

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].

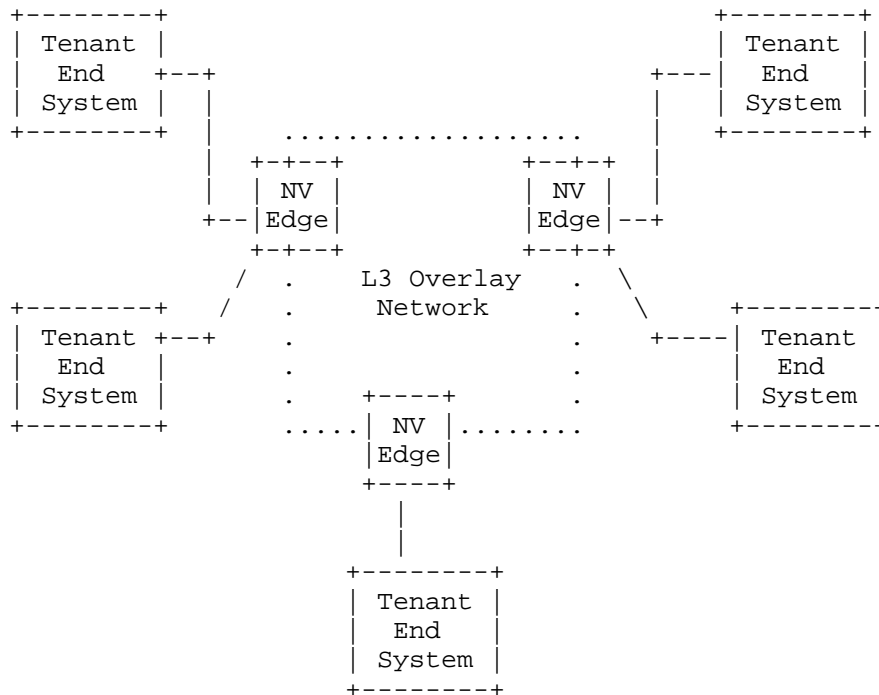


Figure 1: Generic reference model for DC network virtualization over a Layer3 infrastructure

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

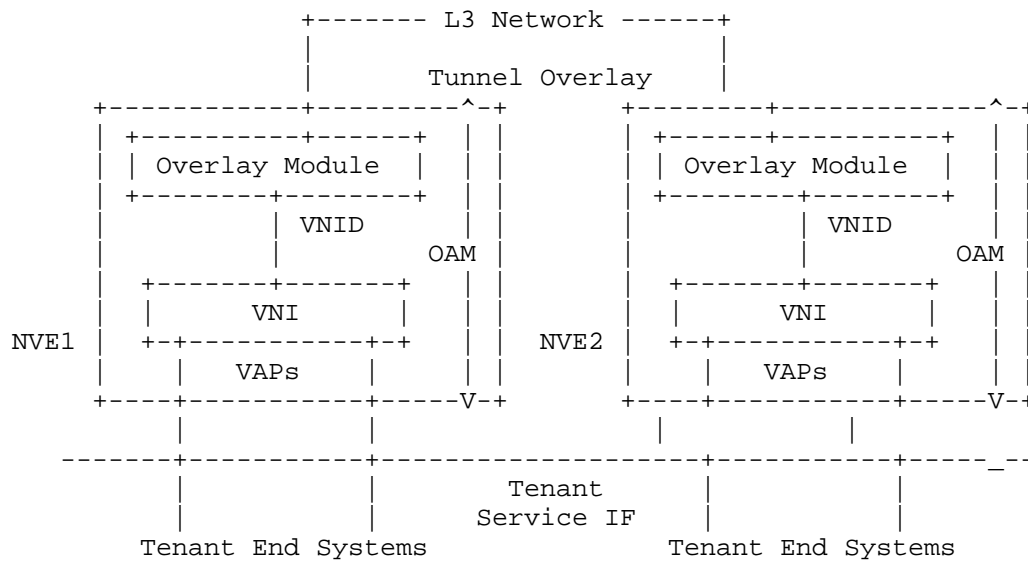


Figure 2: Generic reference model for NV Edge

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.

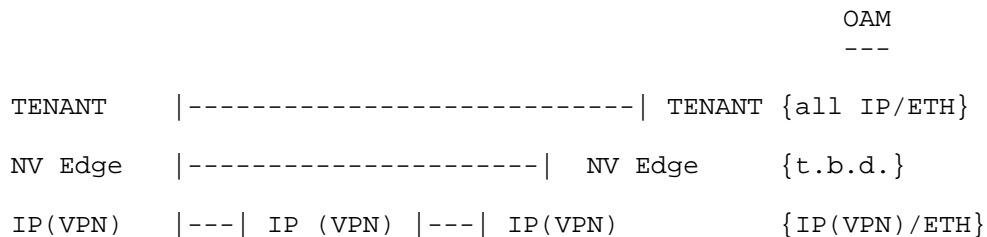


Figure 3: OAM layers in an NVO3 network

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

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Christian Jacquenet
France Telecom
Rennes 35000
France

Email: christian.jacquenet@orange.com

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

Email: ms-daikoku@kddi.com

NV03 working group
Internet Draft
Category: Informational

L. Dunbar
D. Eastlake
Huawei

Expires: April 4 2014

June 24, 2013

NV03 NVA Gap Analysis

draft-dunbar-nvo3-nva-gap-analysis-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>.

Copyright Notice

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

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

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

Abstract

This draft briefly describes the TRILL's directory assistance components. The intent of the draft is to identify the gaps against NVO3's Control Plane requirement. Through the gap analysis, the document provides a basis for future works that develop solutions for Network Virtualization Authority (NVA).

Table of Contents

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

1. Introduction

This draft briefly describes the TRILL's directory assistance components. The intent of the draft is to identify the gaps against NVO3's Control Plane requirement. Through the gap analysis, the document provides a basis for future works that develop solutions for Network Virtualization Authority (NVA).

Section 4.5 of [nvo3-problem-statement] describes the back-end Network Virtualization Authority (NVA) that is responsible for distributing the mapping information for entire overlay system.

There are some similarities between TRILL [RFC6325] and NVO3, e.g. TRILL using TRILL header to achieve overlay, vs. NV03 using L3 headers plus VNID to achieve overly. This draft analyzes the TRILL directory assistance components that are applicable to NVO3's NVA and summarize the gaps.

2. Terminology

The following terms are used interchangeably in this document:

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

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

DA: Destination Address

DC: Data Center

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

End Station: Guest OS running on a physical server or on a virtual machine. An end station in this document has at least one IP address and at least one MAC address, which could be in DA or SA field of a data frame.

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

NVA: Network Virtualization Authority

NVE: Network Virtualization Edge

SA: Source Address

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

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

TRILL: Transparent Interconnection of Lots of Links [RFC6325]

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

TS: Tenant System

VM: Virtual Machines

VN: Virtual Network

VNID: Virtual Network Instance Identifier

3. Overall Requirement for NVA

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

Section 3.1 of [nvo3-cp-req] states that a protocol is needed to provide this inner to outer mapping and VN Context to each NVE that requires it and keep the mapping updated in a timely manner. Timely updates are important for maintaining connectivity between Tenant Systems.

TRILL has two ways to achieve the inner<->outer address mapping: one is via data plane flooding and learning; the other way is via directory assistance [TRILL-Directory]. One goal of NVO3 using a control protocol is to eliminate this flooding. Therefore it is worthwhile to examine the TRILL's directory assistance components and analyze the gaps.

4. Existing Directory Components

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

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

4.1. Types of NVA:

NVA can be centralized, even though there could be multiple entities for redundancy purpose, or distributed with each NVA holding the mapping information for a subset of VNs. NVA could be standalone servers/VMs (i.e. end systems), or could be integrated with some NVEs. When NVA is a standalone server/VM, it has to be reachable by NVEs through the underlay network.

4.2. Key components of the information kept in the NVA

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

1. TS (host) IP Address;
2. TS (host) L2 Address, i.e. MAC;
3. TS Local L2 VLAN ID list (one local VLAN can be represented by multiple VIDs);
4. The list of locally attached edges (NVEs); normally one TS is attached to one edge, TS could also be attached to 2 edges for redundancy (dual homing). One TS is rarely attached to more than 2 edges, though it could be possible;

5. Timer for NVEs to keep the entry when pushed down to or pulled from NVEs.
6. Optionally the list of interested remote edges (NVEs). This information is for NVA to promptly update relevant edges (NVEs) when there is any change to this TS' attachment to edges (NVEs). However, this information doesn't have to be kept per TS. It can be kept per VN.

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

4.3. Mapping Entries Distribution Mechanism

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

4.3.1. Push Mode

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

- simply drop the data frame,
- flood it to all other edges that are in the same VN,
or
- start the ''pull'' process to get information from Pull Directory Server(s) (or NVA)

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

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

down all the entries for all the VNs that are enabled on the NVE.

4.3.1.1. Requesting Push Service

In the Push Mode, it is necessary to have a way for an edge node (NVE) to request directory server(s) (NVA) to start the pushing process, e.g. when the NVE is initialized or re-started.

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

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

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

4.3.1.2. Incremental Push Service

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

If the Push Directory server is configured to believe it has complete mapping information for VN X then, after it has actually transmitted all of its ESADI-LSPs for X it waits its CSNP time (see Section 6.1 of [ESADI]), and then updates its

ESADI-Parameters APPsub-TLV to set the Complete Push (CP) bit to one. It then maintains the CP bit as one as long as it is Active.

4.3.2. Pull Mode

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

One advantage of the Pull Mode is that edge nodes (NVEs) can age out mapping entries if they haven't been used for a certain period of time. Therefore, each edge (NVE) will only keep the entries that are frequently used, so its mapping table size will be smaller than a complete table pushed down from NVA.

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

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

4.3.2.1. Pull Requests

Here are some events that can trigger the pulling process:

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

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

4.3.2.2. Pull Response

There are several possibilities of the Pull Response:

1. Valid inner-outer address mapping, coupled with the valid timer indicating how long the entry can be cached by the edge (NVE).
The timer for cache should be short in an environment where VMs move frequently. The cache timer can also be configured.
2. The target being queried is not available or
3. The requestor is administratively prohibited from getting an informative response.

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

4.3.2.3. Cache Consistency

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

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

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

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

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

3. For each unit of data held by the server and each address about which a negative response was sent, a list of RBridges that were sent that unit as the response or sent a negative response to the address, with the expected time to expiration at each of them.

4.3.2.4. Pull Request Errors

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

4.3.2.5. Redundant Pull Directories (NVAs)

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

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

4.3.3. Hybrid Mode

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

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

In addition, an NVE can also be configured to flood and learn via data plane if target doesn't exist in the pushed mapping entries and no response is received from Pull mode. However, if the response from Pull Directory indicates that the NVE is administratively prohibited from forwarding data frame to the requested target, the NVE should drop the data frame.

5. Redundancy

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

6. Inconsistency Processing

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

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

information and if information from Push Directory servers conflicts, the information from the higher priority Push Directory server is preferred.

7. Gap Summary

7.1. Features necessary to NVO3 but not present in TRILL

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

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

7.2. Additional detailed requirement applicable to NVO3's NVA

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

- Push Mode:

- o For redundancy purposes, for each VN there should be multiple NVA entities holding the mapping information for the TSs in the VN. At any given time, only one or a small number of the NVAs are considered as Active for a particular VN. All NVAs should announce their capability and priority to all the edges.
- o If the destination of a data frame arriving at the Ingress Edge (NVE) can't be found in its inner-outer mapping table that are pushed down from the Directory Server(s) (NVA), the Ingress edge could be configured to:
 - simply drop the data frame,
 - flood it to all other edges that are in the same VN,
 - or
 - start the ''pull'' process to get information from Pull Directory Server(s) (or NVA)
- o If an NVE lost its connection to its NVA, it MUST ignore any Push Directory data from that server because it is no longer being updated and may be stale.
- o When transient conflict occurs: higher priority data take precedence.

- Pull Mode:
 - o The Pull Directory response could indicate that the address being queried is not available in NVA or that the requestor is administratively prohibited from getting an informative response.
 - o The timer for ingress NVE caching should be short in an environment where VMs move frequently. The cache timer could be configured or could be sent along with the Pulled reply from the NVA.
 - o Each Pull request can have multiple queries for different TSs.
 - o It is highly desirable to have a mechanism to prevent NVEs from using the stale mapping entries pulled from NVA.
 - o While waiting for query response from NVA, the NVE has to buffer the subsequent data frames with destination address to the same target. The buffer could overflow before the NVE gets the response from NVA.
 - o If no response is received to a Pull Directory request within a configurable timeout, the request should be re-transmitted with the same Sequence Number up to a configurable number of times.
- Hybrid Mode:
 - o NVE can be configured to get some VN's mapping entries via push mode and other VN's mapping entries via pull mode.

8. Security Considerations

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

For general TRILL security considerations, see [RFC6325].

9. IANA Considerations

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

10. Acknowledgements

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

11. References

11.1. Normative References

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

11.2. Informative References

[802.1Q] IEEE Std 802.1Q-2011, "IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks", May 2011.

[802.1Qbg] IEEE Std 802.1Qbg-2012, ''Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks ---Edge Virtual Bridging'', July 2012.

[RFC826] Plummer, D., "An Ethernet Address Resolution Protocol", RFC 826, November 1982.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC6325] Perlman, et, al ''RBridge: Base Protocol Specification'', <https://datatracker.ietf.org/doc/rfc6325/>, July, 2011

[RFC6439] Perlman, et, al ''RBridges: Appointed Forwarders'', <https://datatracker.ietf.org/doc/rfc6439/>, Nov 2011

Authors' Addresses

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

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

NV03 working group
Internet Draft
Category: Informational

L. Dunbar
D. Eastlake
Huawei

Expires: April 4 2014

September 20, 2013

NV03 NVA Gap Analysis

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

Status of this Memo

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

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

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

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

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

Copyright Notice

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

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

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

Abstract

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

Table of Contents

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

1. Introduction

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

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

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

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

2. Terminology

The following terms are used interchangeably in this document:

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

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

DA: Destination Address

DC: Data Center

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

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

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

LISP: Locator/ID Separation Protocol

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

NVA: Network Virtualization Authority

NVE: Network Virtualization Edge

SA: Source Address

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

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

TRILL: Transparent Interconnection of Lots of Links [RFC6325]

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

TS: Tenant System

VM: Virtual Machines

VN: Virtual Network

VNID: Virtual Network Instance Identifier

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

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

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

Timely updates are important for maintaining connectivity between Tenant Systems.

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

4. Existing Directory Components

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

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

4.1. Types of NVA:

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

4.2. Key components of the information kept in the NVA

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

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

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

4.3. Mapping Entries Distribution Mechanism

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

4.3.1. Push Mode

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

- simply drop the data frame,

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

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

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

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

4.3.1.1. Requesting Push Service

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

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

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

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

4.3.1.2. Incremental Push Service

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

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

4.3.2. Pull Mode

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

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

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

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

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

4.3.2.1. Pull Requests

Here are some events that can trigger the pulling process:

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

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

4.3.2.2. Pull Response

There are several possibilities of the Pull Response:

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

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

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

4.3.2.3. Cache Consistency

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

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

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

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

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

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

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

4.3.2.4. Pull Request Errors

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

4.3.2.5. Redundant Pull Directories (NVAs)

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

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

4.3.3. Hybrid Mode

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

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

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

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

5. Redundancy

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

6. Inconsistency Processing

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

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

7. Gap Summary

7.1. Features necessary to NVO3 but not present in TRILL

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

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

7.2. Additional detailed requirement applicable to NVO3's NVA

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

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

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

8. Security Considerations

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

For general TRILL security considerations, see [RFC6325].

9. IANA Considerations

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

10. Acknowledgements

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

11. References

11.1. Normative References

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

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

11.2. Informative References

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

Authors' Addresses

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

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

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

E. Gray, Ed.
Ericsson
N. Bitar
Verizon
X. Chen
Huawei Technologies
M. Lasserre
Alcatel-Lucent
T. Tsou
Huawei Technologies (USA)
July 15, 2013

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

Abstract

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

Status of This Memo

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

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

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

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

1. Introduction

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

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

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

The major areas covered in this document include:

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

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

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

The initial candidate technologies are:

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

2. Terminology and Conventions

2.1. Requirements Language

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

2.2. Conventions

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

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

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

2.3. Terms and Abbreviations

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

ECN: Explicit Congestion Notification [RFC3168]

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

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

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

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

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

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

This document uses the following additional general terms and abbreviations:

DSCP: Differentiated Services Code-Point

ECMP: Equal Cost Multi-Path

L2VPN: Layer 2 Virtual Private Network

L3VPN: Layer 3 Virtual Private Network

NVO3: Network Virtualization Overlay over L3

VM: Virtual Machine

VN: Virtual Network

3. Operational Requirements

TBD

4. Management Requirements

TBD

5. Control Plane Requirements

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

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

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

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

Current control-plane requirement documents include the following:

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

5.1. Overall Control-Plane Requirements

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

(3.1) Inner to Outer Address Mapping

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

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

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

Table 1: Inner:Outer Address Mapping

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

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

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

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

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

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

Table 2: Multi-Destination Delivery

(3.3) VN Connect/Disconnect Notification

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

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

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
Connect Notification					

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

Table 3: Connect/Disconnect Notification

(3.4) VN Name to VNID Mapping

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

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

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

Table 4: VN Name to VN ID Mapping

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

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

(4.1) VN Connect/Disconnect

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

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

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

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

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

Table 5: VN Connect/Disconnect

(4.2) VNIC Address Association

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

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

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

Table 6: VNIC Address Association

(4.3) VNIC Address Disassociation

TBD

(4.4) VNIC Shutdown/Startup/Migration

TBD

(4.5) VN Profile

TBD

6. Data Plane Requirements

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

(3.1) Virtual Access Points (VAPs)

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

Table 7: VAP Identification Requirements

(3.2) Virtual Network Instance (VNI)

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

Table 8: VAP-VNI Association

(3.2.1) L2 VNI

Requirement	NVGRE	VxLAN	VPLS	EVPN	L3VPN
L2 VNI MUST provide an emulated Ethernet multipoint service as if Tenant Systems are interconnected by a bridge (but instead by using a set of NVO3 tunnels).					

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

Table 9: L2 VNI Service

(3.2.2) L3 VNI

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

Table 10: L3 VNI Service

(3.3.1) NVO3 overlay header

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

Table 11: Overlay Header

(3.3.1.1) Virtual Network Context Identification

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

Table 12: Virtual Network Context Identification

(3.3.1.2) Service QoS identifier

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

Table 13: QoS Service Identification

(3.3.2.1) LAG and ECMP

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

Table 14: Multipath Support

(3.3.2.2) DiffServ and ECN marking

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

Table 15: DSCP and ECN Marking

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

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

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

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

(3.4) External NVO3 connectivity

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

Table 17: Interoperation

(3.5) Path MTU

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

Table 18: Path MTU

(3.7) NVE Multi-Homing Requirements

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

Table 19: Multihoming

(3.8) OAM

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

Table 20: OAM Messaging

7. Summary and Conclusions

TBD

8. Acknowledgements

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

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

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

11. References

11.1. Normative References

- [I-D.ashwood-nvo3-operational-requirement]
Ashwood-Smith, P., Iyengar, R., Tsou, T., Sajassi, A., Boucadair, M., Jacquenet, C., and M. Daikoku, "NVO3 Operational Requirements", draft-ashwood-nvo3-operational-requirement-02 (work in progress), January 2013.
- [I-D.ietf-l2vpn-evpn]
Sajassi, A., Aggarwal, R., Henderickx, W., Balus, F., Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-03 (work in progress), February 2013.
- [I-D.ietf-nvo3-dataplane-requirements]
Bitar, N., Lasserre, M., Balus, F., Morin, T., Jin, L., and B. Khasnabish, "NVO3 Data Plane Requirements", draft-ietf-nvo3-dataplane-requirements-01 (work in progress), July 2013.
- [I-D.ietf-nvo3-framework]
Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-03 (work in progress), July 2013.
- [I-D.ietf-nvo3-overlay-problem-statement]
Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-03 (work in progress), May 2013.
- [I-D.kreeger-nvo3-hypervisor-nve-cp]
Kreeger, L., Narten, T., and D. Black, "Network Virtualization Hypervisor-to-NVE Overlay Control Protocol Requirements", draft-kreeger-nvo3-hypervisor-nve-cp-01 (work in progress), February 2013.

- [I-D.kreeger-nvo3-overlay-cp]
Kreeger, L., Dutt, D., Narten, T., Black, D., and M. Sridharan, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp-04 (work in progress), June 2013.
- [I-D.mahalingam-dutt-dcops-vxlan]
Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-04 (work in progress), May 2013.
- [I-D.sridharan-virtualization-nvgre]
Sridharan, M., Greenberg, A., 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.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.
- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.
- [RFC4365] Rosen, E., "Applicability Statement for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4365, February 2006.
- [RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.
- [RFC4762] Lasserre, M. and V. Kompella, "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, January 2007.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.
- [RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, November 2010.

[RFC6074] Rosen, E., Davie, B., Radoaca, V., and W. Luo,
"Provisioning, Auto-Discovery, and Signaling in Layer 2
Virtual Private Networks (L2VPNs)", RFC 6074, January
2011.

11.2. Informative References

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

Authors' Addresses

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

Email: eric.gray@ericsson.com

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, Massachusetts 02145
USA

Email: nabil.bitar@verizon.com

Xiaoming Chen
Huawei Technologies

Email: ming.chen@huawei.com

Marc Lasserre
Alcatel-Lucent

Email: marc.lasserre@alcatel-lucent.com

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

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

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 16, 2014

S. Hartman
Painless Security
D. Zhang
Huawei
M. Wasserman
Painless Security
July 15, 2013

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

Abstract

This draft discusses the security requirements and several issues which need to be considered in securing a virtualized data center network for multiple tenants (a NVO3 network for short). In addition, the draft also attempts to discuss how such issues could be addressed or mitigated.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 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	2
2. Terminology	3
3. NVO3 Overlay Architecture	4
4. Threat Model	4
4.1. Outsider Capabilities	5
4.2. Insider Capabilities	5
4.3. Security Properties	6
5. Basic Security Approaches	7
5.1. Securing the Communications between NVEs and TSes	7
5.2. Securing the Communications within Overlays	8
5.2.1. Control Plane Security	8
5.2.2. Data Plan Security	10
6. Security Issues Imposed by the New Overlay Design Characteristics	11
6.1. Scalability Issues	11
6.2. Influence on Security Devices	11
6.3. Security Issues with VM Migration	11
7. IANA Considerations	12
8. Security Considerations	12
9. Acknowledgements	12
10. References	12
10.1. Normative References	12
10.2. Informative References	12
Authors' Addresses	13

1. Introduction

Security is the key issue which needs to be considered in the design of a data center network. This document first lists the security risks that a NVO3 network may encounter and the security requirements that a NVO3 network need to fulfill. Then, this draft discusses the

essential security approaches which could be applied to fulfill such requirements.

The remainder of this document is organized as follows. (Section 4) introduces the attack model of this work and the properties that a NOV3 security mechanism needs to enforce. Section 5 describes the essential security mechanisms which should be provide in the generation of a NVO3 network. Then, in Section 6, we analyze the challenges brought by the new features mentioned in[I-D.ietf-nvo3-overlay-problem-statement].

2. Terminology

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

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

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

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

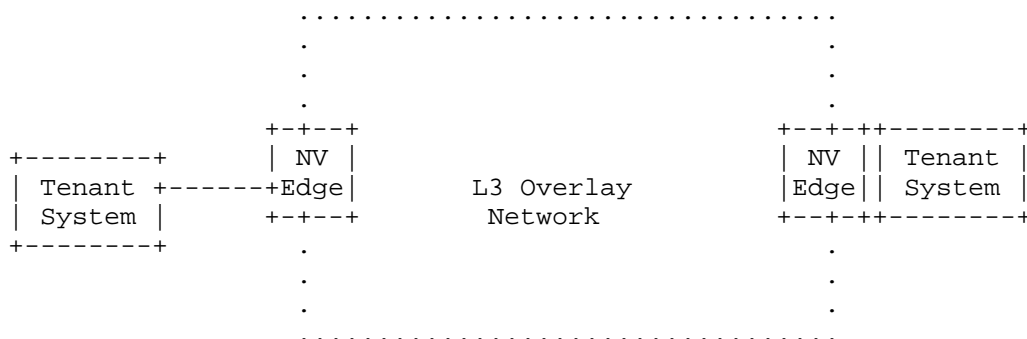
Virtual Network (VN): This is a virtual L2 or L3 domain that belongs to a tenant.

Information Mapping Authority (IMA). 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. In [I-D.ietf-nvo3-overlay-problem-statement], such a back-end system is referred to as a "oracle".

3. NV03 Overlay Architecture

Please view in a fixed-width font such as Courier.

Please view in a fixed-width font such as Courier.



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

4. Threat Model

To benefit the discussion, in this analysis work, attacks are classified into two categories: inside attacks and outside attacks. An attack is considered as an inside attack if the adversary performing the attack (inside attacker or insider) has got certain privileges in changing the configuration or software of a NV03 device (or a network devices of the underlying network where the overlay is located upon) and initiates the attack within the overlay security perimeter. In contrast, an attack is referred to as an outside attack if the adversary performing the attack (outside attacker or outsider) has no such privilege and can only initiate the attacks from compromised TSes. Note that in a complex attack inside and outside attacking operations may be performed in a well organized way to expand the damages caused by the attack.

This analysis assumes that security protocols, algorithms, and implementations provide the security properties for which they are designed; attacks depending on a failure of this assumption are out of scope. As an example, an attack caused by a weakness in a cryptographic algorithm is out of scope, while an attack caused by failure to use confidentiality when confidentiality is a security requirement is in scope.

4.1. Outsider Capabilities

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

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

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

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

4.2. Insider Capabilities

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

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

Note that in practice an insider controlling an underlying network device may break the communication of the overlays by discarding or delaying the delivery of the packets passing through it. However, this type of attack is out of scope.

4.3. Security Properties

When encountering an attack, a virtual data center network **MUST** guarantee the following security properties:

1. Isolation of the VNs: In [I-D.ietf-nvo3-overlay-problem-statement], the data plane isolation requirement amongst different VNs has been discussed. The traffic within a virtual network can only be transited into another one in a controlled fashion (e.g., via a configured router and/or a security gateway). In addition, it **MUST** be ensured that an entity cannot make use of its privilege obtained within a VN to manipulate the overlay control plane to affect on the operations of other VNs.
2. Spoofing detection: Under the attacks performed by a privileged inside attacker, the attacker cannot use the obtained cryptographic materials to impersonate another one.
3. Integrity protection and message origin authentication for the control packets: The implementation of an overlay control plane **MUST** support the integrity protection on the signaling packets. No entity can modify a overlay signaling packet during its transportation without being detected. Also, an attacker cannot impersonate a legal victim (e.g., a NVE or another servers within the overlay) to send signaling packets without detection.
4. Availability of the control plane: The design of the control plan must consider the DoS/DDoS attacks. Especially when there are centralized servers in the control plan of the overlay, the servers need to be well protected and make sure that they will not become the bottle neck of the control plane especially under DDOS attacks.

The following properties **SHOULD** be optionally provided:

1. Confidentiality and integrity of the data traffic of TSes. In some conditions, the cryptographic protection on the TS traffic is not necessary. For example, if most of the ES data is headed towards the Internet and nothing is confidential, encryption or integrity protection on such data may be unnecessary. In addition, in the cases where the underlay network is secure enough, no additional cryptographic protection needs to be provided.
2. Confidentiality of the control plane. On many occasions, the signaling messages can be transported in plaintext. However, when the information contained within the signaling messages are sensitive or valuable to attackers (e.g., the location of a ES, when a VM migration happens), the signaling messages related with that tenant SHOULD be encrypted.

5. Basic Security Approaches

This section introduces the security mechanisms which could be used to provided in order to guarantee the security properties mentioned in section 4 when encountering attacks.

5.1. Securing the Communications between NVEs and TSes

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

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

In the former case, the data and control traffic between the NVE and the TSes are exchanged over network. If the NVE supports multiple VNs concurrently, the data/control traffics in different VNs MUST be isolated physically or by using VPN technologies. If the network connecting the NVE and the TSes is potentially accessible to attackers, the security properties of data traffic (e.g., integrity, confidentiality, and message origin authenticity) SHOULD be provided. The security mechanisms such as IPsec, SSL, and TCP-AO, can be used according to different security requirements.

In order to guarantee the integrity and the origin authentication of signaling messages, integrated security mechanisms or additional security protocols need to be provided. In order to secure the data/

control traffic, cryptographic keys need to be distributed to generate digests or signatures for the control packets. Such cryptographic keys can be manually deployed in advance or dynamically generated with certain automatic key management protocols (e.g., TLS [RFC5246]). The TSes belonging to different VNs MUST use different keys to secure the control packets exchanges with their NVE. Therefore, an attacker cannot use the keys it obtained from a compromised TS to generate bogus signaling messages and inject them into other VNs without being detected. For a better damage confinement capability, different TSes SHOULD use different keys to secure their control packet exchanges with NVEs, even if they belong to the same VN.

In the co-located case, all the information exchanges between the NVE and the TSes are within the same device, and no standardized protocol need to be provided for transporting control/data packets. It is also important to keep the isolation of the TS traffic in different VNs. In addition, in the co-location fashion, because the NVE, the hypervisor, and the VMs are deployed on the same device, the computing and memory resources used by the NVE, the hypervisor, and the TSes need to be isolated to prevent a malicious or compromised TS from, e.g., accessing the memory of the NVE or affecting the performance of the NVE by occupying large amounts of computing resources.

5.2. Securing the Communications within Overlays

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

5.2.1. Control Plane Security

It is the responsibility of the NVO3 network to protect the control plane packets transported over the underlay network against the attacks from the underlying network. The integrity and origin authentication of the messages MUST be guaranteed. The signaling packets SHOULD be encrypted when the signaling messages are confidential. To achieve such objectives, when the network devices exchange control plane packets, integrated security mechanisms or security protocols need to be provided. In addition, cryptographic keys need to be deployed manually in advance or dynamically generated by using certain automatic key management protocols (e.g., TLS [RFC5246]).

In order to enforce the security boundary of different VNs in the existence of inside adversaries, the signaling messages belonging to different VNs need to be secured by different keys. Otherwise, an inside attacker may try to use the keys obtained within a VN to

impersonate the NVEs in other VNs and generate illegal signaling messages without being detected. If we expect to provide a better attack confinement capability and prevent a compromised NVE to impersonate other NVEs in the same VN, different NVEs working inside a VN need to secure their signaling messages with different keys. When there are centralized servers providing mapping information (IMAs) within the overlay, it will be important to prevent a compromised NVE from impersonating the centralized servers to communicate with other NVEs. A straightforward solution is to associate different NVEs with different keys when they exchange information with the centralized servers.

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

When an automated key management solution for NVO3 overlays is deployed, as a part of the key management protocol, mutual authentication needs to be performed before two network devices in the overlay (NVEs or IMAs) start exchanging the control packets. After an authentication, an device can find out whether its peer holds valid security credentials is is the one who it has claimed. The authentication results is also necessary for authorization; it is important for a device to clarify the roles (e.g., a NVE or a IMA) that its authentication peer acts as in the overlay. Therefore, a compromised NVE cannot use it credential to impersonate an IMA to communicate with other NVEs. Only the control messages from the authenticated entity will be adopted. In addition, authorization MAY need to be performed. For instance, before accepting a control message, the receiver NVE needs to verify whether the message comes from one which is authorized to send that message. If the

authorization fail, the control message will be discarded. For instance, if a control packet about a VN is sent from a NVE which is not authorized to support the VN, the packet will be discarded.

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

In addition, during the design of the control plane, it is important to consider the amplification effects which may potential be used by attackers to carry out reflection attacks.

5.2.2. Data Plan Security

[I-D.ietf-nvo3-framework] specifies a NVO3 overlay needs to generate tunnels between NVEs for data transportation. When a data packet reaches the boundary of a overlay, it will be encapsulated and forwarded to the destination NVE through a proper tunnel. It is normally assume that the underlying network connecting NVEs are secure to outside attacks since it is under the management of DC vendor and cannot be directly accessed by tenants. However, when facing inside attacks, conditions could be complex. For instance, an inside attacker compromising a underlying network device may intercept an encapsulated data packet transported a tunnel, modify the contents in the encapsulating tunnel packet and, transfer it into another tunnel without being detected. When the modified packet reaches a NVE, the NVE may decapsulated the data packet and forward it into a VN according to the information within the encapsulating header generated by the attacker. Similarly, a compromised NVE may try to redirect the data packets within a VN into another VN by adding improper encapsulating tunnel headers to the data packets. Under such circumstances, in order to enforce the VN isolation property, signatures or digests need to be generated for both data packets and the encapsulating tunnel headers in order to provide data origin authentication and integrity protection. In addition, NVEs SHOULD use different keys to secure the packets transported in different tunnels.

6. Security Issues Imposed by the New Overlay Design Characteristics

6.1. Scalability Issues

NOV3 WG requires an overlay be able to work in an environment where there are many thousands of NVEs (e.g. residing within the hypervisors) and large amounts of trust domains (VNs). Therefore, the scalability issues should be considered. In the cases where a NVE only has a limited number of NVEs to communicate with, the scalability problem brought by the overhead of generating and maintaining the security channels with the remote NVEs is not serious. However, if a NVE needs to communicate with a large number of peers, the scalability issue could be serious. For instance, in [I-D.ietf-ipsecme-ad-vpn-problem], it has been demonstrated it is not trivial to enabling a large number of systems to communicate directly using IPsec to protect the traffic between them.

6.2. Influence on Security Devices

If the data packets transported through out an overlay are encrypted (e.g., by NVEs), it is difficult for a security device, e.g., a firewall deployed on the path connecting two NVEs to inspect the contents of the packets. The firewall can only know which VN the packets belong to through the VN ID transported in the outer header. If a firewall would like to identify which end device sends a packets or which end device a packet is sent to, the firewall can be deployed in some place where it can access the packet before it is encapsulated or un-encapsulated by NVEs. However, in this case, the firewall cannot get VN ID from the packet. If the firewall is used to process two VNs concurrently and there are IP or MAC addresses of the end devices in the two VNs overlapped, confusion will be caused. If a firewall can generate multiple firewalls instances for different tenants respectively, this issue can be largely addressed.

6.3. Security Issues with VM Migration

The support of VM migration is an important issue considered in NVO3 WG. The migration may also cause security risks. Because the VMs within a VN may move from one server to another which connects to a different NVE, the packets exchanging between two VMs may be transferred in a new path. If the security policies deployed on the firewalls of the two paths are conflict or the firewalls on the new path lack essential state to process the packets. The communication between the VMs may be broken. To address this problem, one option is to enable the state migration and policy confliction detection between firewalls. The other one is to force all the traffic within a VN be processed by a single firewall. However this solution may cause traffic optimization issues.

7. IANA Considerations

This document makes no request of IANA.

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

8. Security Considerations

TBD

9. Acknowledgements

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

- [I-D.ietf-ipsecme-ad-vpn-problem]
Hanna, S. and V. Manral, "Auto Discovery VPN Problem Statement and Requirements", draft-ietf-ipsecme-ad-vpn-problem-08 (work in progress), July 2013.
- [I-D.ietf-nvo3-framework]
Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-03 (work in progress), July 2013.
- [I-D.ietf-nvo3-overlay-problem-statement]
Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-03 (work in progress), May 2013.
- [I-D.kreeger-nvo3-hypervisor-nve-cp]
Kreeger, L., Narten, T., and D. Black, "Network Virtualization Hypervisor-to-NVE Overlay Control Protocol Requirements", draft-kreeger-nvo3-hypervisor-nve-cp-01 (work in progress), February 2013.
- [I-D.mahalingam-dutt-dcops-vxlan]
Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over

Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-04
(work in progress), May 2013.

[RFC4137] Vollbrecht, J., Eronen, P., Petroni, N., and Y. Ohba,
"State Machines for Extensible Authentication Protocol
(EAP) Peer and Authenticator", RFC 4137, August 2005.

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

Authors' Addresses

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

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

Dacheng Zhang
Huawei
Beijing
China

Email: zhangdacheng@huawei.com

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

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

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

Nabil Bitar
Verizon

Marc Lasserre
Florin Balus
Alcatel-Lucent

Thomas Morin
France Telecom Orange

Lizhong Jin

Bhumip Khasnabish
ZTE

April 15, 2014

NVO3 Data Plane Requirements
draft-ietf-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). 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 Oct 15, 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.

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.....	3
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. Quality of Service (QoS) identifier.....	8
3.3.2. Tunneling function.....	9
3.3.2.1. LAG and ECMP.....	9
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. Gateway (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. Load-balancing.....	13

3.4.2.2. Triangular Routing Issues.....	14
3.5. Path MTU.....	14
3.6. Hierarchical NVE dataplane requirements.....	15
3.7. Other considerations.....	15
3.7.1. Data Plane Optimizations.....	15
3.7.2. NVE location trade-offs.....	15
4. Security Considerations.....	16
5. IANA Considerations.....	16
6. References.....	16
6.1. Normative References.....	16
6.2. Informative References.....	16
7. Acknowledgments.....	17

1. Introduction

1.1. Conventions used in this document

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

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

1.2. General terminology

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

BUM: Broadcast, Unknown Unicast, Multicast traffic

TS: Tenant System

2. Data Path Overview

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

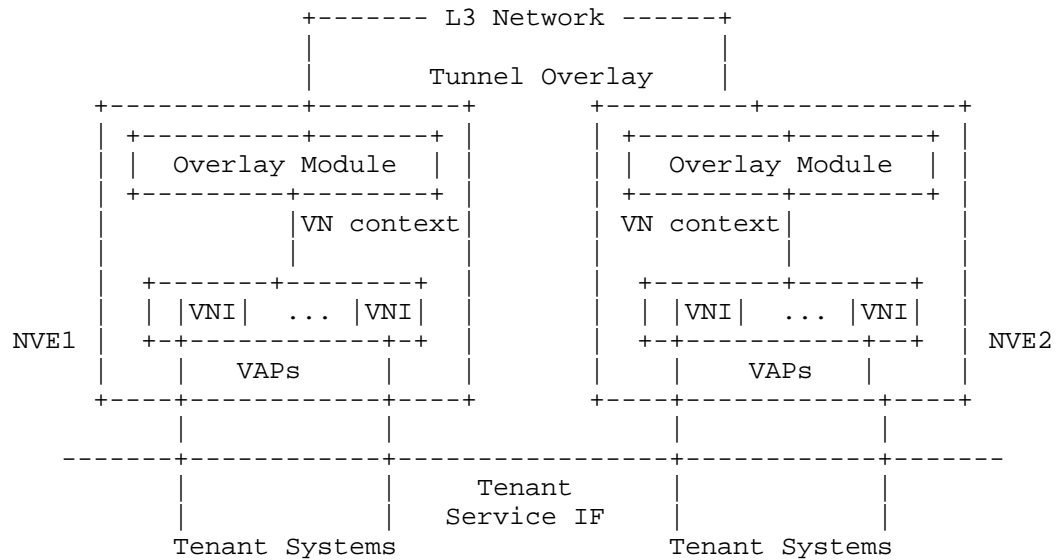


Figure 1 : Generic reference model for NV Edge

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

Once a corresponding VNI is identified, a lookup is performed to determine where the frame needs to be sent. This lookup can be based on any combinations of various fields in the data frame (e.g., destination MAC addresses and/or destination IP addresses). Note that additional criteria such as Ethernet 802.1p priorities 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 virtual switch or a physical port on a ToR switch
- 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 could be 802.1Q enabled (allowing use of VLAN tags as a VAP). An L2 VNI provides per tenant virtual switching instance with MAC addressing isolation and L3 tunneling. Loop avoidance capability **MUST** be provided.

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

Unless a control plane is used to disseminate address mappings, data plane learning **MUST** be used to populate forwarding tables. As frames arrive from VAPs or from overlay tunnels, standard MAC learning procedures are used: The tenant system source MAC address is learned against the VAP or the NVO3 tunneling encapsulation source address on which the frame arrived. Data plane learning implies that unknown unicast traffic will 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.

When using underlay multicast, the NVE **MUST** have one or more underlay multicast trees that can be used by local VNIs for flooding to NVEs belonging to the same VN. For each VNI, there is at least one underlay flooding tree used for Broadcast, Unknown Unicast and Multicast forwarding. This tree **MAY** be shared across VNIs. The flooding tree is equivalent with a multicast (*,G) construct where all the NVEs for which the corresponding VNI is instantiated are members.

When tenant multicast is supported, it **SHOULD** also be possible to select whether the NVE provides optimized underlay 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. [RFC2983] provides additional details on the uniform and pipe models.

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.

When underlay multicast is supported, it MAY be possible to select whether the NVE provides optimized underlay multicast trees inside the VNI for individual tenant multicast groups or whether a default underlay VNI multicasting tree, where all the NVEs of the corresponding VNI are members, is used.

3.3. Overlay Module

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

The following figure shows a generic NVO3 encapsulated frame:

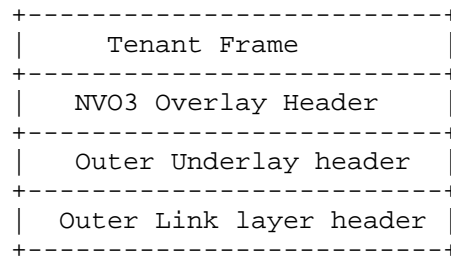


Figure 2 : NVO3 encapsulated frame

where

- . Tenant frame: Ethernet or IP based upon the VNI type

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

3.3.1. NVO3 overlay header

An NVO3 overlay header **MUST** be included after the underlay tunnel header when forwarding tenant traffic.

Note that this information can be carried within existing protocol headers (when overloading of specific fields is possible) or within a separate header.

3.3.1.1. Virtual Network Context Identification

The overlay encapsulation header **MUST** contain a field which allows the encapsulated frame to be delivered to the appropriate virtual network endpoint by the egress NVE.

The egress NVE uses this field to determine the appropriate virtual network context in which to process the packet. This field **MAY** be an explicit, unique (to the administrative domain) virtual network identifier (VNID) or **MAY** express the necessary context information in other ways (e.g. a locally significant identifier).

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 typically no such constraint when using a local identifier.

3.3.1.2. Quality of 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 (Cos) across or between overlay networks **MAY** be required.

To effectively enforce CoS across or between overlay networks without Deep Packet Inspection (DPI) repeat, 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: Support for NVE Service CoS MAY be provided through a QoS field, inside the NVO3 overlay header
 - 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.

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 MAY be supported.

3.3.2.1. LAG and ECMP

For performance reasons, multipath over LAG and ECMP paths MAY 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 needs to present sufficient entropy to exercise all paths through several LAG/ECMP hops.

The entropy information can be inferred from the NVO3 overlay header or underlay header. If the overlay protocol does not support the necessary entropy information or the switches/routers in the underlay do not support parsing of the additional entropy information in the overlay header, underlay switches and routers should be programmable, i.e. select the appropriate fields in the underlay header for hash calculation based on the type of overlay header.

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

The goal is for all paths available to the overlay network to 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.

[RFC6040] defines ECN marking and processing 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-configurable settings 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 maintain 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

It is important that NVO3 services 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 (GW).

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 GW 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. Gateway (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. Internet connected Tenants require translation from NVO3 encapsulation to IP in the NVO3 gateway. The translation function SHOULD minimize provisioning touches.

3.4.1.2. Inter-DC GW

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

3.4.1.3. Intra-DC gateways

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

3.4.2. Path optimality between NVEs and Gateways

Within an 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 may 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. In this case, there are 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 need to 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). 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.

3.4.2.1. Load-balancing

When using 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 specific to each tenant.

3.4.2.2. Triangular Routing Issues

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 (a.k.a. traffic tromboning).

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 tunnel.

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 dataplane requirements

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. In this case, a hub NVE MUST provide additional data processing capabilities such as packet replication.

3.7. Other considerations

3.7.1. 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 virtual switches)

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 NV03 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.

3.7.2. 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, David Black, Dimitrios Stiliadis, Rotem Salomonovitch, Larry Kreeger, Eric Gray and Erik Nordmark.

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

Authors' Addresses

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

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

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

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

Lizhong Jin
Email : lizho.jin@gmail.com

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

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

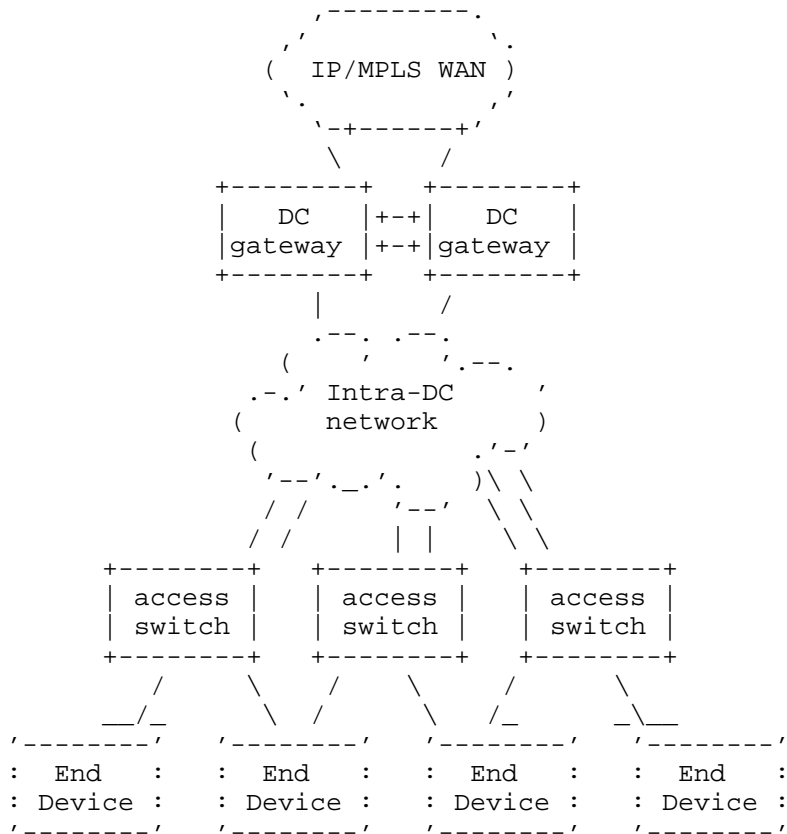


Figure 1 : A Generic Architecture for Data Centers

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.

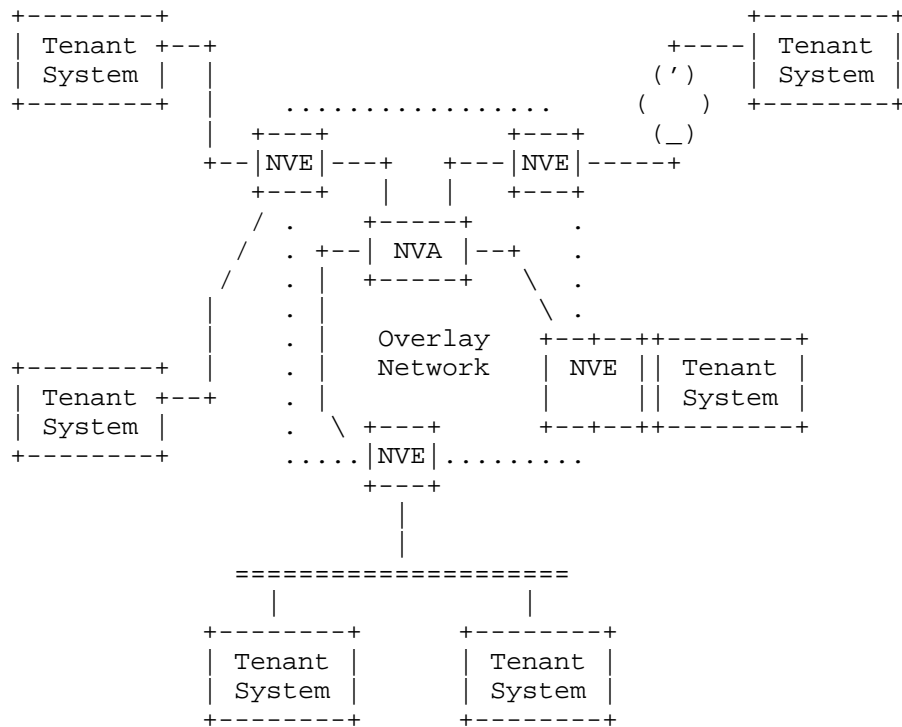


Figure 2 : Generic reference model for DC network virtualization overlay

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.).

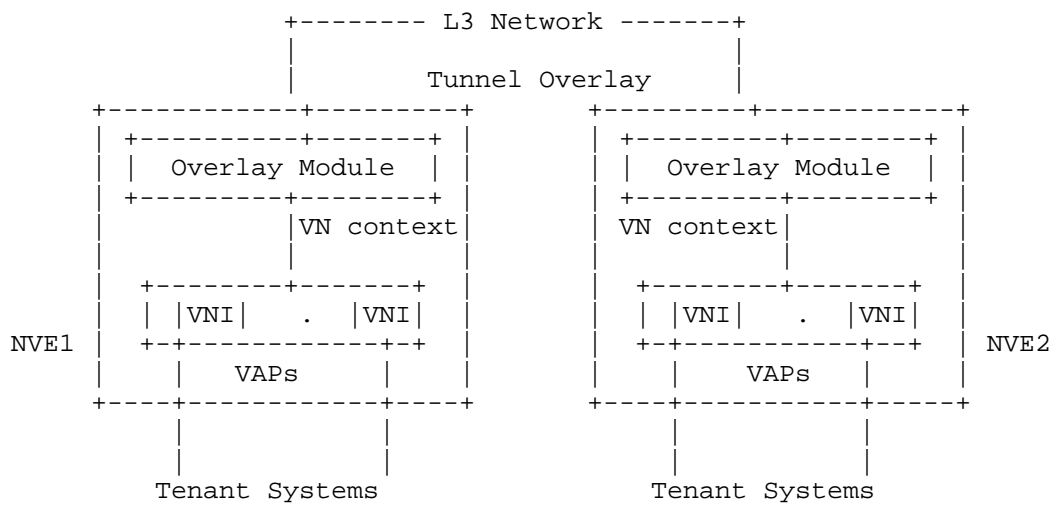


Figure 3 : Generic NVE reference model

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

Network Working Group
Internet Draft
Category: Informational

L. Yong
L. Dunbar
Huawei
M. Toy
Verizon
A. Isaac
Juniper Networks
V. Manral
Ionos Networks

Expires: July 2017

February 20, 2017

Use Cases for Data Center Network Virtualization Overlay Networks

draft-ietf-nvo3-use-case-17

Abstract

This document describes data center network virtualization overlay (NVO3) network use cases that can be deployed in various data centers and serve different data center applications.

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 July 21, 2017.

Copyright Notice

Copyright (c) 2016 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. Terminology.....	4
1.2. NVO3 Background.....	5
2. DC with Large Number of Virtual Networks.....	6
3. DC NVO3 virtual network and External Network Interconnection...	6
3.1. DC NVO3 virtual network Access via the Internet.....	7
3.2. DC NVO3 virtual network and SP WAN VPN Interconnection....	8
4. DC Applications Using NVO3.....	9
4.1. Supporting Multiple Technologies.....	9
4.2. DC Applications Spanning Multiple Physical Zones.....	10
4.3. Virtual Data Center (vDC).....	10
5. Summary.....	12
6. Security Considerations.....	12
7. IANA Considerations.....	13
8. Informative References.....	13
Contributors.....	14
Acknowledgements.....	14
Authors' Addresses.....	15

1. Introduction

Server virtualization has changed the Information Technology (IT) industry in terms of the efficiency, cost, and speed of providing new applications and/or services such as cloud applications. However traditional data center (DC) networks have limits in supporting cloud applications and multi tenant networks [RFC7364]. The goals of data center network virtualization overlay (NVO3) networks are to decouple the communication among tenant systems from DC physical infrastructure networks and to allow one physical network infrastructure to:

- o Carry many NVO3 virtual networks and isolate the traffic of different NVO3 virtual networks on a physical network.
- o Provide independent address space in individual NVO3 virtual network such as MAC and IP.
- o Support flexible Virtual Machines (VM) and/or workload placement including the ability to move them from one server to another without requiring VM address changes and physical infrastructure network configuration changes, and the ability to perform a "hot move" with no disruption to the live application running on those VMs.

These characteristics of NVO3 virtual networks help address the issues that cloud applications face in data centers [RFC7364].

Hosts in one NVO3 virtual network may communicate with hosts in another NVO3 virtual network that is carried by the same physical network, or different physical network, via a gateway. The use case examples for the latter are: 1) DCs that migrate toward an NVO3 solution will be done in steps, where a portion of tenant systems in a VN are on virtualized servers while others exist on a LAN. 2) many DC applications serve to Internet users who are on different physical networks; 3) some applications are CPU bound, such as Big Data analytics, and may not run on virtualized resources. The inter-VN policies are usually enforced by the gateway.

This document describes general NVO3 virtual network use cases that apply to various data centers. The use cases described here represent DC provider's interests and vision for their cloud services. The document groups the use cases into three categories from simple to sophisticated in terms of implementation. However the implementation details of these use cases are outside the scope of this document. These three categories are highlighted below:

- o Basic NVO3 virtual networks (Section 2). All Tenant Systems (TS) in the network are located within the same DC. The individual networks can be either Layer 2 (L2) or Layer 3 (L3). The number of NVO3 virtual networks in a DC is much larger than the number that traditional VLAN based virtual networks [IEEE 802.1Q] can support.
- o A virtual network that spans across multiple Data Centers and/or to customer premises where NVO3 virtual networks are constructed and interconnect other virtual or physical networks outside the data center. An enterprise customer may use a traditional carrier-grade VPN or an IPsec tunnel over the Internet to communicate with its systems in the DC. This is described in Section 3.
- o DC applications or services require an advanced network that contains several NVO3 virtual networks that are interconnected by gateways. Three scenarios are described in Section 4. (1) supporting multiple technologies; (2) constructing several virtual networks as a tenant network; (3) applying NVO3 to a virtual Data Center (vDC).

The document uses the architecture reference model defined in [RFC7365] to describe the use cases.

1.1. Terminology

This document uses the terminology defined in [RFC7365] and [RFC4364]. Some additional terms used in the document are listed here.

ASBR: Autonomous System Border Routers (ASBR)

DMZ: Demilitarized Zone. A computer or small sub-network that sits between a more trusted internal network, such as a corporate private LAN, and an un-trusted or less trusted external network, such as the public Internet.

DNS: Domain Name Service [RFC1035]

DC Operator: An entity that is responsible for constructing and managing all resources in data centers, including, but not limited to, compute, storage, networking, etc.

DC Provider: An entity that uses its DC infrastructure to offer services to its customers.

NAT: Network Address Translation [RFC3022]

vGW: virtual Gateway; a gateway component used for an NVO3 virtual network to interconnect with another virtual/physical network.

NVO3 virtual network: a virtual network that is implemented based NVO3 architecture [NVO3-ARCH].

PE: Provider Edge

SP: Service Provider

TS: A TS can be a physical server/device or a virtual machine (VM) on a server, i.e., end-device [RFC7365].

VRF-LITE: Virtual Routing and Forwarding - LITE [VRF-LITE]

VN: NVO3 virtual network.

WAN VPN: Wide Area Network Virtual Private Network [RFC4364]
[RFC7432]

1.2. NVO3 Background

An NVO3 virtual network is a virtual network in a DC that is implemented based on the NVO3 architecture [RFC8014]. This architecture is often referred to as an overlay architecture. The traffic carried by an NVO3 virtual network is encapsulated at a Network Virtual Edge (NVE) [RFC8014] and carried by a tunnel to another NVE where the traffic is decapsulated and sent to a destination Tenant System (TS). The NVO3 architecture decouples NVO3 virtual networks from the DC physical network configuration. The architecture uses common tunnels to carry NVO3 traffic that belongs to multiple NVO3 virtual networks.

An NVO3 virtual network may be an L2 or L3 domain. The network provides switching (L2) or routing (L3) capability to support host (i.e., tenant systems) communications. An NVO3 virtual network may be required to carry unicast traffic and/or multicast, broadcast/unknown-unicast (for L2 only) traffic from/to tenant systems. There are several ways to transport NVO3 virtual network BUM (Broadcast, Unknown-unicast, Multicast) traffic [NVO3MCAST].

An NVO3 virtual network provides communications among Tenant Systems (TS) in a DC. A TS can be a physical server/device or a virtual machine (VM) on a server end-device [RFC7365].

2. DC with Large Number of Virtual Networks

A DC provider often uses NVO3 virtual networks for internal applications where each application runs on many VMs or physical servers and the provider requires applications to be segregated from each other. A DC may run a larger number of NVO3 virtual networks to support many applications concurrently, where traditional IEEE802.1Q based VLAN solution is limited to 4094 VLANs.

Applications running on VMs may require different quantity of computing resource, which may result in computing resource shortage on some servers and other servers being nearly idle. Shortage of computing resource may impact application performance. DC operators desire VM or workload movement for resource usage optimization. VM dynamic placement and mobility results in frequent changes of the binding between a TS and an NVE. The TS reachability update mechanisms should take significantly less time than the typical re-transmission Time-out window of a reliable transport protocol such as TCP and SCTP, so that end points' transport connections won't be impacted by a TS becoming bound to a different NVE. The capability of supporting many TSs in a virtual network and many virtual networks in a DC is critical for an NVO3 solution.

When NVO3 virtual networks segregate VMs belonging to different applications, DC operators can independently assign MAC and/or IP address space to each virtual network. This addressing is more flexible than requiring all hosts in all NVO3 virtual networks to share one address space. In contrast, typical use of IEEE 802.1Q VLANs requires a single common MAC address space.

3. DC NVO3 virtual network and External Network Interconnection

Many customers (enterprises or individuals) who utilize a DC provider's compute and storage resources to run their applications need to access their systems hosted in a DC through Internet or Service Providers' Wide Area Networks (WAN). A DC provider can construct a NVO3 virtual network that provides connectivity to all the resources designated for a customer and allows the customer to access the resources via a virtual gateway (vGW). WAN connectivity to the virtual gateway can be provided by VPN technologies such as IPsec VPNs [RFC4301] and BGP/MPLS IP VPNs [RFC 4364].

If a virtual network spans multiple DC sites, one design using NVO3 is to allow the network to seamlessly span the sites without DC gateway routers' termination. In this case, the tunnel between a

pair of NVEs can be carried within other intermediate tunnels over the Internet or other WANs, or an intra-DC tunnel and inter DC tunnel(s) can be stitched together to form an end-to-end tunnel between the pair of NVEs that are in different DC sites. Both cases will form one NVO3 virtual network across multiple DC sites.

Two use cases are described in the following sections.

3.1. DC NVO3 virtual network Access via the Internet

A customer can connect to an NVO3 virtual network via the Internet in a secure way. Figure 1 illustrates an example of this case. The NVO3 virtual network has an instance at NVE1 and NVE2 and the two NVEs are connected via an IP tunnel in the Data Center. A set of tenant systems are attached to NVE1 on a server. NVE2 resides on a DC Gateway device. NVE2 terminates the tunnel and uses the VNID on the packet to pass the packet to the corresponding vGW entity on the DC GW (the vGW is the default gateway for the virtual network). A customer can access their systems, i.e., TS1 or TSn, in the DC via the Internet by using an IPsec tunnel [RFC4301]. The IPsec tunnel is configured between the vGW and the customer gateway at the customer site. Either a static route or Interior Border Gateway Protocol (iBGP) may be used for prefix advertisement. The vGW provides IPsec functionality such as authentication scheme and encryption; iBGP protocol traffic is carried within the IPsec tunnel. Some vGW features are listed below:

- o The vGW maintains the TS/NVE mappings and advertises the TS prefix to the customer via static route or iBGP.
- o Some vGW functions such as firewall and load balancer can be performed by locally attached network appliance devices.
- o If the NVO3 virtual network uses different address space than external users, then the vGW needs to provide the NAT function.
- o More than one IPsec tunnel can be configured for redundancy.
- o The vGW can be implemented on a server or VM. In this case, IP tunnels or IPsec tunnels can be used over the DC infrastructure.
- o DC operators need to construct a vGW for each customer.

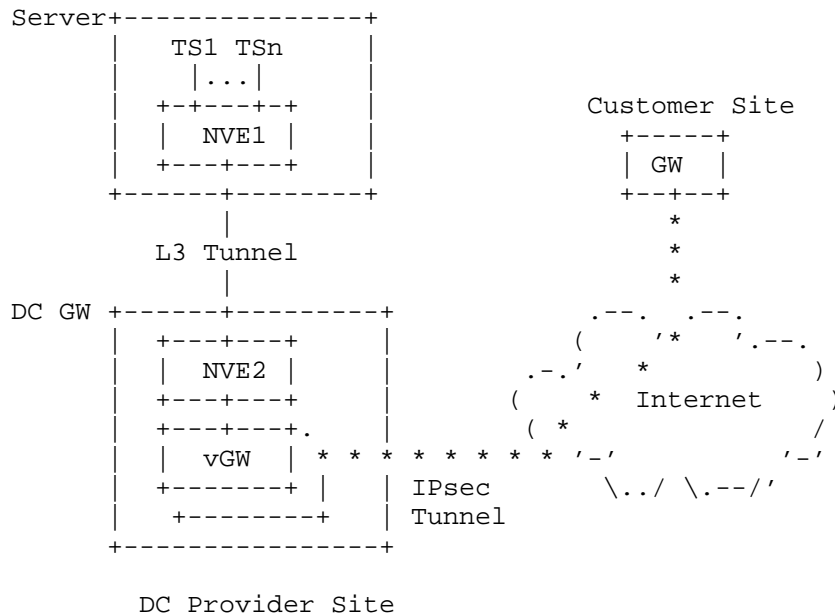


Figure 1 - DC Virtual Network Access via the Internet

3.2. DC NVO3 virtual network and SP WAN VPN Interconnection

In this case, an Enterprise customer wants to use a Service Provider (SP) WAN VPN [RFC4364] [RFC7432] to interconnect its sites with an NVO3 virtual network in a DC site. The Service Provider constructs a VPN for the enterprise customer. Each enterprise site peers with an SP PE. The DC Provider and VPN Service Provider can build an NVO3 virtual network and a WAN VPN independently, and then interconnect them via a local link, or a tunnel between the DC GW and WAN Provider Edge (PE) devices. The control plane interconnection options between the DC and WAN are described in [RFC4364]. Using the option A specified in [RFC4364] with VRF-LITE [VRF-LITE], both Autonomous System Border Routers (ASBR), i.e., DC GW and SP PE, maintain a routing/forwarding table (VRF). Using the option B specified in [RFC4364], the DC ASBR and SP ASBR do not maintain the VRF table; they only maintain the NVO3 virtual network and VPN identifier mappings, i.e., label mapping, and swap the label on the packets in the forwarding process. Both option A and B allow the NVO3 virtual network and VPN using their own identifiers and two identifiers are mapped at DC GW. With the option C in [RFC4364], the VN and VPN use the same identifier and both ASBRs perform the tunnel

stitching, i.e., tunnel segment mapping. Each option has pros/cons [RFC4364] and has been deployed in SP networks depending on the application requirements. BGP is used in these options for route distribution between DCs and SP WANs. Note that if the DC is the SP's Data Center, the DC GW and SP PE in this case can be merged into one device that performs the interworking of the VN and VPN within an AS.

These solutions allow the enterprise networks to communicate with the tenant systems attached to the NVO3 virtual network in the DC without interfering with the DC provider's underlying physical networks and other NVO3 virtual networks in the DC. The enterprise can use its own address space in the NVO3 virtual network. The DC provider can manage which VM and storage elements attach to the NVO3 virtual network. The enterprise customer manages which applications run on the VMs without knowing the location of the VMs in the DC. (See Section 4 for more)

Furthermore, in this use case, the DC operator can move the VMs assigned to the enterprise from one server to another in the DC without the enterprise customer being aware, i.e., with no impact on the enterprise's 'live' applications. Such advanced technologies bring DC providers great benefits in offering cloud services, but add some requirements for NVO3 [RFC7364] as well.

4. DC Applications Using NVO3

NVO3 technology provides DC operators with the flexibility in designing and deploying different applications in an end-to-end virtualization overlay environment. The operators no longer need to worry about the constraints of the DC physical network configuration when creating VMs and configuring a network to connect them. A DC provider may use NVO3 in various ways, in conjunction with other physical networks and/or virtual networks in the DC. This section highlights some use cases for this goal.

4.1. Supporting Multiple Technologies

Servers deployed in a large data center are often installed at different times, and may have different capabilities/features. Some servers may be virtualized, while others may not; some may be equipped with virtual switches, while others may not. For the servers equipped with Hypervisor-based virtual switches, some may support a standardized NVO3 encapsulation, some may not support any encapsulation, and some may support a documented encapsulation protocol (e.g. VxLAN [RFC7348], NVGRE [RFC7637]) or proprietary encapsulations. To construct a tenant network among these servers

and the ToR switches, operators can construct one traditional VLAN network and two virtual networks where one uses VxLAN encapsulation and the other uses NVGRE, and interconnect these three networks via a gateway or virtual GW. The GW performs packet encapsulation/decapsulation translation between the networks.

Another case is that some software of a tenant has high CPU and memory consumption, which only makes a sense to run on standalone servers; other software of the tenant may be good to run on VMs. However provider DC infrastructure is configured to use NVO3 to connect VMs and VLAN [IEEE802.1Q] to physical servers. The tenant network requires interworking between NVO3 and traditional VLAN.

4.2. DC Applications Spanning Multiple Physical Zones

A DC can be partitioned into multiple physical zones, with each zone having different access permissions and runs different applications. For example, a three-tier zone design has a front zone (Web tier) with Web applications, a mid zone (application tier) where service applications such as credit payment or ticket booking run, and a back zone (database tier) with Data. External users are only able to communicate with the Web application in the front zone; the back zone can only receive traffic from the application zone. In this case, communications between the zones must pass through one or more security functions in a physical DMZ zone. Each zone can be implemented by one NVO3 virtual network and the security functions in DMZ zone can be used to between two NVO3 virtual networks, i.e., two zones. If network functions (NF), especially the security functions in the physical DMZ can't process encapsulated NVO3 traffic, the NVO3 tunnels have to be terminated for the NF to perform its processing on the application traffic.

4.3. Virtual Data Center (vDC)

An enterprise data center today may deploy routers, switches, and network appliance devices to construct its internal network, DMZ, and external network access; it may have many servers and storage running various applications. With NVO3 technology, a DC Provider can construct a virtual Data Center (vDC) over its physical DC infrastructure and offer a virtual Data Center service to enterprise customers. A vDC at the DC Provider site provides the same capability as the physical DC at a customer site. A customer manages its own applications running in its vDC. A DC Provider can further offer different network service functions to the customer. The network service functions may include firewall, DNS, load balancer, gateway, etc.

Figure 2 below illustrates one such scenario at the service abstraction level. In this example, the vDC contains several L2 VNs (L2VNx, L2VNy, L2VNz) to group the tenant systems together on a per-application basis, and one L3 VN (L3VNa) for the internal routing. A network firewall and gateway runs on a VM or server that connects to L3VNa and is used for inbound and outbound traffic processing. A load balancer (LB) is used in L2VNx. A VPN is also built between the gateway and enterprise router. An Enterprise customer runs Web/Mail/Voice applications on VMs within the vDC. The users at the Enterprise site access the applications running in the vDC via the VPN; Internet users access these applications via the gateway/firewall at the provider DC site.

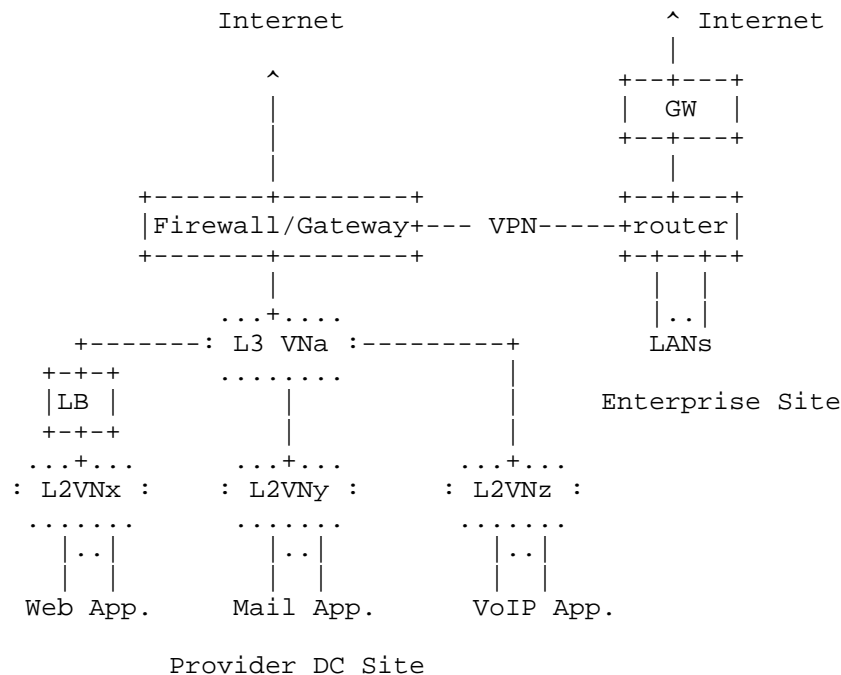


Figure 2 - Virtual Data Center Abstraction View

The enterprise customer decides which applications should be accessible only via the intranet and which should be assessable via both the intranet and Internet, and configures the proper security policy and gateway function at the firewall/gateway. Furthermore, an

enterprise customer may want multi-zones in a vDC (See section 4.2) for the security and/or the ability to set different QoS levels for the different applications.

The vDC use case requires an NVO3 solution to provide DC operators with an easy and quick way to create an NVO3 virtual network and NVEs for any vDC design, to allocate TSs and assign TSs to the corresponding NVO3 virtual network, and to illustrate vDC topology and manage/configure individual elements in the vDC in a secure way.

5. Summary

This document describes some general NVO3 use cases in DCs. The combination of these cases will give operators the flexibility and capability to design more sophisticated support for various cloud applications.

DC services may vary, NVO3 virtual networks make it possible to scale a large number of virtual networks in DC and ensure the network infrastructure not impacted by the number of VMs and dynamic workload changes in DC.

NVO3 uses tunnel techniques to deliver NVO3 traffic over DC physical infrastructure network. A tunnel encapsulation protocol is necessary. An NVO3 tunnel may in turn be tunneled over other intermediate tunnels over the Internet or other WANs.

An NVO3 virtual network in a DC may be accessed by external users in a secure way. Many existing technologies can help achieve this.

6. Security Considerations

Security is a concern. DC operators need to provide a tenant with a secured virtual network, which means one tenant's traffic is isolated from other tenants' traffic and is not leaked to the underlay networks. Tenants are vulnerable to observation and data modification/injection by the operator of the underlay and should only use operators they trust. DC operators also need to prevent a tenant application attacking their underlay DC network; further, they need to protect a tenant application attacking another tenant application via the DC infrastructure network. For example, a tenant application attempts to generate a large volume of traffic to overload the DC's underlying network. This can be done by limiting the bandwidth of such communications.

7. IANA Considerations

This document does not request any action from IANA.

8. Informative References

- [IEEE802.1Q] IEEE, "IEEE Standard for Local and metropolitan area networks -- Media Access Control (MAC) Bridges and Virtual Bridged Local Area", IEEE Std 802.1Q, 2011.
- [NIST] National Institute of Standards and Technology, "The NIST Definition of Cloud Computing", SP 880-145, September, 2011.
- [NVO3MCAST] Ghanwani, A., Dunbar, L., et al, "A Framework for Multicast in Network Virtualization Overlays", draft-ietf-nvo3-mcast-framework-05, work in progress.
- [RFC1035] Mockapetris, P., "DOMAIN NAMES - Implementation and Specification", RFC1035, November 1987.
- [RFC3022] Srisuresh, P. and Egevang, K., "Traditional IP Network Address Translator (Traditional NAT)", RFC3022, January 2001.
- [RFC4301] Kent, S., "Security Architecture for the Internet Protocol", rfc4301, December 2005
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [RFC7348] Mahalingam, M., Dutt, D., et al, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC7348 August 2014.
- [RFC7364] Narten, T., et al "Problem Statement: Overlays for Network Virtualization", RFC7364, October 2014.
- [RFC7365] Lasserre, M., Motin, T., et al, "Framework for DC Network Virtualization", RFC7365, October 2014.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A. and J. Uttaro, "BGP MPLS Based Ethernet VPN", RFC7432, February 2015

[RFC7637] Garg, P., and Wang, Y., "NVGRE: Network Virtualization using Generic Routing Encapsulation", RFC7637, Sept. 2015.

[RFC8014] Black, D., et al, "An Architecture for Overlay Networks (NVO3)", rfc8014, January 2017.

[VRF-LITE] Cisco, "Configuring VRF-lite", <http://www.cisco.com>

Contributors

David Black
Dell EMC
176 South Street
Hopkinton, MA 01748
David.Black@dell.com

Vinay Bannai
PayPal
2211 N. First St,
San Jose, CA 95131
Phone: +1-408-967-7784
Email: vbannai@paypal.com

Ram Krishnan
Brocade Communications
San Jose, CA 95134
Phone: +1-408-406-7890
Email: ramk@brocade.com

Kieran Milne
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
Phone: +1-408-745-2000
Email: kmilne@juniper.net

Acknowledgements

Authors like to thank Sue Hares, Young Lee, David Black, Pedro Marques, Mike McBride, David McDysan, Randy Bush, Uma Chunduri, Eric Gray, David Allan, Joe Touch, Olufemi Komolafe, Matthew Bocci, and Alia Atlas for the review, comments, and suggestions.

Authors' Addresses

Lucy Yong
Huawei Technologies

Phone: +1-918-808-1918
Email: lucy.yong@huawei.com

Linda Dunbar
Huawei Technologies,
5340 Legacy Dr.
Plano, TX 75025 US

Phone: +1-469-277-5840
Email: linda.dunbar@huawei.com

Mehmet Toy
Verizon

E-mail : mtoy054@yahoo.com

Aldrin Isaac
Juniper Networks
E-mail: aldrin.isaac@gmail.com

Vishwas Manral

Email: vishwas@ionosnetworks.com

Network Virtualization Overlays
Internet-Draft
Intended status: Standards Track
Expires: January 13, 2014

Chen. Li
Rong. Gu
China Mobile
July 12, 2013

Network as a service requirement in cloud datacenter
draft-li-nvo3-clouddatacenter-requirement-00

Abstract

This document describes some specific features in CDC, especially in the public cloud.

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 13, 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.

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.

Table of Contents

1. Introduction	2
2. Problem statement	2
3. Solution	3
4. Acknowledgements	3
5. References	3
Authors' Addresses	3

1. Introduction

CDC (Cloud Data Center) network has the fastest innovation of the network standards and the most proposed technology. Especially in the public clouds. From our prespective, there are several network capacity can be sold by public clouds' operator: IP address, VLAN, bandwidth, loadbalance, firewall and some other network resouces. The target of NAAS(network as a service) is to provide end to end virtual network with above capacity for tenants in datacenter. However, many taditional tochonlogy become the bottleneck of public cloud service, such as the number of VLAN. It becomes unable to meet the constantly updated needs of providing users with the hosted networks for the data segregation.

In this draft, we forcus on proposing network requirement of NAAS.

2. Problem statement

NAAS is supposed to provide a virtual CDC network for a tenant. we propose four specific network features of NAAS as follows

a. The isolation of different tenants

Different tenants are isolated by vpn, No matter layer 2 or layer 3, no matter by vlan tag or mpls tag or some others. Meanwhile, the network service devices, such as loadbalance and firewall, also need to be isolated.tenants have a logical isolated network, which can be implement any IP and VLAN by themselves (different tenants should reused IP/VLAN).

b. tenant's logical network in GUI

tenant's logical network GUI should be simple and intuitive. For example it only display a L2 switch, a L3 gateway, a broader router, a loadbalance, a firewall and some other security devices. All the link is logical. VMS or servers connect to these logical network devices.

c. bandwidth guarantee

Each logical network should allocate the specific end to end bandwidth, including server uplink switch port rate, switch to gateway link rate, gateway to LB/FW link rate and broader router link rate.

all the logical bandwidth allocation should map in physical network devices.

d. self network management

Each tenant manage and config their own logical network. While operator is responsible for the physical one.

3. Solution

to be continued.

4. Acknowledgements

5. References

Authors' Addresses

Chen Li
China Mobile
Unit2, Dacheng Plaza, No.28 Xuanwumenxi Ave, Xuanwu District
Beijing 100053
P.R. China

Email: lichenyj@chinamobile.com

Rong Gu
China Mobile
Unit2, Dacheng Plaza, No.28 Xuanwumenxi Ave, Xuanwu District
Beijing 100053
P.R. China

Email: arielgurong@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 09, 2014

D. Black
EMC
J. Hudson
Brocade
L. Kreeger
Cisco
M. Lasserre
Alcatel-Lucent
T. Narten
IBM
July 08, 2013

An Architecture for Overlay Networks (NVO3)
draft-narten-nvo3-arch-00

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 09, 2014.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Terminology	3
3. Background	4
3.1. VN Service (L2 and L3)	5
3.2. Network Virtualization Edge (NVE)	6
3.3. Network Virtualization Authority (NVA)	8
3.4. VM Orchestration Systems	8
4. Network Virtualization Edge (NVE)	9
4.1. NVE Co-located With Server Hypervisor	10
4.2. Split-NVE	10
4.3. NVE State	11
5. Tenant Systems Types	12
5.1. Overlay-Aware Network Service Appliances	12
5.2. Bare Metal Servers	12
5.3. Gateways	13
6. Network Virtualization Authority	13
6.1. How an NVA Obtains Information	14
6.2. Internal NVA Architecture	14
6.3. NVA External Interface	15
7. NVE-to-NVA Protocol	15
7.1. NVE-NVA Interaction Models	15
7.2. Direct NVE-NVA Protocol	16
7.3. Push vs. Pull Model	17
8. Federated NVAs	17
8.1. Inter-NVA Peering	20
9. Control Protocol Work Areas	20
10. NVO3 Data Plane Encapsulation	20
11. Operations and Management	21
12. Summary	21
13. Acknowledgments	22
14. IANA Considerations	22

15. Security Considerations	22
16. Informative References	22
Authors' Addresses	23

1. Introduction

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

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

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

2. Terminology

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

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

NV Region A set of two or more NV Domains that share information about part or all of a set of virtual networks that the individual NV Domains manage. Two NVAs share information about particular

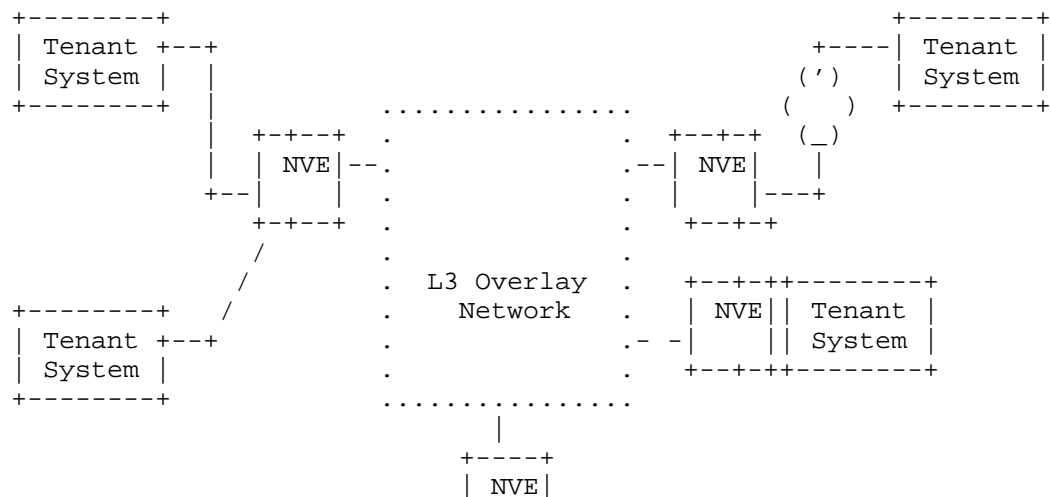
virtual networks for the purpose of supporting connectivity between tenants located in different NVA Domains. NVAs can share information about an entire NV domain, or just individual virtual networks.

3. Background

Overlay networks are an approach for providing network virtualization services to a set of Tenant Systems (TSS) [I-D.ietf-nvo3-framework]. With overlays, data traffic between tenants is tunneled across the underlying data center's IP network. The use of tunnels provides a number of benefits by decoupling the network as viewed by tenants from the underlying physical network across which they communicate.

Tenant Systems connect to Virtual Networks (VNs), with each VN having associated attributes defining properties of the network, such as the set of members that connect to it. Tenant Systems connected to a virtual network typically communicate freely with other Tenant Systems on the same VN, but communication between Tenant Systems on one VN and those external to the VN (whether on another VN or connected to the Internet) is carefully controlled and governed by policy.

A Network Virtualization Edge (NVE) [I-D.ietf-nvo3-framework] is the entity that implements the overlay functionality. An NVE resides at the boundary between a Tenant System and the overlay network as shown in Figure 1. An NVE creates and maintains local state about each Virtual Network for which it is providing service on behalf of a Tenant System.



note that whether NVO3 provides L2 or L3 service to a Tenant System, the Tenant System does not generally need to be aware of the distinction. In both cases, the virtual network presents itself to the Tenant System as an L2 Ethernet interface. An Ethernet interface is used in both cases simply as a widely supported interface type that essentially all Tenant Systems already support. Consequently, no special software is needed on Tenant Systems to use an L3 vs. an L2 overlay service.

3.2. Network Virtualization Edge (NVE)

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

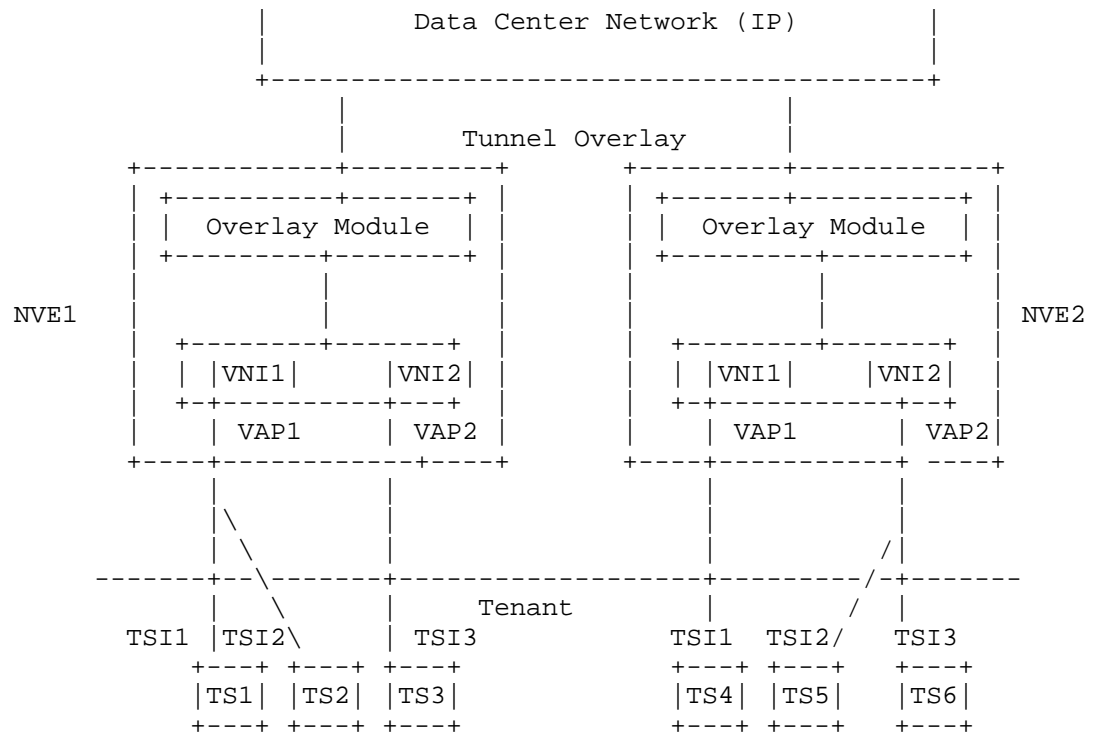


Figure 2: NVE Reference Model

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

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

Conceptually, the NVE is a single entity implementing the NV03 functionality. In practice, there are a number of different implementation scenarios, as described in detail in Section 4.

3.3. Network Virtualization Authority (NVA)

Address dissemination refers to the process of learning, building and distributing the mapping/forwarding information that NVEs need in order to tunnel traffic to each other on behalf of communicating Tenant Systems. For example, in order to send traffic to a remote Tenant System, the sending NVE must know the destination NVE for that Tenant System.

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

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

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

NVAs are discussed in more detail in Section 6.

3.4. VM Orchestration Systems

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

When a new VM image is started, the VM Orchestration system determines where the VM should be placed, interacts with the hypervisor on the target server to load and start the server and controls when a VM should be shutdown or migrated elsewhere. VM Orchestration systems also have knowledge about how a VM should connect to a network, possibly including the name of the virtual

network to which a VM is to connect. The VM orchestration system can pass such information to the hypervisor when a VM is instantiated. VM orchestration systems have significant (and sometimes global) knowledge over the domain they manage. They typically know on what servers a VM is running, and meta data associated with VM images can be useful from a network virtualization perspective. For example, the meta data may include the addresses (MAC and IP) the VMs will use and the name(s) of the virtual network(s) they connect to.

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

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

4. Network Virtualization Edge (NVE)

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

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

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

4.1. NVE Co-located With Server Hypervisor

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

Implementing NVE functionality entirely on a server has the disadvantage that server CPU resources must be spent implementing the NVO3 functionality. Experimentation with overlay approaches and previous experience with TCP and checksum adapter offloads suggests that offloading some portions of the encapsulation and decapsulation operations an NVE performs onto the physical network adaptor can produce performance improvements. As has been done with checksum and /or TCP server offload and other optimization approaches, there may be benefits to offloading common operations onto adaptors where possible. Just as important, the addition of an overlay header can disable existing adaptor offload capabilities that are generally not prepared to handle the addition of a new header. In any case, how to distribute the implementation of specific functionality between a server and network adaptors is a matter between server and adaptor vendors and does not require any IETF standardization.

4.2. Split-NVE

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

While the above talks about a scenario involving a hypervisor, it should be noted that the same scenario can apply to Network Service Appliances as discussed in Section 5.1. In general, when this document discusses the interaction between a hypervisor and NVE, the discussion applies to Network Service Appliances as well.

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

4.3. NVE State

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

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

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

5. Tenant Systems Types

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

5.1. Overlay-Aware Network Service Appliances

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

5.2. Bare Metal Servers

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

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

5.3. Gateways

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

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

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

6. Network Virtualization Authority

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

The Network Virtualization Authority (NVA) is the entity that provides address mapping and other information to NVEs. NVEs interact with an NVA to obtain any required information they need in

order to properly forward traffic on behalf of tenants. The term NVA refers to the overall system, without regards to its scope or how it is implemented.

6.1. How an NVA Obtains Information

There are two primary ways in which an NVA can obtain the address dissemination information it manages.

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

However, as described in Section 4 not all NVEs are associated with hypervisors. In such cases, NVAs cannot leverage VM orchestration protocols to interact with an NVE and will instead need to peer directly with them. By peering directly with an NVE, NVAs can obtain information about the TSes connected to that NVE and can distribute information to the NVE about the VNs those TSes are associated with. For example, whenever a Tenant System attaches to an NVE, that NVE would notify the NVA that the TS is now associated with that NVE. Likewise when a TS detaches from an NVE, that NVE would inform the NVA. By communicating directly with NVEs, both the NVA and the NVE are able to maintain up-to-date information about all active tenants and the NVEs to which they are attached.

6.2. Internal NVA Architecture

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

A number of architectural approaches could be used to implement NVAs themselves. NVAs manage address bindings and distribute them to where they need to go. One approach would be to use BGP (possibly

with extensions) and route reflectors. Another approach could use a transaction-based database model with replicated servers. Because the implementation details are local to an NVA, there is no need to pick exactly one solution technology, so long as the external interfaces to the NVEs (and remote NVAs) are sufficiently well defined to achieve interoperability.

6.3. NVA External Interface

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

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

One architectural question is whether interactions between an NVE and NVA all use a single NVA IP address. If NVEs only have one IP address to interact with, it would be the responsibility of the NVA to handle NVA component failures, e.g., by using a "floating IP address" that migrates among NVA components to ensure that the NVA can always be reached via the one address.

Alternatively, an NVA could export multiple IP addresses, making it the responsibility of the NVE to failover to alternate addresses should one fail. The NVA would then also have to ensure that the information provided through all addresses is consistent, so that it would not matter to the NVE which address it used.

7. NVE-to-NVA Protocol

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

As outlined in Section 4.3, an NVE needs certain information in order to perform its functions. To obtain such information from an NVA, an NVE-to-NVA protocol is needed. While having a direct NVE-to-NVA protocol might seem straightforward, the existence of existing VM orchestration systems complicates the choices an NVE has for interacting with the NVA.

7.1. NVE-NVA Interaction Models

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

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

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

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

7.2. Direct NVE-NVA Protocol

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

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

Requirements for a direct NVE-NVA protocol can be found in [I-D.kreeger-nvo3-overlay-cp]

7.3. Push vs. Pull Model

[Note: This section is included to stimulate discussion, as the WG has had a number of discussions on this point. Depending on how WG discussion goes, this section may not even be needed in future versions of the document.]

There has been discussion within NVO3 about a "push vs. pull" approach for NVE-to-NVA interaction. In the push model, the NVA would push address binding information to the NVE. Since the NVA has current knowledge of which NVE each Tenant System is connected to, the NVA can proactively push updates out to the NVEs when they occur. With a push model, the NVE can be more passive, relying on the NVA to ensure that an NVE always has most current information. The push model has the benefit that NVEs will always have the mapping information they need, and do not need to query the NVA on a cache miss. Note that in the push model, it is not required that an NVE maintain information about all virtual networks in the entire NV Domain; an NVE only needs to maintain information about the VNs associated with TSs associated with the NVE.

In the pull model, an NVE may not have all the mappings it needs when it attempts to forward tenant traffic. If an NVE attempts to send traffic to a destination for which it has no forwarding entry, the NVE queries the NVA to get the needed information or to definitively determine that no such entry exists. While the pull model has the advantage that an NVE doesn't need table entries for destinations it is not forwarding traffic to, it has the disadvantage of delaying the sending of traffic on a cache miss.

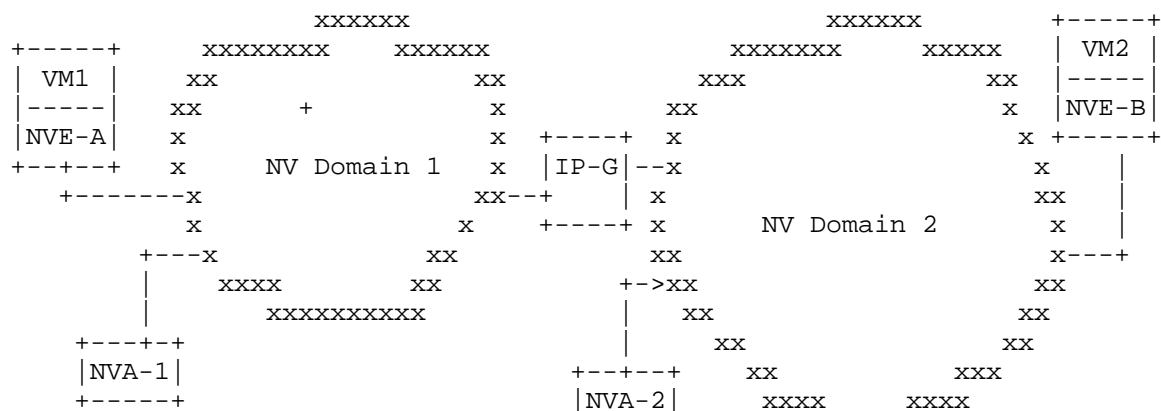
Rather than pick exactly one approach, the NVO3 architecture will likely support flavors of both the push and pull model. In the case that the NVA has updated information to push to the NVEs, there is no reason to prohibit such a model. Likewise, when the NVA is willing to generate queries for missing information on demand, there is no reason to have the architecture prevent such a model.

8. Federated NVAs

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

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

Figure Figure 3 shows a scenario where two separate NV Domains (1 and 2) share information about Virtual Network "1217". VM1 and VM1 both connect to the same Virtual Network (1217), even though the two VMs are in separate NV Domains. There are two cases to consider. In the first case, NV Domain B (NVB) does not allow NVE-A to tunnel traffic directly to NVE-B. There could be a number of reasons for this. For example, NV Domains 1 and 2 may not share a common address space (i.e., require traversal through a NAT device), or for policy reasons, a domain might require that all traffic between separate NV Domains be funneled through a particular device (e.g., a firewall). In such cases, NVA-2 will advertise to NVA-1 that VM1 on virtual network 1217 is available, and direct that traffic between the two nodes go through IP-G. IP-G would then decapsulate received traffic from one NV Domain, translate it appropriately for the other domain and re-encapsulate the packet for delivery.



+-----+ XXXXXXXXX

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

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

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

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

- o Provide isolation between NVAs operating at different sites at different geographic locations.
- o Control the quantity and rate of information updates that flow (and must be processed) between different NVAs in different data centers.
- o Control the set of external NVAs (and external sites) a site peers with. A site will only peer with other sites that are cooperating in providing an overlay service.
- o Allow policy to be applied between sites. A site will want to carefully control what information it exports (and to whom) as well as what information it is willing to import (and from whom).

- o Allow different protocols and architectures to be used to for intra- vs. inter-NVA communication. For example, within a single data center, a replicated transaction server using database techniques might be an attractive implementation option for an NVA, and protocols optimized for intra-NVA communication would likely be different from protocols involving inter-NVA communication between different sites.
- o Allow for optimized protocols, rather than using a one-size-fits all approach. Within a data center, networks tend to have lower-latency, higher-speed and higher redundancy when compared with WAN links interconnecting data centers. The design constraints and tradeoffs for a protocol operating within a data center network are different from those operating over WAN links. While a single protocol could be used for both cases, there could be advantages to using different and more specialized protocols for the intra- and inter-NVA case.

8.1. Inter-NVA Peering

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

9. Control Protocol Work Areas

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

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

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

10. NVO3 Data Plane Encapsulation

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

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

While one might argue that a new encapsulation should be defined just for NVO3, no compelling requirements for doing so have been identified yet. Moreover, optimized implementations for existing encapsulations are already starting to become available on the market (i.e., in silicon). If the IETF were to define a new encapsulation format, it would take at least 2 (and likely more) years before optimized implementations of the new format would become available in products. In addition, a new encapsulation format would not likely displace existing formats, at least not for years. Thus, there seems little reason to define a new encapsulation. However, it does make sense for NVO3 to support multiple encapsulation formats, so as to allow NVEs to use their preferred encapsulations when possible. This implies that the address dissemination protocols must also include an indication of supported encapsulations along with the address mapping details.

11. Operations and Management

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

12. Summary

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

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

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

13. Acknowledgments

Helpful comments and improvements to this document have come from Dennis (Xiaohong) Qin.

14. IANA Considerations

This memo includes no request to IANA.

15. Security Considerations

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

16. Informative References

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

Ashwood-Smith, P., Iyengar, R., Tsou, T., Sajassi, A., Boucadair, M., Jacquenet, C., and M. Daikoku, "NVO3 Operational Requirements", draft-ashwood-nvo3-operational-requirement-02 (work in progress), January 2013.

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

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

[I-D.ietf-nvo3-framework]

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

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

Narten, T., Gray, E., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-03 (work in progress), May 2013.

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

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

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

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

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

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

[I-D.sridharan-virtualization-nvgre]

Sridharan, M., Greenberg, A., 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.

[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.

Authors' Addresses

David Black
EMC

Email: david.black@emc.com

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

Email: jon.hudson@gmail.com

Lawrence Kreeger
Cisco

Email: kreeger@cisco.com

Marc Lasserre
Alcatel-Lucent

Email: marc.lasserre@alcatel-lucent.com

Thomas Narten
IBM

Email: narten@us.ibm.com

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

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

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

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on April 25, 2014.

Copyright Notice

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

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

Table of Contents

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

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

1. Introduction

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

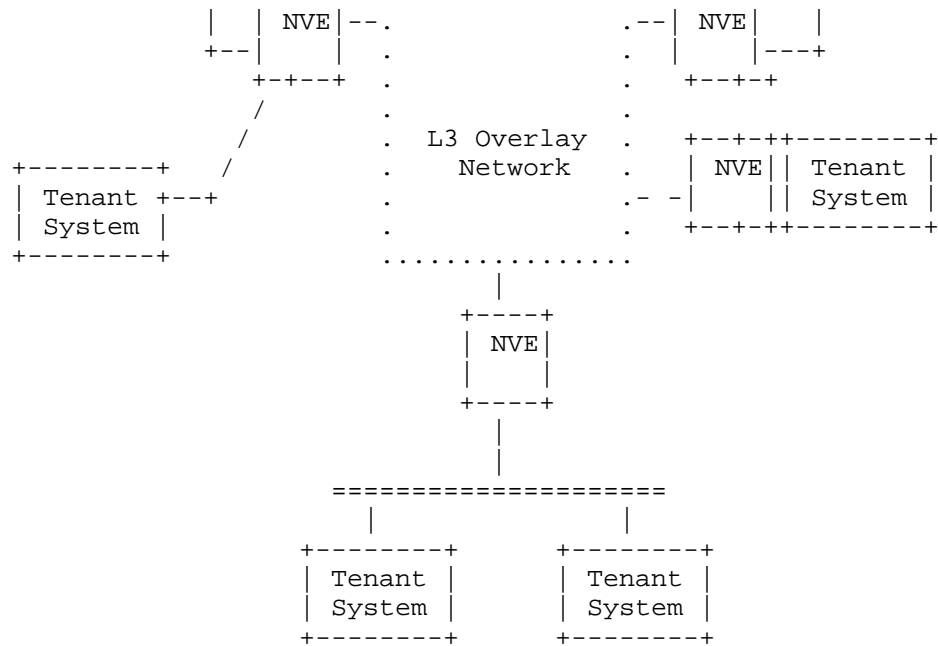
The motivation for overlay networks is given in [I-D.ietf-nvo3-overlay-problem-statement]. "Framework for DC Network Virtualization" [I-D.ietf-nvo3-framework] provides a framework for discussing overlay networks generally and the various components that must work together in building such systems. This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

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

2. Terminology

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

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



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

Figure 1: NV03 Generic Reference Model

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

3.1. VN Service (L2 and L3)

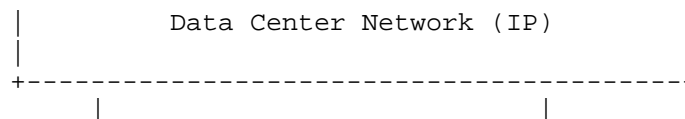
A Virtual Network provides either L2 or L3 service to connected tenants. For L2 service, VNs transport Ethernet frames, and a Tenant System is provided with a service that is analogous to being connected to a specific L2 C-VLAN. L2 broadcast frames are delivered to all (and multicast frames delivered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

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

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

3.2. Network Virtualization Edge (NVE)

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



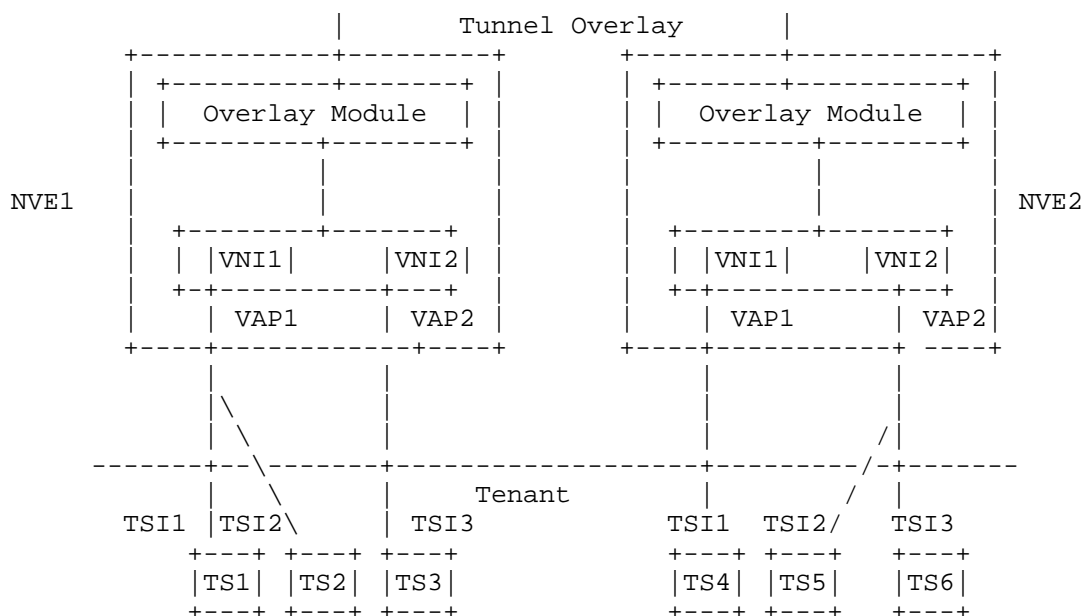


Figure 2: NVE Reference Model

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

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

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

3.3. Network Virtualization Authority (NVA)

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

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

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

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

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

NVAs are discussed in more detail in Section 6.

3.4. VM Orchestration Systems

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

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

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

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

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

4. Network Virtualization Edge (NVE)

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

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

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

4.1. NVE Co-located With Server Hypervisor

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

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

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

4.2. Split-NVE

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

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

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

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

4.3. NVE State

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

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

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

5. Tenant System Types

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

5.1. Overlay-Aware Network Service Appliances

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

5.2. Bare Metal Servers

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

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

5.3. Gateways

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

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

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

5.4. Distributed Gateways

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

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

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

6. Network Virtualization Authority

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

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

6.1. How an NVA Obtains Information

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

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

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

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

6.2. Internal NVA Architecture

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

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

6.3. NVA External Interface

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

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

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

[Note: the following is a strawman proposal.]

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

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

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

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

7. NVE-to-NVA Protocol

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

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

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

7.1. NVE-NVA Interaction Models

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

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

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

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

7.2. Direct NVE-NVA Protocol

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

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

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

7.3. Propagating Information Between NVEs and NVAs

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

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

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

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

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

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

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

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

8. Federated NVAs

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

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

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

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

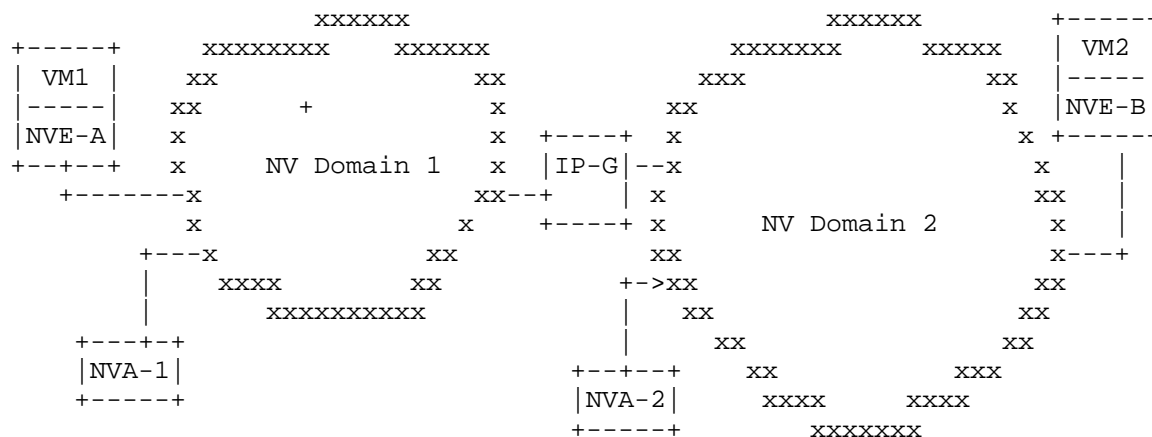


Figure 3: VM1 and VM2 are in different NV Domains.

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

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

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

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

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

8.1. Inter-NVA Peering

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

9. Control Protocol Work Areas

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

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

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

10. NVO3 Data Plane Encapsulation

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

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

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

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

11. Operations and Management

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

12. Summary

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

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

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

13. Acknowledgments

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

14. IANA Considerations

This memo includes no request to IANA.

15. Security Considerations

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

16. Informative References

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

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

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

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

[I-D.ietf-nvo3-framework]

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

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

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

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

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

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

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

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

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

[I-D.sridharan-virtualization-nvgre]

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

[IEEE-802.1Q]

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

Appendix A. Change Log

A.1. Changes From -00 to -01

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

Authors' Addresses

David Black
EMC

Email: david.black@emc.com

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

Email: jon.hudson@gmail.com

Lawrence Kreeger
Cisco

Email: kreeger@cisco.com

Marc Lasserre
Alcatel-Lucent

Email: marc.lasserre@alcatel-lucent.com

Thomas Narten
IBM

Email: narten@us.ibm.com