may be run between server and NVE to let each side know that the other is still operational; if the liveness protocol fails, each side may remove state installed in response to messages from the other.

The descriptions below generally assume that the NVEs participate in a mechanism for control plane distribution of VM addresses, as opposed to doing this in the data plane. If this is not the case, NVE elements can lazily evaluate (via data plane discovery) the parts of the procedures below that involve address distribution.

As VIDs are local to server-NVE communication, in fact to a specific port connecting these two elements, a mapping table containing 4-tuples of the following form will prove useful to the NVE:

<VID, port, VNID, VM network address>



The valid VID values are from 1 to 4094, inclusive. A value of 0 is used to mean "unassigned". When a VID can be shared by more than one VM, it is necessary to reference-count entries in this table; the list of addresses in an entry serves this purpose. Entries in this table have multiple uses:

- o Finding the VNID for a VID and port for association, activation and traffic forwarding;
- o Determining whether a VID exists (has already been assigned) for a VNID and port.
- o Determining which <VID, port> pairs to use for forwarding traffic that requires flooding on the DCVPN.

For simplicity and clarity, this draft assumes that the network interfaces in VMs (vNICs) do not use VLAN tags.

#### 5.2.2. Creating a VM

When a VM is instantiated on a server (powered on, e.g., after creation), each of the VM's interfaces is assigned a VNID, one or more network addresses and an encapsulation type for the DCVPN. The VM addresses may be any of IPv4, IPv6 and MAC addresses. There may also be network policies specific to the VM or its interfaces. To connect the VM to its DCVPN, the server signals these parameters to the l-NVE via an "associate" operation followed by an "activate" operation to put the parameters into use. (Note that the l-NVE may consist of more than one device.)

On receiving an associate message on port P from server S, an NVE device does the following for each network address in that message:

A.1: Validate the authentication (if present). If not, inform the provisioning system, log the error, and stop processing the associate message. This validation may include authorization checks.

A.2: Check the per-address-VID-allocation flag in the associate message:

- \* if this flag is not set:

- + Check if the VID in the associate message is zero (i.e., the associate message requests VID allocation); if so, look up the VID for <VNID, port, network address> ; if there is no current VID for that tuple, allocate a new VID

- + If the VID in the associate message is non-zero, look up the VID for <VNID, port>. If that lookup results in the same VID as the one in the associate message, associate that VID with <VNID, network address>. If the lookup indicates that there is no current VID for that tuple, associate the VID in the associate message with <VNID, port, network address>. Otherwise, the VID in the associate message does not match the VID that is currently in use for <VNID, port>, so respond to S with an error, and stop processing the associate message.
- \* if this flag is set, check if the VID in the associate message is zero :
  - + if so, this is an allocation request, so allocate a new VID, distinct from other VIDs allocated on this port;
  - + if the VID is non-zero, check that the provided VID is distinct from other VIDs allocated on this port; if so, associate the VID with <VNID, port, network address>. If not, the provided VID is already in used and hence cannot be dedicated to this network address, so respond to S with an error, and stop processing the associate message.

A.3: Add the <VID, port, VNID, network address> entry to the NVE's mapping table. This table entry includes information about the DCVPN encapsulation type for the VNID.

A.4: Communicate with the control plane to advertise the network address, and (if the VNID is new to the NVE) also to get other network addresses in the DCVPN. Populate the NVE's mapping table with all of these network addresses (some control planes may not provide all or even any of the other addresses in the DCVPN at this point).

A.5: Finally, respond to S with the VID for <VNID, port, network address>, and indicate that the operation was successful.

After a successful associate, the network has been provisioned (at least in the local NVE) for traffic, but forwarding has not been enabled. On receiving an activate message on port P from server S, an NVE device does the following (activate is a one-way message that does not have a response):

B.1: Validate the authentication (if present). If not, inform the provisioning system, log the error, and stop processing the associate message. This validation may include authorization checks. The authentication and authorization may be implicit when

the activate message is a dataplane frame (e.g., a "gratuitous" ARP or RARP).

- B.2: Check if the VID in the activate message is zero. If so, log the error, and stop processing the activate message.
- B.3: Use the VID and port P to look up the VNID from a previous associate message. If there is no mapping table state for that VID and port, log the error and stop processing the activate message.
- B.4: If forwarding is not enabled for <VID, port, network address> activate it, mapping VID -> VNID on this port (P) for traffic sent to and received from r-NVEs.
- B.5: If the activate message is a dataplane frame that requires forwarding beyond the NVE, (e.g., a "gratuitous" ARP or RARP), use the activated forwarding to send the frame onward via the virtual network identified by the VNID.

#### 5.2.3. Terminating a VM

On receiving a request from the provisioning system to terminate execution of a VM (powering off the VM, whether or not the VM's image is retained on storage), the server sends a dissociate message to the l-NVE with the hold time set to zero. The dissociate message contains the operation, authentication, VNID, encapsulation type, and VM addresses. On receiving the dissociate message on port P from server S, each NVE device L does the following:

- D.1: Validate the authentication (if present). If not, inform the provisioning system, log the error, and stop processing the associate message.
- D.2: Communicate with the control plane to withdraw the VM's addresses. If the hold time is as non-zero, wait until the hold time expires before proceeding to the next step.
- D.3: Delete the VM's addresses from the mapping table and delete any VM-specific network policies associated with any of the VM addresses. If a mapping tuple contains no VM addresses as a result delete that tuple. If the mapping table contains no entries for the VNID involved after deleting the tuple, optionally delete any network policies for the VNID.
- D.4: Respond to S saying that the operation was successful.

At step D.2, the control plane is responsible for not disrupting network operation if the addresses are in use at another l-NVE. Also, l-NVEs cannot rely on receiving dissociate messages for all terminated VMs, as a server crash may implicitly terminate a VM before a dissociate message can be sent.

#### 5.2.4. Migrating a VM

Consider a VM that is being migrated from server S (connected to l-NVE device L) to server S' (connected to l-NVE device L'). This section assumes shared storage, so that both S and S' have access to the VM's storage. The sequence of steps for a successful VM migration is:

- M.1: S' gets a request to prepare to receive a copy of the VM from S.
- M.2: S gets a request to copy the VM to S'.
- M.3: The copy of the VM (memory, configuration state, etc.) occurs while the VM continues to execute.
- M.4: When that copy has made sufficient progress, S pauses the VM, and completes the copy, including the VM's execution state.
- M.5: S' gets a request to resume the paused VM.
- M.6: After that resume has succeeded, S then proceeds to terminate the paused VM on S, see section Section 5.2.3, but this operation may specify a non-zero hold time during which traffic received may be forwarded to the VM's new location.

Steps M.1 and M.2 initiate the copy of the VM. During step M.3, S' sends an "associate" message to L' for each of the VM's network addresses (S' receives information about these addresses as part of the VM copy). Step M.4 occurs when the VM copy has made sufficient progress that the pause required to transfer the VM's execution from S to S' is sufficiently short. At step M.4, or M.5 at the latest, S' sends an "activate" message to L' for each of the VM's interfaces. At Step M.6, S sends a "dissociate" message to L for each of the VM's network addresses, optionally with a non-zero hold time.

From the DCVPN's view, there are two important overlaps in the apparent network location of the VM's addresses:

- o The VM's addresses are associated with both L and L' between steps M.3 and M.6.

- o The VM's addresses are activated at L' during step M.4 or step M.5 at the latest (e.g., if activate is a dataplane operation based on traffic sent at that step); both of these typically occur before these addresses are dissociated at L during step M.6

The DCVPN control plane must work correctly in the presence of these overlaps, and in particular must not:

- o Fail to activate the VM's network addresses at L' because they have not yet been withdrawn at L, or
- o Disruptively withdraw the VM's network addresses from use at step M.6 of a migration when the VM continues to execute on a different server.

An additional scenario that is important for migration is that the source and destination servers, S and S', may share a common l-NVE, i.e., L and L' are the same. In this scenario there is no need for remote interaction of that l-NVE with other NVEs, but that NVE must be aware of the possibility of a new association of the VM's addresses with a different port and the need to promptly activate them on that port even though they have not (yet) been dissociated from their original port.

### 5.3. Signaling Protocols

There are multiple protocols that can be used to signal the above messages. One could invent a new protocol for this purpose, or reuse existing protocols, among them LLDP, XMPP, HTTP REST, and VDP [VDP], a new protocol standardized for the purposes of signaling a VM's network parameters from server to l-NVE. Multiple factors influence the choice of protocol(s); this draft's focus is on what needs to be signaled, leaving choices of how the information is signaled, and specific encodings for other drafts to consider.

## 6. Interfacing with DCVPN Control Planes

The control plane for a DCVPN manages the creation/deletion, membership and span of the DCVPN ([I-D.ietf-nvo3-overlay-problem-statement],[I-D.kreeger-nvo3-overlay-cp]). Such a control plane needs to work with the server-to-nve signaling in a coordinated manner, to ensure that address changes at a local NVE are reflected appropriately in remote NVEs. The details of such coordination are specified in separate documents.

## 7. Security Considerations

## 8. IANA Considerations

## 9. Acknowledgments

Many thanks to Amit Shukla for his help with the details of EVB and his insight into data center issues. Many thanks to members of the nvo3 WG for their comments, including Yingjie Gu.

## 10. Informative References

## [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-03 (work in progress), February 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-02 (work in progress), February 2013.

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

Narten, T., Gray, E., Black, D., Dutt, D., Fang, L., Kreeger, L., Napierala, M., and M. Sridharan, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-02 (work in progress), February 2013.

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

Kreeger, L., Dutt, D., Narten, T., and M. Sridharan, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp-02 (work in progress), October 2012.

## [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-03 (work in progress), February 2013.

## [I-D.sridharan-virtualization-nvgre]

Sridharan, M., Greenberg, A., Venkataramaiah, N., Wang, Y., Duda, K., Ganga, I., Lin, G., Pearson, M., Thaler, P., and C. Tumuluri, "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-02 (work in progress), February 2013.

- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, 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.
- [VDP] IEEE, "Edge Virtual Bridging (IEEE Std 802.1Qbg-2012)", July 2012.

## Authors' Addresses

Kireeti Kompella  
Juniper Networks  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: kireeti@juniper.net

Yakov Rekhter  
Juniper Networks  
1194 N. Mathilda Ave.  
Sunnyvale, CA 94089  
US

Email: yakov@juniper.net

Thomas Morin  
France Telecom - Orange Labs  
2, avenue Pierre Marzin  
Lannion 22307  
France

Email: thomas.morin@orange.com

David L. Black  
EMC Corporation  
176 South St.  
Hopkinton, MA 01748

Email: david.black@emc.com



Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: August 29, 2013

L. Kreeger  
Cisco  
T. Narten  
IBM  
D. Black  
EMC  
February 25, 2013

Network Virtualization Hypervisor-to-NVE Overlay Control Protocol  
Requirements  
draft-kreeger-nvo3-hypervisor-nve-cp-01

Abstract

The document "Problem Statement: Overlays for Network Virtualization" discusses the needs for network virtualization using overlay networks in highly virtualized data centers. The problem statement outlines a need for control protocols to facilitate running these overlay networks. This document outlines the high level requirements related to the interaction between hypervisors and the Network Virtualization Edge device when the two entities are not co-located on the same physical device.

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 August 29, 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.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	4
3. Entity Relationships . . . . .	6
3.1. VNIC Containment Relationship . . . . .	6
3.1.1. Layer 2 Virtual Network Service . . . . .	7
3.1.2. Layer 3 Virtual Network Service . . . . .	8
4. Hypervisor-to-NVE Control Plane Protocol Functionality . . . . .	9
4.1. VN Connect/Disconnect . . . . .	11
4.2. VNIC Address Association . . . . .	12
4.3. VNIC Address Disassociation . . . . .	12
4.4. VNIC Shutdown/Startup/Migration . . . . .	13
4.5. VN Profile . . . . .	14
5. Security Considerations . . . . .	14
6. Acknowledgements . . . . .	14
7. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

Note: the contents of this document were originally in [I-D.kreeger-nvo3-overlay-cp]. The content has been pulled into its own document because the problem area covered is distinct and different from what most folk think of as a "control protocol" for NVO3. Other related documents on this same general topic include [I-D.kompella-nvo3-server2nve], [I-D.gu-nvo3-overlay-cp-arch], and [I-D.gu-nvo3-tes-nve-mechanism].

"Problem Statement: Overlays for Network Virtualization" [I-D.ietf-nvo3-overlay-problem-statement] discusses the needs for network virtualization using overlay networks in highly virtualized data centers and provides a general motivation for building such networks. "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. The reader is assumed to be familiar with both documents.

Section 4.5 of [I-D.ietf-nvo3-overlay-problem-statement] describes three separate work areas that fall under the general category of a control protocol for NVO3. This document focuses entirely on the control protocol related to the hypervisor-to-NVE interaction, labeled as the "third work item" in [I-D.ietf-nvo3-overlay-problem-statement]. Requirements for the interaction between an NVE and the "oracle" are described in [I-D.kreeger-nvo3-overlay-cp].

The NVO3 WG needs to decide on a better term for "oracle". This document will use Information Mapping Authority (IMA) until a decision is made.

This document uses the term "hypervisor" throughout when describing the scenario where NVE functionality is implemented on a separate device from the "hypervisor" that contains a VM connected to a VN. In this context, the term "hypervisor" is meant to cover any device type where the NVE functionality is offloaded in this fashion, e.g., a Network Service Appliance.

This document often uses the term "VM" and "Tenant System" (TS) interchangeably, even though a VM is just one type of Tenant System that may connect to a VN. For example, a service instance within a Network Service Appliance may be another type of TS. When this document uses the term VM, it will in most cases apply to other types of TSs.

## 2. Terminology

This document uses the same terminology as found in the NV03 Framework document, [I-D.ietf-nvo3-framework]. 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.

IMA: Information Mapping Authority.

[I-D.ietf-nvo3-overlay-problem-statement] uses the term "oracle" to describe this. It is 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.

Tenant System: 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 Device: A physical system to which networking service is provided. Examples include hosts (e.g. server or server blade), storage systems (e.g., file servers, iSCSI storage systems), and network devices (e.g., firewall, load-balancer, IPSec gateway). An end device may include internal networking functionality that interconnects the device's components (e.g. virtual switches that interconnect VMs running on the same server). NVE functionality may be implemented as part of that internal networking.

Network Service Appliance: A stand-alone physical device or a virtual device that provides a network service, such as a firewall, load balancer, etc. Such appliances may embed Network Virtualization Edge (NVE) functionality within them in order to more efficiently operate as part of a virtualized network.

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

VDC: Virtual Data Center. A container for virtualized compute, storage and network services. Managed by a single tenant, a VDC can contain multiple VNs and multiple Tenant Systems that are connected to one or more of these VNs.

VN Alias: A string name for a VN as used by administrators and customers to name a specific VN. A VN Alias is a human-usable string that can be listed in contracts, customer forms, email, configuration files, etc. and that can be communicated easily

vocally (e.g., over the phone). A VN Name is independent of the underlying technology used to implement a VN and will generally not be carried in protocol fields of control protocols used in virtual networks. Rather, a VN Alias will be mapped into a VN Name where precision is required.

**VN Name:** A globally unique identifier for a VN suitable for use within network protocols. A VN Name will usually be paired with a VN Alias, with the VN Alias used by humans as a shorthand way to name and identify a specific VN. A VN Name should have a compact representation to minimize protocol overhead where a VN Name is carried in a protocol field. Using a Universally Unique Identifier (UUID) as discussed in RFC 4122, may work well because it is both compact and a fixed size and can be generated locally with a very high likelihood of global uniqueness.

**VN ID:** A unique and compact identifier for a VN within the scope of a specific NVO3 administrative domain. It will generally be more efficient to carry VN IDs as fields in control protocols than VN Aliases. There is a one-to-one mapping between a VN Name and a VN ID within an NVO3 Administrative Domain. Depending on the technology used to implement an overlay network, the VN ID could be used as the Context Identifier in the data plane, or would need to be mapped to a locally-significant Context Identifier.

**VN Profile:** Meta data associated with a VN that is used by an NVE when ingressing/egressing packets to/from a specific VN. Meta data could include such information as ACLs, QoS settings, etc. The VN Profile contains parameters that apply to the VN as a whole. Control protocols could use the VN ID or VN Name to obtain the VN Profile.

**VNIC:** A Virtual NIC that connects a Tenant System to a Virtual Network Instance (VNI). Virtual NICs have virtual MAC addresses that may not be globally unique, but must be unique within a VN for proper network operation.

**VNIC Name:** A globally unique identifier for a VNIC suitable for use within network protocols. Note that because VNIC MAC addresses may not be globally unique, they cannot be used as the VNIC Name. A VNIC Name should have a compact representation to minimize protocol overhead where a VNIC Name is carried in a protocol field. Using a Universally Unique Identifier (UUID) as discussed in RFC 4122, may work well because it is both compact and a fixed size and can be generated locally with a very high likelihood of global uniqueness.

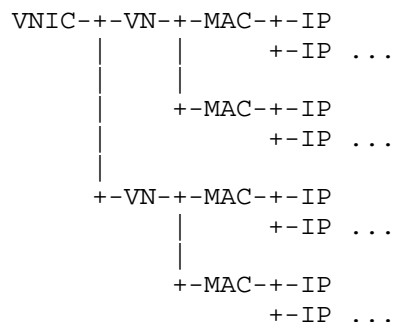
### 3. Entity Relationships

This section describes the relationships between the entities involved in the Hypervisor-to-NVE control protocol.

#### 3.1. VNIC Containment Relationship

The root of the containment tree is a VNIC. Even though a VM may have multiple VNICs, from the point of view of an NVE, each VNIC can be treated independently. There is no need to identify the VM itself within the Hypervisor-to-NVE protocol.

Each VNIC can connect to multiple VNs. Within each VNIC-VN pair, multiple MAC addresses may be reachable. Within each VNIC-VN-MAC triplet, there may be multiple IP addresses. This containment hierarchy is depicted below.



VNIC Containment Relationship

Figure 1

Any of these entities can be added or removed dynamically at any time.

The relationship implies that if one entity in the hierarchy is deleted then all the entities it contains are also deleted. For example, if a given VNIC disassociates from one VN, all the MAC and IP addresses are also disassociated. There is no need to signal the deletion of every entity within a VNIC when the VNIC is brought down or deleted (or the VM it is attached to is powered off or migrates away from the hypervisor).

If a VNIC provides connectivity to a range of IP addresses (e.g. the VM is a load balancer with many Virtual IP addresses), it will be more efficient to signal a range or address mask in place of

individual IP addresses.

In the majority of cases, a VM will be acting as a simple host that will have the following containment tree:

VNIC--VN--MAC--IP

Figure 2

Since this is the most common case, the Hypervisor-to-NVE protocol should be optimized to handle this case.

Tenant Systems (TS) that are providing network services (such as firewall, load balancer, VPN gateway) are likely to have a more complex containment hierarchy. For example, a TS acting as a load balancer is quite likely to terminate multiple IP addresses, one for each application, or farm of servers that it is providing the front end for.

Hypervisors often have a limit on the number of VNICS that a VM can have (e.g. in the range of 8 to 10 VNICS). If a VM has the need to connect to more networks than the number of VNICS the hypervisor supports, the solution is often to configure the VNIC (and the associated virtual port on the virtual switch the VNIC connects to) as an 802.1Q trunk. In the case of a virtual switch that supports only VLANs, the VLAN tags used by all the VNICS connected to the switch (as well as the bridged network the hypervisor is physically connected to) share a common VLAN ID.

In a multi-tenant scenario using overlay Virtual Networks instead of VLANs, VNICS can still use 802.1Q tagging to isolate traffic from different VNs as it crosses the virtual link between the VNIC and the virtual switch; However, The tags would have only local significance across that virtual link, with the virtual switch mapping each tag value to a different VN. This implies that two different virtual links may use different 802.1Q tag values but with each mapped to the same VN by the virtual switch. Similarly, two VNICS could use the same VLAN tag value but the virtual switch can map each vPort/Tag pair to a different VN.

Each VNIC must attach to at least one VN and have at minimum one MAC address. An IP address can be optional depending on whether the VN is providing L2 or L3 service.

### 3.1.1. Layer 2 Virtual Network Service

When the Virtual Network is providing only Layer 2 forwarding, the NVEs only require knowledge of the Tenant System's MAC addresses,

while layer 3 termination and routing happens only in the Tenant Systems.

For example, if a VM is acting as a router to connect together two layer 2 VNs, the overlay system will forward frames to this router VM based on the VNIC's MAC address, but inside the frames may be packets destined to many different IP addresses. There is no need for the NVEs to know the IP address of the router VM itself, nor the IP addresses of other TS that have packets routing through the VM. However, it may be useful for the NVE to know the IP address of the router itself for either troubleshooting, or for providing other network optimizations such as local termination of ARP (even though ARP optimizations are not strictly layer 2). It is recommended (but optional) for an End Device to provide an IP address for a VNIC even if the NVE is providing an L2 service.

When the overlay VN is forwarding at layer 2, it is possible for Tenant Systems to perform bridging between two VNs belonging to that tenant (provided the tenant MAC addresses do not overlap between the two VNs that are being bridged). Reasons for VMs to do this are the same as in the physical world, such as the insertion of a transparent firewall device. For example, a VM running firewall software can be inserted in between two groups of Tenant Systems on the same subnet by putting each group on a different VN and having the firewall VM bridge between them.

When a VM is acting as a transparent bridge, it will appear to the overlay system that the VM is terminating multiple MAC addresses - one for each TS that exists on the other VN the VM is bridging to. In order for the overlay system to properly forward traffic to the bridging VM, it must know the MAC addresses of all the tenant systems the VM is bridging towards. This is one case where a VNIC can appear to terminate more than one MAC address for the same VN/VNIC.

### 3.1.2. Layer 3 Virtual Network Service

When the Virtual Network is providing Layer 3 forwarding, the NVEs must have knowledge of the Tenant System IP addresses. In the case where there is a Tenant System providing L3 forwarding for the tenant (e.g. an L3 VPN gateway), The TS VNIC may only terminate frames with a single MAC address, but will be forwarding IP packets on the behalf of other Tenant Systems. This scenario requires more exploration to determine how the TS forwarding interacts with the VN forwarding; However, in one scenario, the TS VNIC may be seen as containing many IP addresses.

Note that a MAC address is required even for a pure L3 VN service because VNICs filter out frames with destination MAC addresses that



do not match the VNIC's address; Therefore, the NVE providing an L3 service must first encapsulate an IP packet in an Ethernet frame with the VNIC's destination MAC before it is sent to the End Device containing the VNIC.

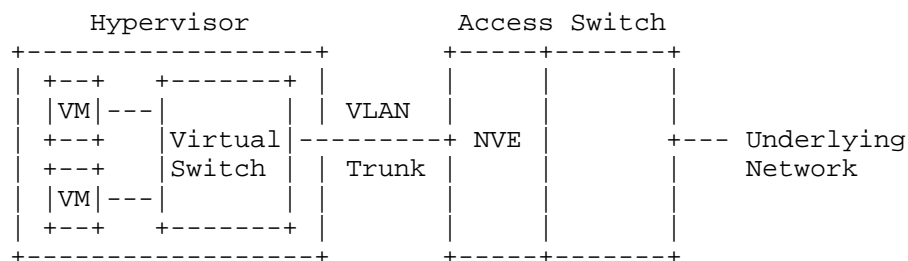
#### 4. Hypervisor-to-NVE Control Plane Protocol Functionality

The problem statement [I-D.ietf-nvo3-overlay-problem-statement], discusses the needs for a control plane protocol (or protocols) to populate each NVE with the state needed to perform its functions.

In one common scenario, an NVE provides overlay encapsulation/decapsulation packet forwarding services to Tenant Systems (TSs) that are co-resident with the NVE on the same End Device (e.g. when the NVE is embedded within a hypervisor or a Network Service Appliance). In such cases, there is no need for a standardized protocol between the hypervisor and NVE, as the interaction is implemented via software on a single device.

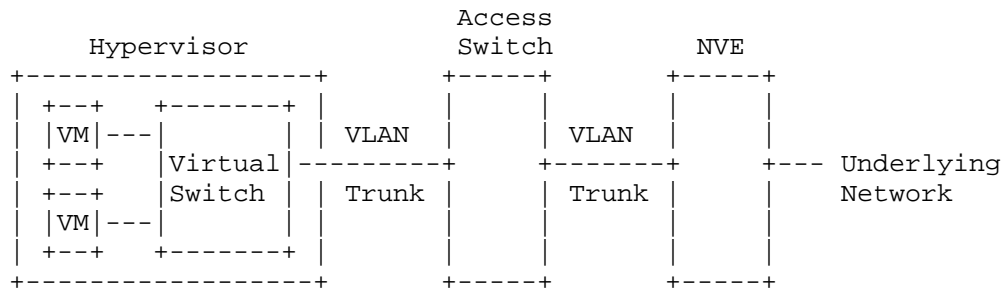
Alternatively, a Tenant System may use an externally connected NVE. An external NVE can provide an offload of the encapsulation / decapsulation function, network policy enforcement, as well as the VN Overlay protocol overheads. This offloading may provide performance improvements and/or resource savings to the End Device (e.g. hypervisor) making use of the external NVE.

The following figures give example scenarios where the Tenant System and NVE are on different devices separated by an access network.



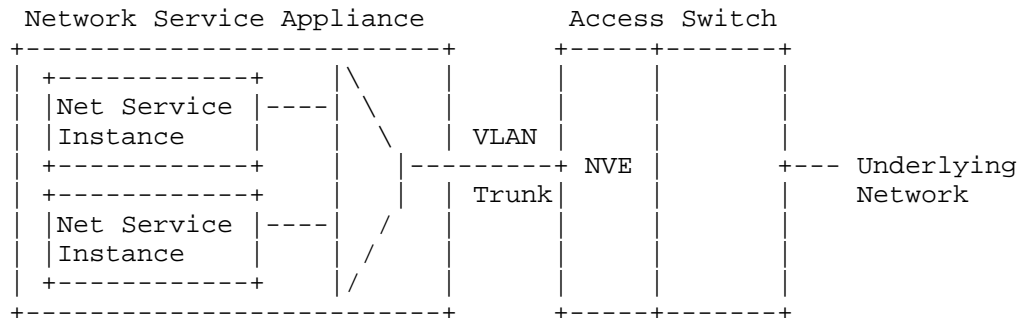
Hypervisor with an External NVE.

Figure 3



Hypervisor with an External NVE across an Ethernet Access Switch.

Figure 4



Physical Network Service Appliance with an External NVE.

Figure 5

In the examples above, the physical VLAN Trunk from the Hypervisor or Network Services Appliance towards the external NVE only needs to carry locally significant VLAN tag values. How "local" the significance is depends on whether the Hypervisor has a direct physical connection to the NVE (in which case the significance is local to the physical link), or whether there is an Ethernet switch (e.g. a blade switch) connecting the Hypervisor to the NVE (in which case the significance is local to the intervening switch and all the links connected to it).

These VLAN tags are used to differentiate between different VNs as packets cross the shared access network to the external NVE. When the NVE receives packets, it uses the VLAN tag to identify the VN of packets coming from a given Tenant System's VNIC, strips the tag, and

adds the appropriate overlay encapsulation for that VN.

On the hypervisor-facing side of the NVE, a control plane protocol is necessary to provide an NVE with the information it needs to provide connectivity across the Virtual Network for a given VNIC. Specifically, the Hypervisor (or Network Service Appliance) utilizing an external NVE needs to "attach to" and "detach from" a VN, as well as communicate the addresses within that VN that are reachable within it. Thus, they will need a protocol that runs across the access network between the two devices that identifies the Tenant System (TS) VNIC addresses and VN Name (or ID) for which the NVE is providing service. In addition, such a protocol will identify a locally significant tag (e.g., an 802.1Q VLAN tag) that can be used to identify the data frames that flow between the TS VNIC and the VN.

#### 4.1. VN Connect/Disconnect

In the previous figures, NVEs reside on an external networking device (e.g. an access switch). When an NVE is external, a protocol is needed between the End Device (e.g. Hypervisor) making use of the external NVE and the external NVE in order to make the NVE aware of the changing VN membership requirements of the Tenant Systems within the End Device.

A key driver for using a protocol rather than using static configuration of the external NVE is because the VN connectivity requirements can change frequently as VMs are brought up, moved and brought down on various hypervisors throughout the data center.

The NVE must be notified when an End Device requires connection to a particular VN and when it no longer requires connection. In addition, the external NVE must provide a local tag value for each connected VN to the End Device to use for exchange of packets between the End Device and the NVE (e.g. a locally significant 802.1Q tag value).

The Identification of the VN in this protocol could either be through a VN Name or a VN ID. A globally unique VN Name facilitates portability of a Tenant's Virtual Data Center. When a VN within a VDC is instantiated within a particular administrative domain, it can be allocated a VN Context which only the NVE needs to use. Once an NVE receives a VN connect indication, the NVE needs a way to get a VN Context allocated (or receive the already allocated VN Context) for a given VN Name or ID (as well as any other information needed to transmit encapsulated packets). How this is done is the subject of the NVE-to-oracle (called NVE-to-IMA in this document) protocol which are part of work items 1 and 2 in [I-D.ietf-nvo3-overlay-problem-statement].

An End Device that is making use of an offloaded NVE only needs to communicate the VN Name or ID to the NVE, and get back a locally significant tag value.

#### 4.2. VNIC Address Association

Typically, a VNIC is assigned a single MAC address and all frames transmitted and received on that VNIC use that single MAC address. As discussed in the section above on the containment hierarchy, it is also possible for a Tenant System to exchange frames using multiple MAC addresses (ones that are not assigned to the VNIC) or packets with multiple IP addresses.

Particularly in the case of a TS that is forwarding frames or packets from other TSs, the NVE will need to communicate the mapping between the NVE's IP address (on the underlying network) and ALL the addresses the TS is forwarding on behalf of to the Information Mapping Authority (IMA).

The NVE has two ways in which it can discover the tenant addresses for which frames must be forwarded to a given End Device (and ultimately to the TS within that End Device).

1. It can glean the addresses by inspecting the source addresses in packets it receives from the End Device.
2. The End Device can explicitly signal the addresses to the NVE. The End Device could have discovered the addresses for a given VNIC by gleaning them itself from data packets sent by the VNIC, or by some other internal means within the End Device itself.

To perform the second approach above, the "hypervisor-to-NVE" protocol requires a means to allow End Devices to communicate new tenant addresses associations for a given VNIC within a given VN.

#### 4.3. VNIC Address Disassociation

When a VNIC within an End Device terminates function (due to events such as VNIC shutdown, Tenant System (TS) shutdown, or VM migration to another hypervisor), all addresses associated with that VNIC must be disassociated with the End Device on the connected NVE.

If the VNIC only has a single address associated with it, then this can be a single address disassociate message to the NVE. However, if the VNIC had hundreds of addresses associated with it, then the protocol with the NVE would be better optimized to simply disassociate the VNIC with the NVE, and the NVE can automatically disassociate all addresses that were associated with the VNIC.

Having TS addresses associated with a VNIC can also provide scalability benefits when the VM migrates between hypervisors that are connected to the same NVE. When a VM migrates to another hypervisor connected to the same NVE, if the NVE is aware of the migration, there is no need for all the addresses to be purged from NVE (and IMA) only to be immediately re-established again when the VM migration completes.

If the device containing the NVE is supporting many hypervisors, it may be quite likely that the VM migration will result in the VNICs still being associated with the same NVE, but simply on a different port. From the point of view of the IMA, nothing has changed and it would be inefficient to signal these changes to the IMA for no benefit. The NVE only needs to associate the addresses with a different port/tag pair.

It is possible for the NVE to handle a VM migration by using a timer to retain the VNIC addresses for a short time to see if the disassociated VNIC re-associates on another NVE port, but this could be better handled if the NVE knew the difference between a VNIC/VM shutdown and a VM migration. This leads to the next section.

#### 4.4. VNIC Shutdown/Startup/Migration

As discussed above, the NVE can make optimizations if it knows which addresses are associated with which VNICs within an End Device and also is notified of state changes of that VNIC, specifically the difference between VNIC shutdown/startup and VNIC migration arrival/departure.

Upon VNIC shutdown, the NVE can immediately signal to the IMA that the bindings of the VNIC's addresses to the NVE's IP address can be removed.

Upon VNIC arrival, the NVE could either start a timer to hold the VNIC address bindings waiting to see if the VNIC arrives on a different port, or if there is a pre-arrival handshake with the NVE, then it will already know that the VNIC is going to be reassociated with the same NVE.

Upon VNIC arrival, the NVE knows that any addresses previously bound to the VNIC are still present and has no need to signal any change in address mappings to the IMA.

Note that if the IMA is also aware of VNIC address bindings, it can similarly participate efficiently in a VM migration that occurs across two different NVEs.

#### 4.5. VN Profile

Once an NVE (embedded or external) receives a VN connect indication with a specified VN Name or ID, the NVE must determine the VN Context value to encapsulate packets with as well as other information that may be needed (e.g., QoS settings). The NVE serving that hypervisor needs a way to get a VN Context allocated or receive the already allocated VN Context for a given VN Name or ID (as well as any other information needed to transmit encapsulated packets). A protocol for an NVE to get this mapping may be a useful function, but would be the subject of work items 1 and 2 in [I-D.ietf-nvo3-overlay-problem-statement].

#### 5. Security Considerations

Editor's Note: This is an initial start on the security considerations section; it will need to be expanded, and suggestions for material to add are welcome.

NVEs must ensure that only properly authorized Tenant Systems are allowed to join and become a part of any specific Virtual Network. In addition, NVEs will need appropriate mechanisms to ensure that any hypervisor wishing to use the services of an NVE are properly authorized to do so. One design point is whether the hypervisor should supply the NVE with necessary information (e.g., VM addresses, VN information, or other parameters) that the NVE uses directly, or whether the hypervisor should only supply a VN ID and an identifier for the associated VM (e.g., its MAC address), with the NVE using that information to obtain the information needed to validate the hypervisor-provided parameters or obtain related parameters in a secure manner.

#### 6. Acknowledgements

Thanks to the following people for reviewing and providing feedback: Vipin Jain and Shyam Kapadia.

#### 7. Informative References

[I-D.gu-nvo3-overlay-cp-arch]  
Yingjie, G. and W. Hao, "Analysis of external assistance to NVE and consideration of architecture",  
draft-gu-nvo3-overlay-cp-arch-00 (work in progress),  
July 2012.

[I-D.gu-nvo3-tes-nve-mechanism]

Yingjie, G. and L. Yizhou, "The mechanism and signalling between TES and NVE", draft-gu-nvo3-tes-nve-mechanism-01 (work in progress), October 2012.

[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-02 (work in progress), February 2013.

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

Narten, T., Gray, E., Black, D., Dutt, D., Fang, L., Kreeger, L., Napierala, M., and M. Sridharan, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-02 (work in progress), February 2013.

[I-D.kompella-nvo3-server2nve]

Kompella, K., Rekhter, Y., and T. Morin, "Signaling Virtual Machine Activity to the Network Virtualization Edge", draft-kompella-nvo3-server2nve-01 (work in progress), October 2012.

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

Kreeger, L., Dutt, D., Narten, T., and M. Sridharan, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp-02 (work in progress), October 2012.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

Authors' Addresses

Lawrence Kreeger  
Cisco

Email: kreeger@cisco.com

Thomas Narten  
IBM

Email: narten@us.ibm.com

Internet-Draft NV03 Hypervisor-NVE Control Protocol Reqs February 2013

David Black  
EMC

Email: david.black@emc.com





Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: December 16, 2013

L. Kreeger  
Cisco  
D. Dutt  
Cumulus Networks  
T. Narten  
IBM  
D. Black  
EMC  
M. Sridharan  
Microsoft  
June 14, 2013

Network Virtualization Overlay Control Protocol Requirements  
draft-kreeger-nvo3-overlay-cp-04

Abstract

The document "Problem Statement: Overlays for Network Virtualization" discusses the needs for network virtualization using overlay networks in highly virtualized data centers. The problem statement outlines a need for control protocols to facilitate running these overlay networks. This document outlines the high level requirements to be fulfilled by the control protocols related to building and managing the mapping tables and other state information used by the Network Virtualization Edge to transmit encapsulated packets across the underlying network.

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 December 16, 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.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Control Plane Protocol Functionality . . . . .	4
3.1. Inner to Outer Address Mapping . . . . .	5
3.2. Underlying Network Multi-Destination Delivery Address(es) . . . . .	6
3.3. VN Connect/Disconnect Notification . . . . .	6
3.4. VN Name to VN ID Mapping . . . . .	6
4. Control Plane Characteristics . . . . .	7
5. Security Considerations . . . . .	9
6. Acknowledgements . . . . .	9
7. Informative References . . . . .	10
Authors' Addresses . . . . .	10

## 1. Introduction

"Problem Statement: Overlays for Network Virtualization" [I-D.ietf-nvo3-overlay-problem-statement] discusses the needs for network virtualization using overlay networks in highly virtualized data centers and provides a general motivation for building such networks. "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. The reader is assumed to be familiar with both documents.

Section 4.5 of [I-D.ietf-nvo3-overlay-problem-statement] describes three separate work areas that fall under the general category of a control protocol for NVO3. This document focuses entirely on those aspects of the control protocol related to the building and distributing the mapping tables an NVE uses to tunnel traffic from one VM to another. Specifically, this document focuses on work areas 1 and 2 given in Section 4.5 of [I-D.ietf-nvo3-overlay-problem-statement]. Work areas 1 and 2 cover

the interaction between an NVE and the Network Virtualization Authority (NVA) (work area 2) or operation of the NVA itself (work area 1). Requirements related to interaction between a hypervisor and NVE when the two entities reside on separate physical devices (work area 3) are covered in [I-D.kreeger-nvo3-hypervisor-nve-cp-req].

## 2. Terminology

This document uses the same terminology as found in [I-D.ietf-nvo3-framework]. This section defines additional terminology used by this document.

**Network Service Appliance:** A stand-alone physical device or a virtual device that provides a network service, such as a firewall, load balancer, etc. Such appliances may embed Network Virtualization Edge (NVE) functionality within them in order to more efficiently operate as part of a virtualized network.

**VN Alias:** A string name for a VN as used by administrators and customers to name a specific VN. A VN Alias is a human-usable string that can be listed in contracts, customer forms, email, configuration files, etc. and that can be communicated easily vocally (e.g., over the phone). A VN Alias is independent of the underlying technology used to implement a VN and will generally not be carried in protocol fields of control protocols used in virtual networks. Rather, a VN Alias will be mapped into a VN Name where precision is required.

**VN Name:** A globally unique identifier for a VN suitable for use within network protocols. A VN Name will usually be paired with a VN Alias, with the VN Alias used by humans as a shorthand way to name and identify a specific VN. A VN Name should have a compact representation to minimize protocol overhead where a VN Name is carried in a protocol field. Using a Universally Unique Identifier (UUID) as discussed in RFC 4122, may work well because it is both compact and a fixed size and can be generated locally with a very high likelihood of global uniqueness.

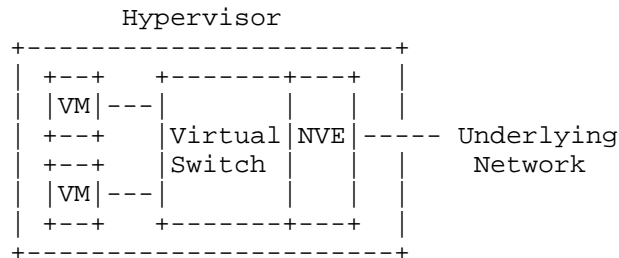
**VN ID:** A unique and compact identifier for a VN within the scope of a specific NVO3 administrative domain. It will generally be more efficient to carry VN IDs as fields in control protocols than VN Names or VN Aliases. There is a one-to-one mapping between a VN Name and a VN ID within an NVO3 Administrative Domain. Depending on the technology used to implement an overlay network, the VN ID could be used as the VN Context in the data plane, or would need to be mapped to a locally-significant context ID.

### 3. Control Plane Protocol Functionality

The NV03 problem statement [I-D.ietf-nvo3-overlay-problem-statement], discusses the needs for a control plane protocol (or protocols) to populate each NVE with the state needed to perform its functions.

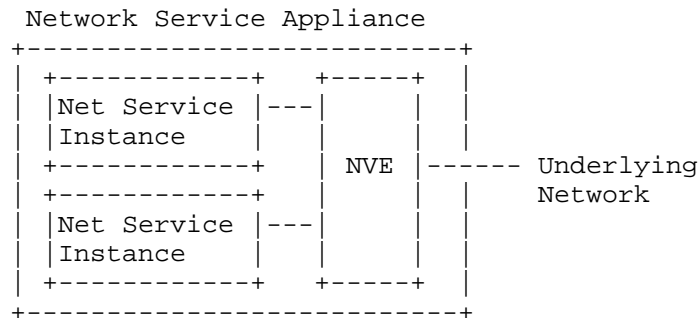
In one common scenario, an NVE provides overlay encapsulation/decapsulation packet forwarding services to Tenant Systems that are co-resident with the NVE on the same End Device (e.g. when the NVE is embedded within a hypervisor or a Network Service Appliance). Alternatively, a Tenant System may use an externally connected NVE (e.g. an NVE residing on a physical Network Switch connected to the hypervisor via an access network). The latter scenario is not discussed in this document, but is covered in [I-D.kreeger-nvo3-hypervisor-nve-cp-req].

The following figures show examples of scenarios in which the NVE is co-resident within the same End Device as the Tenant System connected to a given VN.



Hypervisor with an Embedded NVE.

Figure 1



Network Service Appliance (physical or virtual) with an Embedded NVE.

Figure 2

To support an NVE, a control plane protocol is necessary to provide an NVE with the information it needs to maintain its own internal state necessary to carry out its forwarding functions as explained in detail below.

1. An NVE maintains a per-VN table of mappings from Tenant System (inner) addresses to Underlying Network (outer) addresses of remote NVEs.
2. An NVE maintains per-VN state for delivering tenant multicast and broadcast packets to other Tenant Systems. Such state could include a list of multicast addresses and/or unicast addresses on the Underlying Network for the NVEs associated with a particular VN.
3. End Devices (such as a Hypervisor or Network Service Appliance) utilizing an external NVE need to "attach to" and "detach from" an NVE. Specifically, a mechanism is needed to notify an NVE when a Tenant System attaches to or detaches from a specific VN. Such a mechanism would provide the necessary information to the NVE that it needs to provide service to a particular Tenant System. The details of such a mechanism are out-of-scope for this document and are covered in [I-D.kreeger-nvo3-hypervisor-nve-cp-req].
4. An NVE needs a mapping from each unique VN name to the VN Context value used within encapsulated data packets within the administrative domain that the VN is instantiated.

### 3.1. Inner to Outer Address Mapping

When presented with a data packet to forward to a Tenant System within a VN, the NVE needs to know the mapping of the Tenant System destination (inner) address to the (outer) address on the Underlying Network of the remote NVE which can deliver the packet to the destination Tenant System. In addition, the NVE needs to know what VN Context to use when sending to a destination Tenant System.

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 when one Tenant System is a VM.

Note that one technique that could be used to create this mapping without the need for a control protocol is via data plane learning; However, the learning approach requires packets to be flooded to all NVEs participating in the VN when no mapping exists. One goal of using a control protocol is to eliminate this flooding.

### 3.2. Underlying Network Multi-Destination Delivery Address(es)

Each NVE needs a way to deliver multi-destination packets (i.e. tenant broadcast/multicast) within a given VN to each remote NVE which has a destination Tenant System for these packets. Three possible ways of accomplishing this are:

- o Use the multicast capabilities of the Underlying Network.
- o Have each NVE replicate the packets and send a copy across the Underlying Network to each remote NVE currently participating in the VN.
- o Use one or more distribution servers that replicate the packets on the behalf of the NVEs.

Whichever method is used, a protocol is needed to provide on a per VN basis, one or more multicast addresses (assuming the Underlying Network supports multicast), and/or one or more unicast addresses of either the remote NVEs which are not multicast reachable, or of one or more distribution servers for the VN.

The protocol must also keep the list of addresses up to date in a timely manner as the set of NVEs for a given VN changes over time. For example, the set of NVEs for a VN could change as VMs power on/off or migrate to different hypervisors.

### 3.3. VN Connect/Disconnect Notification

For the purposes of this document, it is assumed that an NVE receives appropriate notifications when a Tenant System attaches to or detaches from a specific VN. The details of how that is done are orthogonal to the NVE-to-NVA control plane, so long as such notification provides the necessary information needed by the control plane. As one example, the attach/detach notification would presumably include a VN Name that identifies the specific VN to which the attach/detach operation applies to.

### 3.4. VN Name to VN ID Mapping

Once an NVE (embedded or external) receives a VN connect indication with a specified VN Name, the NVE must determine what VN Context

value and other necessary information to use to forward Tenant System traffic to remote NVEs. In one approach, the NVE-to-NVA protocol uses VN Names directly when interacting, with the NVA providing such information as the VN Context (or VN ID) along with egress NVE's address. Alternatively, it may be desirable for the NVE-to-NVA protocol to use a more compact representation of the VN name, that is, a VN ID. In such a case, a specific NVE-to-NVA operation might be needed to first map the VN Name into a VN ID, with subsequent NVE-to-NVA operations utilizing the VN ID directly. Thus, it may be useful for the NVE-to-NVA protocol to support an operation that maps VN Names into VN IDs.

#### 4. Control Plane Characteristics

NVEs are expected to be implemented within both hypervisors (or Network Service Appliances) and within access switches. Any resources used by these protocols (e.g. processing or memory) takes away resources that could be better used by these devices to perform their intended functions (e.g. providing resources for hosted VMs).

A large scale data center may contain hundreds of thousands of these NVEs (which may be several independent implementations); Therefore, any savings in per-NVE resources can be multiplied hundreds of thousands of times.

Given this, the control plane protocol(s) implemented by NVEs to provide the functionality discussed above should have the below characteristics.

1. Minimize the amount of state needed to be stored on each NVE. The NVE should only be required to cache state that it is actively using, and be able to discard any cached state when it is no longer required. For example, an NVE should only need to maintain an inner-to-outer address mapping for destinations to which it is actively sending traffic as opposed to maintaining mappings for all possible destinations.
2. Fast acquisition of needed state. For example, when a Tenant System emits a packet destined to an inner address that the NVE does not have a mapping for, the NVE should be able to acquire the needed mapping quickly.



3. Fast detection/update of stale cached state information. This only applies if the cached state is actually being used. For example, when a VM moves such that it is connected to a different NVE, the inner to outer mapping for this VM's address that is cached on other NVEs must be updated in a timely manner (if they are actively in use). If the update is not timely, the NVEs will forward data to the wrong NVE until it is updated.
4. Minimize processing overhead. This means that an NVE should only be required to perform protocol processing directly related to maintaining state for the Tenant Systems it is actively communicating with. This requirement is for the NVE functionality only. The network node that contains the NVE may be involved in other functionality for the underlying network that maintains connectivity that the NVE is not actively using (e.g., routing and multicast distribution protocols for the underlying network).
5. Highly scalable. This means scaling to hundreds of thousands of NVEs and several million VNs within a single administrative domain. As the number of NVEs and/or VNs within a data center grows, the protocol overhead at any one NVE should not increase significantly.
6. Minimize the complexity of the implementation. This argues for using the least number of protocols to achieve all the functionality listed above. Ideally a single protocol should be able to be used. The less complex the protocol is on the NVE, the more likely interoperable implementations will be created in a timely manner.
7. Extensible. The protocol should easily accommodate extension to meet related future requirements. For example, access control or QoS policies, or new address families for either inner or outer addresses should be easy to add while maintaining interoperability with NVEs running older versions.
8. Simple protocol configuration. A minimal amount of configuration should be required for a new NVE to be provisioned. Existing NVEs should not require any configuration changes when a new NVE is provisioned. Ideally NVEs should be able to auto configure themselves.
9. Do not rely on IP Multicast in the Underlying Network. Many data centers do not have IP multicast routing enabled. If the Underlying Network is an IP network, the protocol should allow for, but not require the presence of IP multicast services within the data center.

10. Flexible mapping sources. It should be possible for either NVEs themselves, or other third party entities (e.g. data center management or orchestration systems) to create inner to outer address mappings in the NVA. The protocol should allow for mappings created by an NVE to be automatically removed from all other NVEs if it fails or is brought down unexpectedly.
11. Secure. See the Security Considerations section below.

## 5. Security Considerations

Editor's Note: This is an initial start on the security considerations section; it will need to be expanded, and suggestions for material to add are welcome.

The protocol(s) should protect the integrity of the mapping against both off-path and on-path attacks. It should authenticate the systems that are creating mappings, and rely on light weight security mechanisms to minimize the impact on scalability and allow for simple configuration.

Use of an overlay exposes virtual networks to attacks on the underlying network beyond attacks on the control protocol that is the subject of this draft. In addition to the directly applicable security considerations for the networks involved, the use of an overlay enables attacks on encapsulated virtual networks via the underlying network. Examples of such attacks include traffic injection into a virtual network via injection of encapsulated traffic into the underlying network and modifying underlying network traffic to forward traffic among virtual networks that should have no connectivity. The control protocol should provide functionality to help counter some of these attacks, e.g., distribution of NVE access control lists for each virtual network to enable packets from non-participating NVEs to be discarded, but the primary security measures for the underlying network need to be applied to the underlying network. For example, if the underlying network includes connectivity across the public Internet, use of secure gateways (e.g., based on IPsec [RFC4301]) may be appropriate.

The inner to outer address mappings used for forwarding data towards a remote NVE could also be used to filter incoming traffic to ensure the inner address sourced packet came from the correct NVE source address, allowing access control to discard traffic that does not originate from the correct NVE. This destination filtering functionality should be optional to use.

## 6. Acknowledgements

Thanks to the following people for reviewing and providing feedback:  
Fabio Maino, Victor Moreno, Ajit Sanzgiri, Chris Wright.

## 7. Informative References

[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-02 (work in progress), February 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-03 (work in progress), May 2013.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the  
Internet Protocol", RFC 4301, December 2005.

## Authors' Addresses

Lawrence Kreeger  
Cisco

Email: kreeger@cisco.com

Dinesh Dutt  
Cumulus Networks

Email: ddutt@cumulusnetworks.com

Thomas Narten  
IBM

Email: narten@us.ibm.com

David Black  
EMC

Email: david.black@emc.com

Murari Sridharan  
Microsoft

Email: [muraris@microsoft.com](mailto:muraris@microsoft.com)

Network working group  
Internet Draft  
Category: Informational

L. Yong  
Huawei  
M. Toy  
Comcast  
A. Isaac  
Bloomberg  
V. Manral  
Hewlett-Packard  
L. Dunbar  
Huawei

Expires: April 2013

October 22, 2012

## Use Cases for DC Network Virtualization Overlays

draft-mity-nvo3-use-case-04

### 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, 2013.

### Copyright Notice

Copyright (c) 2009 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

This draft describes the general NVO3 use cases. The work intention is to help validate the NVO3 framework and requirements as along with the development of the solutions.

## 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. Terminology.....	4
3. Basic Virtual Networks in a Data Center.....	4
4. Interconnecting DC Virtual Network and External Networks.....	6
4.1. DC Virtual Network Access via Internet.....	6
4.2. DC Virtual Network and WAN VPN Interconnection.....	7
5. DC Applications Using NVO3.....	9
5.1. Supporting Multi Technologies in a Data Center.....	9
5.2. Tenant Virtual Network with Bridging/Routing.....	10
5.3. Virtual Data Center (VDC).....	11
5.4. Federating NVO3 Domains.....	13
6. OAM Considerations.....	13
7. Summary.....	13
8. Security Considerations.....	14
9. IANA Considerations.....	14
10. Acknowledgements.....	14
11. References.....	15
11.1. Normative References.....	15
11.2. Informative References.....	15
Authors' Addresses.....	16

## 1. Introduction

Compute Virtualization has dramatically and quickly changed IT industry in terms of efficiency, cost, and the speed in providing a new applications and/or services. However the problems in today's data center hinder the support of an elastic cloud service and dynamic virtual tenant networks [NVO3PRBM]. The goal of DC Network Virtualization Overlays, i.e. NVO3, is to decouple a communication among tenant end systems (VMs) from DC physical networks and to allow the network infrastructure to provide: 1) traffic isolation among one virtual network and another; 2) independent address space in each virtual network and address isolation from the infrastructure's; 3) Flexible VM placement and move from one server to another without any physical network limitation. These characteristics will help address the issues in the data centers.

Although NVO3 may enable a true virtual environment where VMs and net service appliances communicate, the NVO3 solution has to address how to communicate between a virtual network and a physical network. This is because 1) many traditional DCs exist and will not disappear any time soon; 2) a lot of DC applications serve to Internet and/or cooperation users; 3) some applications like Big Data analytics which are CPU bound may not want the virtualization capability.

This document is to describe general NVO3 use cases that apply to various data center networks to ensure nvo3 framework and solutions can meet the demands. Three types of the use cases are:

- o A virtual network connects many tenant end systems within a Data Center and form one L2 or L3 communication domain. A virtual network segregates its traffic from others and allows the VMs in the network moving from one server to another. The case may be used for DC internal applications that constitute the DC East-West traffic.
- o A DC provider offers a secure DC service to an enterprise customer and/or Internet users. In these cases, the enterprise customer may use a traditional VPN provided by a carrier or an IPsec tunnel over Internet connecting to an overlay virtual network offered by a Data Center provider. This is mainly constitutes DC North-South traffic.
- o A DC provider uses NVO3 to design a variety of DC applications that make use of the net service appliance, virtual compute, storage, and networking. In this case, the NVO3 provides the virtual networking functions for the applications.

The document uses the architecture reference model and terminologies defined in [NVO3FRWK] to describe the use cases.

## 2. Terminology

This document uses the terminologies defined in [NVO3FRWK], [RFC4364]. Some additional terms used in the document are listed here.

CUG: Closed User Group

L2 VNI: L2 Virtual Network Instance

L3 VNI: L3 Virtual Network Instance

ARP: Address Resolution Protocol

CPE: Customer Premise Equipment

DNS: Domain Name Service

DMZ: DeMilitarized Zone

NAT: Network Address Translation

VNIF: Internal Virtual Network Interconnection Interface

## 3. Basic Virtual Networks in a Data Center

A virtual network may exist within a DC. The network enables a communication among tenant end systems (TESS) that are in a Closed User Group (CUG). A TES may be a physical server or virtual machine (VM) on a server. A virtual network has a unique virtual network identifier (may be local or global unique) for switches/routers to properly differentiate it from other virtual networks. The CUGs are formed so that proper policies can be applied when the TESSs in one CUG communicate with the TESSs in other CUGs.

Figure 1 depicts this case by using the framework model. [NVO3FRWK] NVE1 and NVE2 are two network virtual edges and each may exist on a server or ToR. Each NVE may be the member of one or more virtual networks. Each virtual network may be L2 or L3 basis. In this illustration, three virtual networks with VN context Ta, Tn, and Tm are shown. The VN 'Ta' terminates on both NVE1 and NVE2; The VN 'Tn' terminates on NVE1 and the VN 'Tm' at NVE2 only. If an NVE is a member of a VN, one or more virtual network instances (VNI) (i.e. routing and forwarding table) exist on the NVE. Each NVE has one



overlay module to perform frame encapsulation/decapsulation and tunneling initiation/termination. In this scenario, a tunnel between NVE1 and NVE2 is necessary for the virtual network Ta.

A TES attaches to a virtual network (VN) via a virtual access point (VAP) on an NVE. One TES may participate in one or more virtual networks via VAPs; one NVE may be configured with multiple VAPs for a VN. Furthermore if individual virtual networks use different address spaces, the TES participating in all of them will be configured with multiple addresses as well. A TES as a gateway is an example for this. In addition, multiple TESes may use one VAP to attach to a VN. For example, VMS are on a server and NVE is on ToR, some VMS may attach to NVE via one VLAN.

A VNI on an NVE is a routing and forwarding table that caches and/or maintains the mapping of a tenant end system and its attached NVE. The table entry may be updated by the control plane or data plane or management plane. It is possible that an NVE has more than one VNIs associated with a VN.

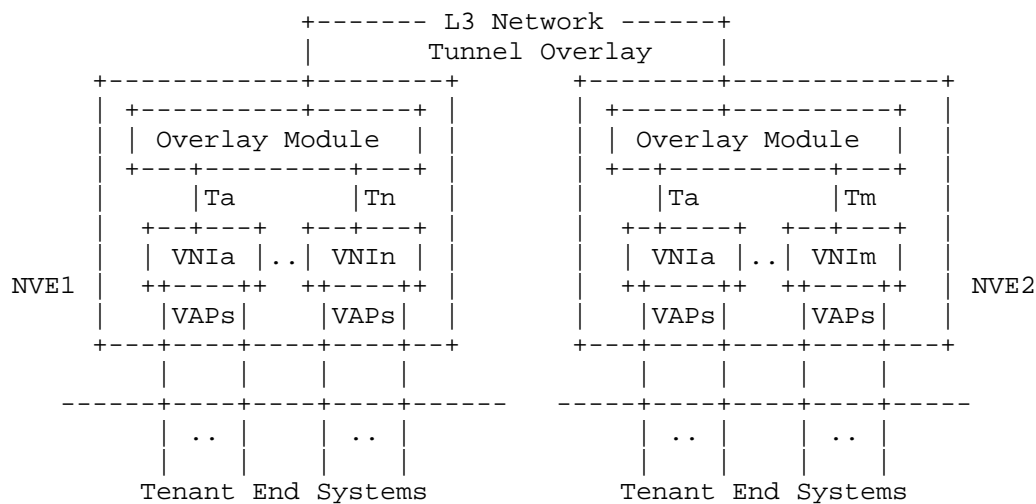


Figure 1 NV03 for Tenant End-System interconnection

One virtual network may have many NVE members and interconnect several thousands of TESs (as a matter of policy), the capability of supporting a lot of TESs per tenant instance and TES mobility is critical for NVO3 solution no matter where an NVE resides.

It is worth to mention two distinct cases here. The first is when TES and NVE are co-located on a same physical device, which means that the NVE is aware of the TES state at any time via internal API. The second is when TES and NVE are remotely connected, i.e. connected via a switched network or point-to-point link. In this case, a protocol is necessary for NVE to know TES state.

Note that if all NVEs are co-located with TESes in a CUG, the communication in the CUG is in a true virtual environment. If a TES connects to a NVE remotely, the communication from this TES to other TESes in the CUG is not in a true virtual environment. The packets to/from this TES are exposed to a physical network directly, i.e. on a wire.

Individual virtual networks may use its own address space and the space is isolated from DC infrastructure. This eliminates the route changes in the DC underlying network when VMs move. Note that the NVO3 solutions still have to address VM move in the overlay network, i.e. the TES/NVE association change when a VM moves.

If a virtual network spans across multiple DC sites, one design is to allow the corresponding NVO3 instance seamlessly span across those sites without DC gateway routers' termination. In this case, the tunnel between a pair of NVEs may in turn be tunneled over other intermediate tunnels over the Internet or other WANs, or the intra DC and inter DC tunnels are stitched together to form an end-to-end tunnel between two NVEs.

#### 4. Interconnecting DC Virtual Network and External Networks

For customers (an enterprise or individuals) who want to utilize the DC provider's compute and storage resources to run their applications, they need to access those end systems hosted in a DC through Carrier WANs or Internet. A DC provider may want to use an NVO3 virtual network to connect these end systems; then it, in turn, becomes the case of interconnecting DC virtual network and external networks. Two cases are described here.

##### 4.1. DC Virtual Network Access via Internet

A user or an enterprise customer may want to connect to a DC virtual network via Internet but securely. Figure 2 illustrates this case.

An L3 virtual network is configured on NVE1 and NVE2 and two NVEs are connected via an L3 tunnel in the Data Center. A set of tenant end systems attach to NVE1. The NVE2 connects to one (may be more) TES that runs the VN gateway and NAT applications (known as net service appliance). A user or customer can access the VN via Internet by using IPsec tunnel [RFC4301]. The encrypted tunnel is established between the VN GW and the user machine or CPE at enterprise location. The VN GW provides authentication scheme and encryption. Note that VN GW function may be performed by a net service appliance or on a DC GW.

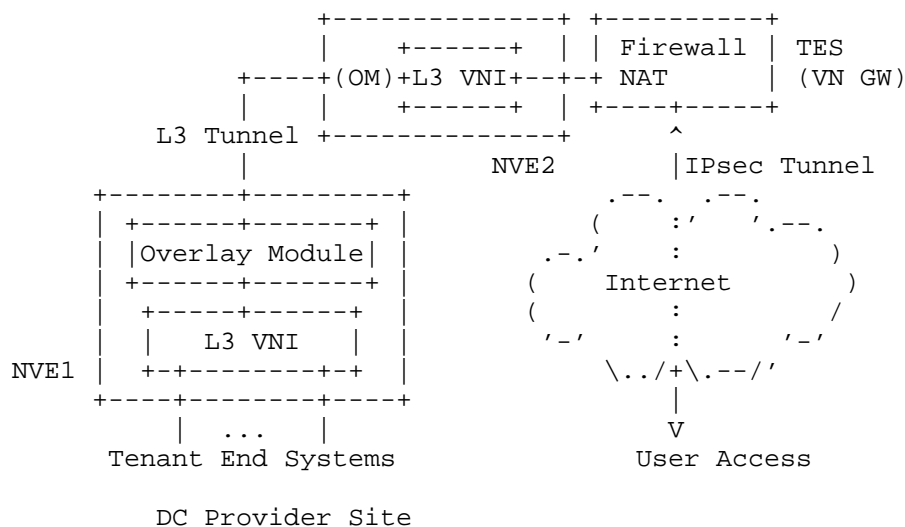


Figure 2 DC Virtual Network Access via Internet

#### 4.2. DC Virtual Network and WAN VPN Interconnection

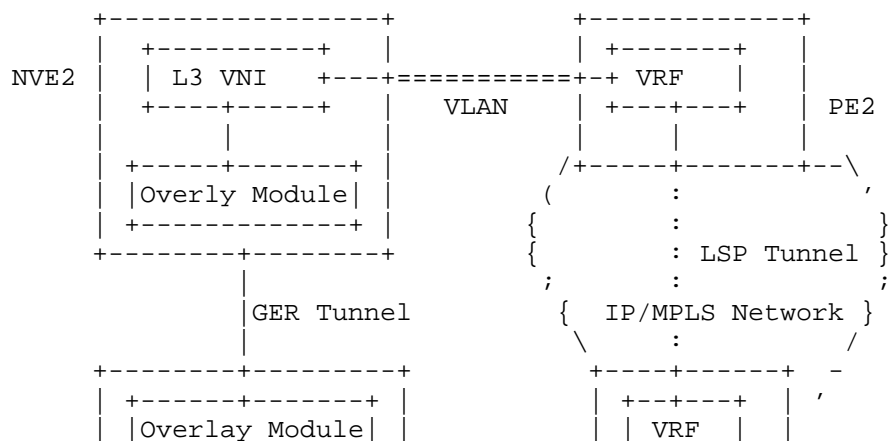
A DC Provider and Carrier may build a VN and VPN independently and interconnect the two at the DC GW and PE for an enterprise customer. Figure 3 depicts this case in a L3 overlay (L2 overlay is the same). The DC provider constructs an L3 VN between the NVE1 on a server and the NVE2 on the DC GW in the DC site; the carrier constructs an L3VPN between PE1 and PE2 in its IP/MPLS network. An Ethernet Interface physically connects the DC GW and PE2 devices. The local VLAN over the Ethernet interface [VRF-LITE] is configured to connect the L3VNI/NVE2 and VRF, which makes the interconnection between the

L3 VN in the DC and the L3VPN in IP/MPLS network. An Ethernet Interface may be used between PE1 and CE to connect the L3VPN and enterprise physical networks.

This configuration allows the enterprise networks communicating to the L3 VN as if its own networks but not communicating with DC provider underlying physical networks as well as not other overlay networks in the DC. The enterprise may use its own address space on the L3 VN. The DC provider can manage the VM and storage assignment to the L3 VN for the enterprise customer. The enterprise customer can determine and run their applications on the VMs. From the L3 VN perspective, an end point in the enterprise location appears as the end point associating to the NVE2. The NVE2 on the DC GW has to perform both the GRE tunnel termination [RFC4797] and the local VLAN termination and forward the packets in between. The DC provider and Carrier negotiate the local VLAN ID used on the Ethernet interface.

This configuration makes the L3VPN over the WANs only has the reachability to the TES in the L3 VN. It does not have the reachability of DC physical networks and other VNs in the DC. However, the L3VPN has the reachability of enterprise networks. Note that both the DC provider and enterprise may have multiple network locations connecting to the L3VPN.

The eBGP protocol can be used between DC GW and PE2 for the route population in between. In fact, this is like the Option A in [RFC4364]. This configuration can work with any NVO3 solution. The eBGP, OSPF, or other can be used between PE1 and CE for the route population.



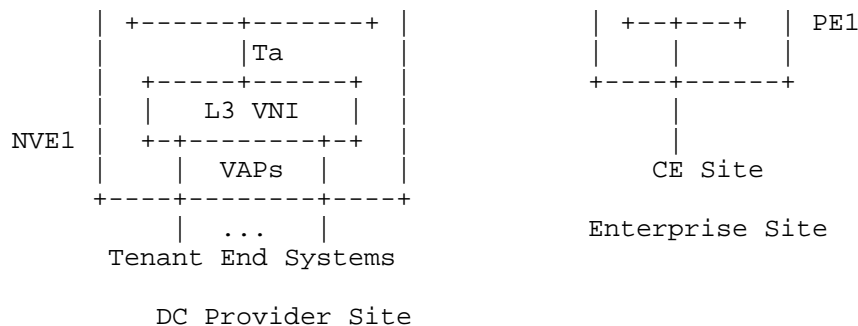


Figure 3 L3 VNI and L3VPN interconnection across multi networks

If an enterprise only has one location, it may use P2P VPWS [RFC4664] or L2TP [RFC5641] to connect one DC provider site. In this case, one edge connects to a physical network and another edge connects to an overlay network.

The interesting feature in this use case is that the L3 VN and compute resource are managed by the DC provider. The DC operator can place them at any location without notifying the enterprise and carrier because the DC physical network is completely isolated from the carrier and enterprise network. Furthermore, the DC operator may move the compute resources assigned to the enterprise from one server to another in the DC without the enterprise customer awareness, i.e. no impact on the enterprise 'live' applications running these resources. Such advanced feature brings some requirements for NVO3 [NVO3PRBM].

## 5. DC Applications Using NVO3

NVO3 brings DC operators the flexibility to design different applications in a true virtual environment without worry about physical network configuration in the Data Center. DC operators may build several virtual networks and interconnect them directly to form a tenant virtual network and implement the communication rules through policy; or may allocate some VMs to run tenant applications and some to run net service applications such as Firewall, DNS for the tenant. Several use cases are given in this section.

### 5.1. Supporting Multi Technologies in a Data Center

Most likely servers deployed in a large data center are rolled in at different times and may have different capacities/features. Some servers may be virtualized, some may not; some may be equipped with

virtual switches, some may not. For the ones equipped with hypervisor based virtual switches, some may support VxLAN [VXLAN] encapsulation, some may support NvGRE encapsulation [NVGRE], and some may not support any types of encapsulation. To construct a tenant virtual network among these servers and the ToRs, it may use two virtual networks and a gateway to allow different implementations working together. For example, one virtual network uses VxLAN encapsulation and another virtual network uses traditional VLAN.

The gateway entity, either on VMs or standalone one, participates in to both virtual networks, and maps the services and identifiers and changes the packet encapsulations.

## 5.2. Tenant Virtual Network with Bridging/Routing

A tenant virtual network may span across multiple Data Centers. DC operator may want to use L2VN within a DC and L3VN outside DCs for a tenant. This is very similar to today's DC physical network configuration. L2 bridging has the simplicity and endpoint awareness while L3 routing has advantages in aggregation and scalability. For this configuration, the virtual gateway function is necessary to interconnect L2VN and L3VN in each DC. Figure 5 illustrates this configuration.

Figure 5 depicts two DC sites. The site A constructs an L2VN that terminates on NVE1, NVE2, and GW1. An L3VN is configured between the GW1 at site A and the GW2 at site Z. An internal Virtual Network Interconnection Interface (VNIF) connects to L2VNI and L3VNI on GW1. Thus the GW1 is the members of the L2VN and L3VN. The L2VNI is the MAC/NVE mapping table and the L3VNI is IP prefix/NVE mapping table. Note that a VNI also has the mapping of TES and VAP at the local NVE. The site Z has the similar configuration. A packet coming to the GW1 from L2VN will be decapsulated and converted into an IP packet and then encapsulated and sent to the site Z. The Gateway uses ARP protocol to obtain MAC/IP mapping. Note that both the L2VN and L3VN in the figure are carried by the tunnels supported by the underlying networks which are not shown in the figure.

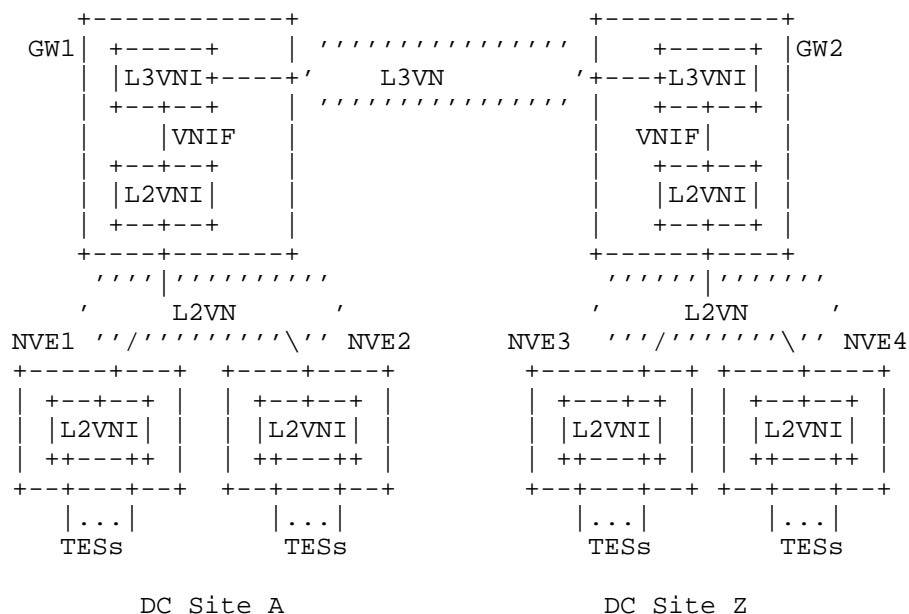


Figure 4 Tenant Virtual Network with Bridging/Routing

### 5.3. Virtual Data Center (VDC)

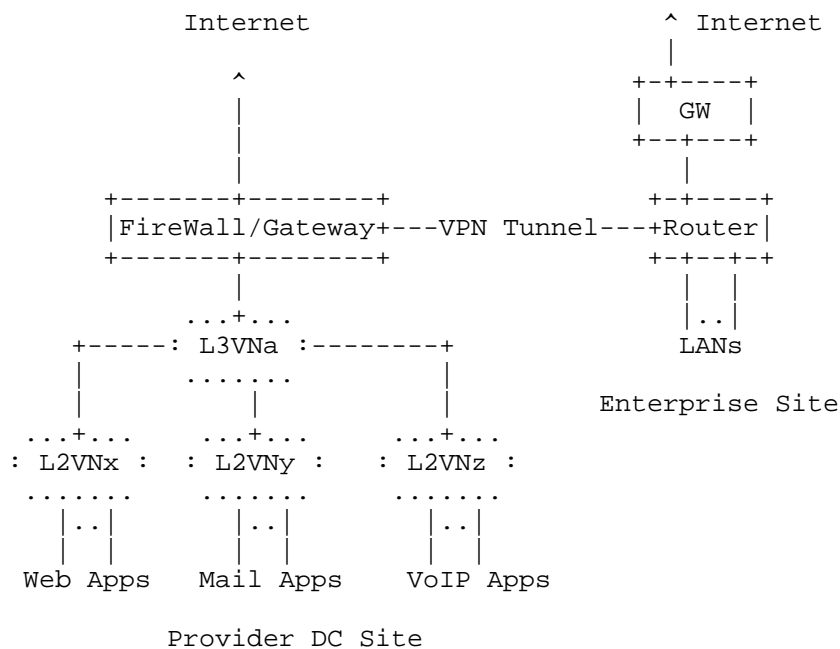
Enterprise DC's today may often use several routers, switches, and service devices to construct its internal network, DMZ, and external network access. A DC Provider may offer a virtual DC to an enterprise customer to run enterprise applications such as website/emails. Instead of using many hardware devices, with the overlay and virtualization technology of NVO3, DC operators can build them on top of a common network infrastructure for many customers and run service applications per customer basis. The service applications may include firewall, gateway, DNS, load balancer, NAT, etc.

Figure 6 below illustrates this scenario. For the simple illustration, it only shows the L3VN or L2VN as virtual and overlay routers or switches. In this case, DC operators construct several L2 VNs (L2VNX, L2VNY, L2VNZ in figure 6) to group the end tenant systems together per application basis, create an L3VNA for the internal routing. A server or VM runs firewall/gateway applications and connects to the L3VNA and Internet. A VPN tunnel is also built between the gateway and enterprise router. The design runs

Enterprise Web/Mail/VoIP applications at the provider DC site; lets the users at Enterprise site to access the applications via the VPN tunnel and Internet via a gateway at the Enterprise site; let Internet users access the applications via the gateway in the provider DC. The enterprise operators can also use the VPN tunnel or IPsec over Internet to access the vDC for the management purpose. The firewall/gateway provides application-level and packet-level gateway function and/or NAT function.

The Enterprise customer decides which applications are accessed by intranet only and which by both intranet and extranet; DC operators then design and configure the proper security policy and gateway function. DC operators may further set different QoS levels for the different applications for a customer.

This application requires the NV03 solution to provide the DC operator an easy way to create NVEs and VNIs for any design and to quickly assign TESSs to a VNI, and easily configure policies on an NVE.



\* firewall/gateway may run on a server or VMs



Figure 5 Virtual Data Center by Using NVO3

#### 5.4. Federating NVO3 Domains

Two general cases are 1) Federating AS managed by a single operator; 2) Federating AS managed by different Operators. The detail will be described in next version.

#### 6. OAM Considerations

NVO3 brings the ability for a DC provider to segregate tenant traffic. A DC provider needs to manage and maintain NVO3 instances. Similarly, the tenant needs to be informed about tunnel failures impacting tenant applications.

Various OAM and SOAM tools and procedures are defined in [IEEE 802.1ag, ITU-T Y.1731, RFC4378, RFC5880, ITU-T Y.1564] for L2 and L3 networks, and for user, including continuity check, loopback, link trace, testing, alarms such as AIS/RDI, and on-demand and periodic measurements. These procedures may apply to tenant overlay networks and tenants not only for proactive maintenance, but also to ensure support of Service Level Agreements (SLAs).

As the tunnel traverses different networks, OAM messages need to be translated at the edge of each network to ensure end-to-end OAM.

It is important that failures at lower layers which do not affect NVO3 instance are to be suppressed.

#### 7. Summary

The document describes some basic potential use cases of NVO3. The combination of these cases should give operators flexibility and power to design more sophisticated cases for various purposes.

The main differences between other overlay network technologies and NVO3 is that the client edges of the NVO3 network are individual and virtualized hosts, not network sites or LANs. NVO3 enables these virtual hosts communicating in a true virtual environment without considering physical network configuration.

NVO3 allows individual tenant virtual networks to use their own address space and isolates the space from the network infrastructure. The approach not only segregates the traffic from multi tenants on a common infrastructure but also makes VM placement and move easier.

DC applications are about providing virtual processing/storage, applications, and networking in a secured and virtualized manner, in which the NVO3 is just a portion of an application. NVO3 decouples the applications and DC network infrastructure configuration.

NVO3's underlying network provides the tunneling between NVEs so that two NVEs appear as one hop to each other. Many tunneling technologies can serve this function. The tunneling may in turn be tunneled over other intermediate tunnels over the Internet or other WANs. It is also possible that intra DC and inter DC tunnels are stitched together to form an end-to-end tunnel between two NVEs.

A DC virtual network may be accessed via an external network in a secure way. Many existing technologies can achieve this.

The key requirements for NVO3 are 1) traffic segregation; 2) supporting a large scale number of virtual networks in a common infrastructure; 3) supporting highly distributed virtual network with sparse memberships 3) VM mobility 4) auto or easy to construct a NVE and its associated TES; 5) Security 6) NVO3 Management [NVO3PRBM].

## 8. Security Considerations

Security is a concern. DC operators need to provide a tenant a secured virtual network, which means the tenant traffic isolated from other tenant's and non-tenant VMs not placed into the tenant virtual network; they also need to prevent DC underlying network from any tenant application attacking through the tenant virtual network or one tenant application attacking another tenant application via DC networks. For example, a tenant application attempts to generate a large volume of traffic to overload DC underlying network. The NVO3 solution has to address these issues.

## 9. IANA Considerations

This document does not request any action from IANA.

## 10. Acknowledgements

Authors like to thank Sue Hares, Young Lee, David Black, Pedro Marques, Mike McBride, David McDysan, and Randy Bush for the review, comments, and suggestions.

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [IEEE 802.1ag] "Virtual Bridged Local Area Networks - Amendment 5: Connectivity Fault Management", December 2007.
- [ITU-T G.8013/Y.1731] OAM Functions and Mechanisms for Ethernet based Networks, 2011.
- [ITU-T Y.1564] "Ethernet service activation test methodology", 2011.
- [RFC4378] Allan, D., Nadeau, T., "A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM)", RFC4378, February 2006
- [RFC4301] Kent, S., "Security Architecture for the Internet Protocol", rfc4301, December 2005
- [RFC4664] Andersson, L., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", rfc4664, September 2006
- [RFC4797] Rekhter, Y., etc, "Use of Provider Edge to Provider Edge (PE-PE) Generic Routing Encapsulation (GRE) or IP in BGP/MPLS IP Virtual Private Networks", RFC4797, January 2007
- [RFC5641] McGill, N., "Layer 2 Tunneling Protocol Version 3 (L2TPv3) Extended Circuit Status Values", rfc5641, April 2009.
- [RFC5880] Katz, D. and Ward, D., "Bidirectional Forwarding Detection (BFD)", rfc5880, June 2010.

### 11.2. Informative References

- [NVGRE] Sridharan, M., "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-01, July 2012

[NVO3PRBM] Narten, T., etc "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-00, September 2012

[NVO3FRWK] Lasserre, M., Motin, T., and etc, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-01, October 2012

[VRF-LITE] Cisco, "Configuring VRF-lite", <http://www.cisco.com>

[VXLAN] Mahalingam, M., Dutt, D., etc "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-02.txt, August 2012

#### Authors' Addresses

Lucy Yong  
Huawei Technologies,  
4320 Legacy Dr.  
Plano, Tx75025 US

Phone: +1-469-277-5837  
Email: [lucy.yong@huawei.com](mailto:lucy.yong@huawei.com)

Mehmet Toy  
Comcast  
1800 Bishops Gate Blvd.,  
Mount Laurel, NJ 08054

Phone : +1-856-792-2801  
E-mail : [mehmet\\_toy@cable.comcast.com](mailto:mehmet_toy@cable.comcast.com)

Aldrin Isaac  
Bloomberg  
E-mail: [aldrin.isaac@gmail.com](mailto:aldrin.isaac@gmail.com)

Vishwas Manral  
Hewlett-Packard Corp.  
191111 Pruneridge Ave.  
Cupertino, CA 95014

Phone: 408-447-1497  
Email: [vishwas.manral@hp.com](mailto:vishwas.manral@hp.com)

Linda Dunbar  
Huawei Technologies,  
4320 Legacy Dr.  
Plano, Tx75025 US

Phone: +1-469-277-5840  
Email: linda.dunbar@huawei.com



Network Working Group  
Internet Draft  
Category: Standards Track  
Expiration Date: April 2013

Y. Rekhter  
Juniper Networks

W. Henderickx  
Alcatel-Lucent

R. Shekhar  
Juniper Networks

Luyuan Fang  
Cisco Systems

Linda Dunbar  
Huawei

Ali Sajassi  
Cisco Systems

October 7 2012

## Network-related VM Mobility Issues

draft-rekhter-nvo3-vm-mobility-issues-03.txt

### 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 and License Notice

Copyright (c) 2011 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

#### Abstract

This document describes a set of network-related issues presented by the desire to support seamless Virtual Machine mobility in the data center and between data centers. In particular, it looks at the implications of meeting the requirements for "seamless mobility".



## Table of Contents

1	Specification of requirements .....	3
2	Introduction .....	3
2.1	Terminology .....	4
3	Problem Statement .....	7
3.1	Usage of VLAN-IDs .....	7
3.2	Maintaining Connectivity in the Presence of VM Mobility ...	8
3.3	Layer 2 Extension .....	8
3.4	Optimal IP Routing .....	9
3.5	Preserving Policies .....	10
4	IANA Considerations .....	10
5	Security Considerations .....	10
6	Acknowledgements .....	10
7	References .....	10
8	Author's Address .....	11

## 1. Specification of requirements

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].

## 2. Introduction

An important feature of data centers identified in [nvo3-problem] is the support of Virtual Machine (VM) mobility within the data center and between data centers. This document describes a set of network-related issues presented by the desire to support seamless Virtual Machine mobility in the data center, where seamless mobility is defined as the ability to move a VM from one server in the data center to another server in the same or different data center, while retaining the IP and MAC address of the VM. In the context of this document the term mobility, or a reference to moving a VM should be considered to imply seamless mobility, unless otherwise stated.

Note that in the scenario where a VM is moved between servers located

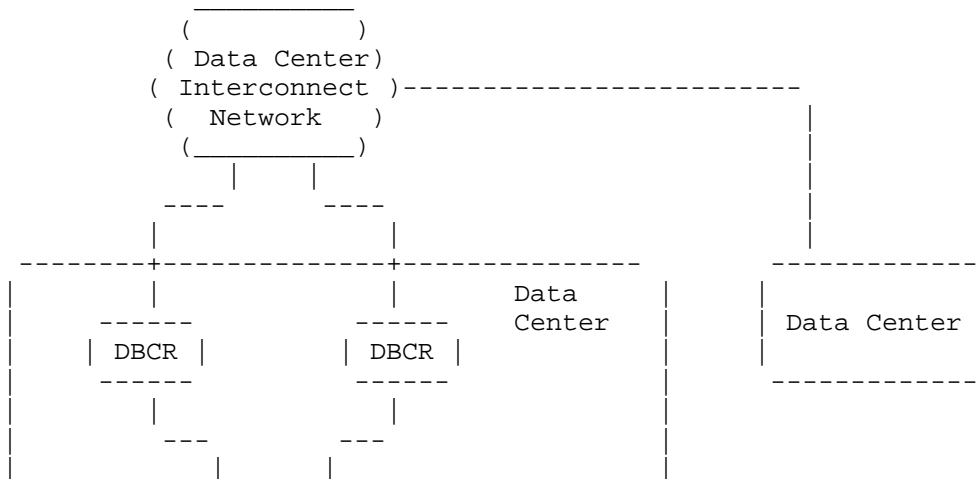
in different data centers, there are certain issues related to the current state of the art of the Virtual Machine technology, the bandwidth that may be available between the data centers, the distance between the data centers, the ability to manage and operate such VM mobility, storage-related issues (the moved VM has to have access to the same virtual disk), etc. Discussion of these issues is outside the scope of this document.

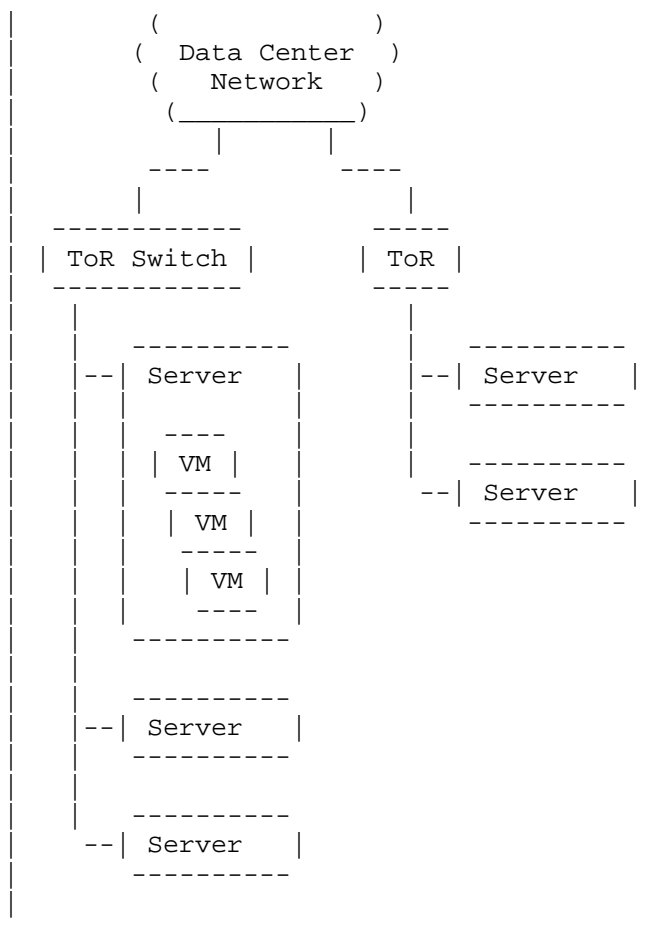
## 2.1. Terminology

In this document the term "Top of Rack Switch (ToR)" is used to refer to a switch in a data center that is connected to the servers that host VMs. A data center may have multiple ToRs. When External Bridge Port Extenders (as defined by 802.1BR) are used to connect the servers to the data center network, the ToR switch is the Controlling Bridge.

Several data centers could be connected by a network. In addition to providing interconnect among the data centers, such a network could provide connectivity between the VMs hosted in these data centers and the sites that contain hosts communicating with such VMs. Each data center has one or more Data Center Border Router (DCBR) that connects the data center to the network, and provides (a) connectivity between VMs hosted in the data center and VMs hosted in other data centers, and (b) connectivity between VMs hosted in the data center and hosts communicating with these VMs.

The following figure illustrates the above:





The data centers and the network that interconnects them may be either (a) under the same administrative control, or (b) controlled by different administrations.

Consider a set of VMs that (as a matter of policy) are allowed to communicate with each other, and a collection of devices that interconnect these VMs. If communication among any VMs in that set could be accomplished in such a way as to preserve MAC source and destination addresses in the Ethernet header of the packets exchanged among these VMs (as these packets traverse from their sources to

their destinations), we will refer to such set of VMs as an Layer 2 based Closed User Group (L2-based CUG).

A given VM may be a member of more than one L2-based CUG.

In terms of IP address assignment this document assumes that all VMs of a given L2-based CUG have their IP addresses assigned out of a single IP prefix. Thus, in the context of this document a single IP subnet corresponds to a single L2-based CUG. If a given VM is a member of more than one L2-based CUG, this VM would have multiple IP addresses and multiple logical interface, one IP address and one logical interface per each such CUG.

A VM that is a member of a given L2-based CUG may (as a matter of policy) be allowed to communicate with VMs that belong to other L2-based CUGs, or with other hosts. Such communication involves IP forwarding, and thus would result in changing MAC source and destination addresses in the Ethernet header of the packets being exchanged.

In this document the term "L2 physical domain" refers to a collection of interconnected devices that perform forwarding based on the information carried in the Ethernet header. A trivial L2 physical domain consists of just one server. In a non-trivial L2 physical domain (domain that contains multiple forwarding entities) forwarding could be provided by such layer 2 technologies as Spanning Tree Protocol (STP), etc... Note that any multi-chassis LAG can not span more than one L2 physical domain. This document assumes that a layer 2 access domain is an L2 physical domain.

A physical server connected to a given L2 physical domain may host VMs that belong to different L2-based CUGs (while each of these CUGs may span multiple L2 physical domains). If an L2 physical domain contains servers that host VMs belonging to different L2-based CUGs, then enforcing L2-based CUGs boundaries among these VMs within that domain is accomplished by relying on Layer 2 mechanisms (e.g., VLANs).

We say that an L2 physical domain contains a given VM (or that a given VM is in a given L2 physical domain), if the server presently hosting this VM is part of that domain, or the server is connected to a ToR that is part of that domain.

We say that a given L2-based CUG is present within a given data center if one or more VMs that are part of that CUG are presently hosted by the servers located in that data center.

In the context of this document when we talk about VLAN-ID used by a

given VM, we refer to the VLAN-ID carried by the traffic that is within the same L2 physical domain as the VM, and that is either originated or destined to that VM - e.g., VLAN-ID only has local significance within the L2 physical domain, unless it is stated otherwise.

### 3. Problem Statement

This section describes the specific problems/issues that need to be addressed to enable seamless VM mobility.

#### 3.1. Usage of VLAN-IDs

This document assumes that within a given non-trivial L2 physical domain traffic from/to VMs that are in that domain, and belong to the same L2-based CUG MUST have the same VLAN-ID. This document assumes that in different non-trivial L2 physical domains traffic from/to VMs that are in these domains and belong to the same L2-based CUG MAY have either the same or different VLAN-IDs. Thus when a given VM moves from one non-trivial L2 physical domain to another, the VLAN-ID of the traffic from/to VM in the former may be different than in the latter, and thus can not assume to stay the same.

This document assumes that within a trivial L2 physical domain traffic from/to VMs that are in this domain may not have VLAN-IDs at all.

If a given VM's Guest OS sends packets that carry VLAN-ID, then when the VM moves from one L2 physical domain to another the VLAN-ID used by the Guest OS can not change (this is irrespective of whether L2 physical domains are trivial or non-trivial). In other words, the VLAN-IDs used by a tagged VM network interface are part of the VM's state and cannot be changed when the VM moves from one L2 physical domain to another, even though it is possible for an entity, such as hypervisor virtual switch, to change the VLAN-ID from the value used by NVE to the value expected by the VM (in contrast, a VLAN tag assigned by a hypervisor for use with an untagged VM network interface can change). If the L2 physical domain is extended to include VM tagged interfaces, the hypervisor virtual switch, and the DC bridged network, then special consideration is needed in assignment of VLAN tags for the VMs, the L2 physical domain and other domains into which the VM may move.

This document assumes that within a given non-trivial L2 physical domain traffic from/to VMs that are in that domain, and belong to different L2-based CUG MUST have different VLAN-IDs.

The above assumptions about VLAN-IDs are driven by (a) the assumption that within a given L2 physical domain VLANs are used to identify individual L2-based CUGs, and (b) the need to overcome the limitation on the number of different VLAN-IDs.

### 3.2. Maintaining Connectivity in the Presence of VM Mobility

In the context of this document the ability to maintain connectivity in the presence of VM mobility means the ability to exchange traffic between a VM and its peer(s), as the VM moves from one server to another, where the peer(s) may be either other VM(s) or hosts. Furthermore, the peer(s) need not be within the same data center as the VM itself.

A given VM could be moved from one server to another in stopped or suspended state ("cold" VM mobility), or the hypervisors might move a running VM ("hot" VM mobility). IP address preservation is sometimes highly desired for cold VM mobility; it's mandatory to preserve transport connections when a running VM is moved.

VM mobility may result in transient loss of IP connectivity between VM and its peers. In the case of hot VM mobility the upper bound on the duration of such transients is (much) lower than in the case of cold VM mobility (due to the requirement of preserving transport connections and potential additional application requirements).

Furthermore, while with cold VM mobility one may assume that VM's ARP cache gets flushed once VM moves to another server, one can not make such an assumption with hot VM mobility.

### 3.3. Layer 2 Extension

Consider a scenario where a VM that is a member of a given L2-based CUG moves from one server to another, and these two servers are in different L2 physical domains, where these domains may be located in the same or different data centers. In order to enable communication between this VM and other VMs of that L2-based CUG, the new L2 physical domain must become interconnected with the other L2 physical domain(s) that presently contain the rest of the VMs of that CUG, and the interconnect must not violate the L2-based CUG requirement to preserve source and destination MAC addresses in the Ethernet header of the packets exchange between this VM and other members of that CUG.

Moreover, if the previous L2 physical domain no longer contains any VMs of that CUG, the previous domain no longer needs to be

interconnected with the other L2 physical domains(s) that contain the rest of the VMs of that CUG.

Note that supporting VM mobility implies that the set of L2 physical domains that contain VMs that belong to a given L2-based CUG may change over time (new domains added, old domains deleted).

We will refer to this as the "layer 2 extension problem".

Note that the layer 2 extension problem is a special case of maintaining connectivity in the presence of VM mobility, as the former restricts communicating VMs to a single/common L2-based CUG, while the latter does not.

### 3.4. Optimal IP Routing

In the context of this document optimal IP routing, or just optimal routing, in the presence of VM mobility could be partitioned into two problems:

- + Optimal routing of a VM's outbound traffic. This means that as a given VM moves from one server to another, the VM's default gateway should be in a close topological proximity to the ToR that connects the server presently hosting that VM. Note that when we talk about optimal routing of the VM's outbound traffic, we mean traffic from that VM to the destinations that are outside of the VM's L2-based CUG. This document refers to this problem as the VM default gateway problem.
- + Optimal routing of VM's inbound traffic. This means that as a given VM moves from one server to another, the (inbound) traffic originated outside of the VM's L2-based CUG, and destined to that VM be routed via the router of the VM's L2-based CUG that is in a close topological proximity to the ToR that connects the server presently hosting that VM, without first traversing some other router of that L2-based CUG (the router of the VM's L2-based CUG may be either DCBR or ToR itself). This is also known as avoiding "triangular routing". This document refers to this problem as the triangular routing problem.

Note that optimal routing is a special case of maintaining connectivity in the presence of VM mobility, as the former assumes not only the ability to maintain connectivity, but also that this connectivity is maintained using optimal routing. On the other hand, maintaining connectivity does not make optimal routing a pre-requisite.

The ability to deliver optimal routing (as defined above) in the presence of stateful devices is outside the scope of this document.

### 3.5. Preserving Policies

Moving VM from one L2 physical domain to another means (among other things) that the NVE in the new domain that provides connectivity between this VM and VMs in other L2 physical domains must be able to implement the policies that control connectivity between this VM and VMs in other L2 physical domains. In other words, the policies that control connectivity between a given VM and its peers MUST NOT change as the VM moves from one L2 physical domain to another. Moreover, policies, if any, within the L2 physical domain that contain a given VM MUST NOT preclude realization of the policies that control connectivity between this VM and its peers. All of the above is irrespective of whether the L2 physical domains are trivial or not.

### 4. IANA Considerations

This document introduces no new IANA Considerations.

### 5. Security Considerations

TBD.

### 6. Acknowledgements

The authors would like to thank Adrian Farrel for his review and comments. The authors would also like to thank Ivan Pepelnjak and David Black for their contributions to this document.

### 7. References

[nvo3-problem] Narten T.et al., "Overlays for Network Virtualization", draft-narten-nvo3-overlay-problem-statement, work in progress.



8. Author's Address

Yakov Rekhter  
Juniper Networks  
1194 North Mathilda Ave.  
Sunnyvale, CA 94089  
Email: yakov@juniper.net

Wim Henderickx  
Alcatel-Lucent  
Email: wim.henderickx@alcatel-lucent.com

Ravi Shekhar  
Juniper Networks  
1194 North Mathilda Ave.  
Sunnyvale, CA 94089  
Email: rshekhar@juniper.net

Luyuan Fang  
Cisco Systems  
111 Wood Avenue South  
Iselin, NJ 08830  
Email: lufang@cisco.com

Linda Dunbar  
Huawei Technologies  
5340 Legacy Drive, Suite 175  
Plano, TX 75024, USA  
Phone: (469) 277 5840  
Email: ldunbar@huawei.com

Ali Sajassi  
Cisco Systems  
Email: sajassi@cisco.com

Rahul Aggarwal  
Arktan, Inc  
Email: raggarwa\_1@yahoo.com



Network working group  
Internet Draft  
Category: Informational

X. Xu  
Huawei Technologies  
Kai Lee  
China Telecom

Expires: January 2013

July 9, 2012

## Path Optimization for LAN Extension

draft-xu-nvo3-lan-extension-path-optimization-00

### 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 January 9, 2013.

### Copyright Notice

Copyright (c) 2009 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

This document describes path optimization issues caused by LAN extension across geographically dispersed data centers. In addition, this document also describes requirements for possible solutions to these issues.

## 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. Problem Statement .....	3
1.1. Suboptimal Routing for Incoming Traffic .....	4
1.2. Suboptimal Routing for Outgoing Traffic .....	4
2. Terminology .....	5
3. Solution Requirements .....	5
3.1. Path Optimization for Incoming Traffic .....	5
3.2. Path Optimization for Outgoing Traffic .....	5
4. Security Considerations .....	5
5. IANA Considerations .....	6
6. Acknowledgements .....	6
7. References .....	6
7.1. Normative References .....	6
7.2. Informative References .....	6
Authors' Addresses .....	6

## 1. Problem Statement

Virtual Machine (VM) migration and geo-clustering across data centers usually require a LAN to be extended across these data centers. Figure 1 depicts a generic data center interconnect architecture where multiple data centers are interconnected with a given LAN extension solution and remote VPN sites (e.g., cloud user sites) are connected to these data centers with L3VPN solution [RFC4364].

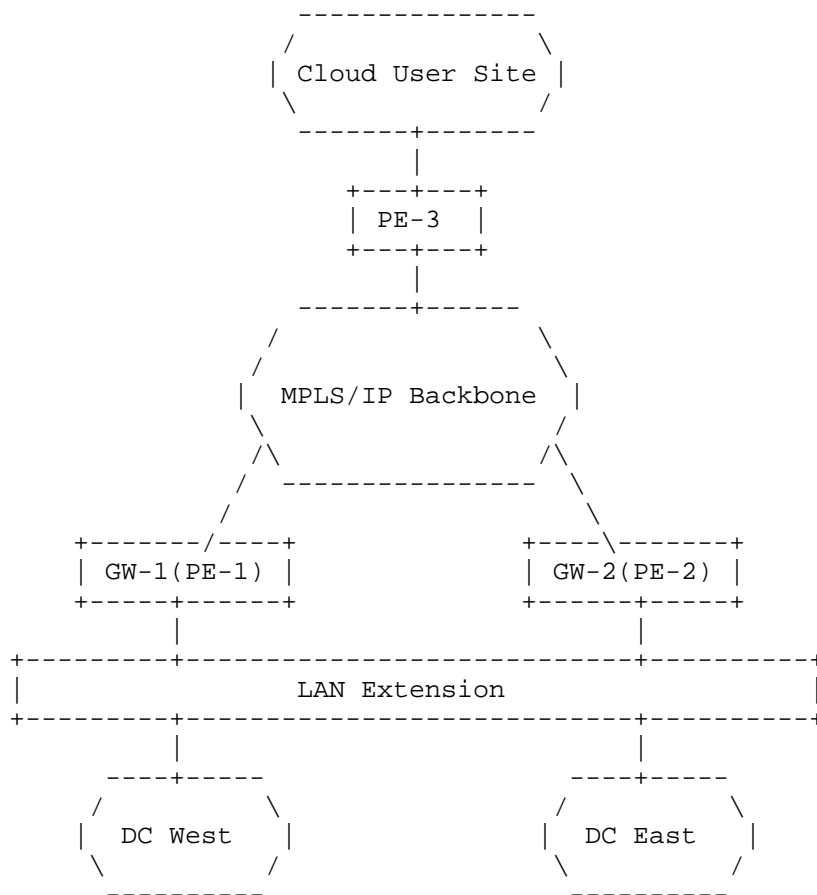


Figure 1: A Generic Data Center Interconnect Architecture

Since the LAN has been extended across multiple data center locations, the IP subnet associated with this LAN is also extended

across these locations. As such, the traffic to/from the extended subnet (e.g., the traffic between cloud user sites and data centers) would encounter suboptimal routing issues as described in the following sub-sections. Such suboptimal routing not only unnecessarily consumes the bandwidth intended for data center interconnect, but also decreases the cloud users' experiences due to increased path latency. Note that here the traffic to/from the extended subnet refers to L3VPN traffic between a remote L3VPN site (e.g., a cloud user site) and data centers, rather than Internet traffic. How to optimize the path for Internet traffic to/from the extended subnet would be explored in the future.

#### 1.1. Suboptimal Routing for Incoming Traffic

Since an IP subnet has been extended across multiple locations, the subnet no longer retains its location semantics. As a result, the incoming traffic towards a given server within the extended subnet could travel through suboptimal paths if the traffic is forwarded based on the corresponding subnet route. For example, assume a server is physically located at data center East of an extended subnet, the incoming traffic towards that server would possibly travel through the default gateway router at data center West when entering that subnet.

#### 1.2. Suboptimal Routing for Outgoing Traffic

Let's assume the existing VPLS solution [RFC4761, RFC4762] is used to achieve LAN extension across multiple data center locations. In this case, VRRP would usually be enabled on default gateway routers of different locations and only one of them would be selected as the VRRP Master for the subnet associated with the extended LAN, which is available for forwarding outgoing traffic of the subnet. In addition, although multiple default gateway routers of different locations could be selected as VRRP masters by filtering VRRP messages among them, since the existing VPLS solution however perform MAC learning as a traditional bridge, the route (e.g., MAC forwarding entry) for a given MAC address would be determined without taking the network distance into account. As a result, if the forwarding path to the VRRP virtual MAC is currently pointed to a default gateway router at data center East, for those servers located at data center West, their outgoing traffic would have to traverse the data center interconnection path so as to reach that default gateway router at data center East, which in turn forwards the traffic out of that subnet.

## 2. Terminology

This memo makes use of the terms defined in [RFC4364] and [RFC2338].

## 3. Solution Requirements

### 3.1. Path Optimization for Incoming Traffic

The basic idea is to allow each default gateway router acting as a L3VPN PE router to propagate host routes for local servers within the extended subnet to remote PE routers. More specifically, a default gateway router at a given data center is allowed to advertise hosts routes only for servers located in that data center, rather than those ones located in other data centers. In this way, remote PE routers would be able to forward traffic destined for a given server within the extended subnet according to the corresponding host route for that server, rather than the subnet route for that extended subnet.

The challenge here is how to make default gateway routers be able to tell which servers within the extended subnet are their local ones. Hence the possible solution for this path optimization issue SHOULD ensure default gateway routers to be able to obtain enough information so as to distinguish local servers from remote ones.

### 3.2. Path Optimization for Outgoing Traffic

To realize the purposes of default gateway redundancy and VM live mobility across data centers, default gateway routers of a given extended subnet at different locations SHOULD be configured with an identical virtual IP/MAC address pair (i.e., virtual router). As such, servers within the extended subnet could use that virtual router's IP address as their default gateway. To ensure the outgoing traffic with destination MAC address being the virtual router's MAC address to be forwarded to a local default gateway router, rather than any remote default gateway router, just like the anycast manner in IP networks, the LAN extension solution SHOULD be able to select the best route for a given MAC address (e.g., the virtual router's MAC address) among multiple possible routes, e.g., by taking network distance as one factor in the decision-making process of best-route selection.

## 4. Security Considerations

TBD.

## 5. IANA Considerations

There is no requirement for IANA.

## 6. Acknowledgements

TBD.

## 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

[RFC2338] Knight, S., et al., "Virtual Router Redundancy Protocol", RFC 2338, April 1998.

[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.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

## Authors' Addresses

Xiaohu Xu  
Huawei Technologies,  
Beijing, China.

Phone: +86 10 60610041  
Email: xuxiaohu@huawei.com

Kai Lee  
China Telecom,  
Beijing, China.

Leekai@ctbri.com.cn