

Network Virtualization Overlays Working
Group
Internet-Draft
Intended status: Informational
Expires: August 22, 2013

R. Schott
Deutsche Telekom
Q. Wu
Huawei
February 18, 2013

Network Virtualization Overlay Architecture
draft-fw-nvo3-server2vcenter-01.txt

Abstract

Multiple virtual machines (VMs) created in a single physical platform Or vServer greatly improve the efficiency of data centers by enabling more work from less hardware. Multiple vServer and associated virtual machines work together as one cluster make good use of resources of each vServer that are scattered into different data centers or vServers. VMs have their lifecycles from VM creation, VM Power on to VM Power off and VM deletion. The VMs may also move across the participating virtualization hosts (e.g., the virtualization server, hypervisor). This document discusses how VMs, vServers and overlay network are managed by leveraging control plane function and management plane function and desired signaling functionalities for Network Virtualization Overlay.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 22, 2013.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	4
2. Terminology	5
2.1. Standards Language	5
3. Discussions	6
3.1. VM awareness and VM movement awareness	6
3.2. Why VM migration	6
3.3. Who manages VM	7
3.4. VM Grouping	7
3.5. What VM information should be managed	8
3.6. Who Triggers or Controls VM Movements	9
3.7. VM Monitoring	9
4. Use Cases	10
4.1. On Demand Network Provision Automation within the data center	10
4.2. Large inter-data centers Layer 2 interconnection and data forwarding	11
4.3. Enable multiple data centers present as one	12
4.4. VM migration and mobility across data centers	13
5. General Network Virtualization Architecture	15
5.1. NVE (Network Virtualization Edge Function)	16
5.2. vServer (virtualization Server)	17
5.3. vCenter (management plane function)	17
5.4. nDirector (Control plane function)	17
6. vServer to vCenter management interface	19
6.1. VM Creation	19
6.2. VM Termination	19
6.3. VM Registration	19
6.4. VM Unregistration	19
6.5. VM Bulk Registration	19
6.6. VM Bulk Unregistration	19
6.7. VM Configuration Modification	20
6.8. VM Profile Lookup/Discovery	20
6.9. VM Relocation	20
6.10. VM Replication	20
6.11. VM Report	20
7. nDirector to NVE Edge control interface	22
8. vServer to NVE Edge control interface	23
9. Security Considerations	24
10. IANA Considerations	25
11. Contributors	26
12. References	27
12.1. Normative References	27
12.2. Informative References	27
Authors' Addresses	28

1. Introduction

Multiple virtual machines (VMs) created in a single physical platform greatly improve the efficiency of data centers by enabling more work from less hardware. Multiple vServer and associated virtual machines work together as one cluster make good use of resources of each vServer that are scattered into different data centers or vServers. VMs have their lifecycles from VM creation, VM startup to VM poweroff and VM deletion. The VMs may also move across the participating virtualization hosts (e.g., the virtualization server or hypervisor). One example is, as the workload on one physical server increases or physical server needs upgrade, VMs can be moved to other available lightweight-workload servers to ensure that service level agreement and response time requirements are met. We call this VM movement or relocation as VM migration. When the workload decreases, the VMs can be moved back, allowing the unused server powered off to save cost and energy. Another example is as one tenant moves, VMs associated with this tenant may also move to the place that is more close to the tenant and provides better user experience (e.g., larger bandwidth with lower latency). We call such movements as VM mobility. VM migration refers to the transfer of a VM image including memory, storage and network connectivity while VM mobility refers to sending data to the moving tenant associated with the VM and emphasizes service non-disruption during a tenant's movement. This document advocates the distinction between VM mobility and VM migration, both important notions in VM management. The implication is that different signaling or protocols for VM mobility and VM migration might be chosen to automate Network Management for VM Movement, thus possibly reusing the existing protocols or schemes to manage VM migration and VM mobility separately. Unfortunately we sometimes mixed them up or don't distinct VM migration management from VM mobility management and intend to utilize one common protocol to support both VM migration and VM mobility, which seems to simplify the overall protocol design but it is difficult or impossible to run one such protocol across both VM mobility management system that manages VM mobility and VM management platform that manages VM attributes.

This document discusses how VMs, vServer and overlay network to which VMs are connecting are managed, signaling for VM, overlay network management and argues VMs need management or control functionality support but can be managed without VM mobility functionality support.

2. Terminology

2.1. Standards 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].

3. Discussions

3.1. VM awareness and VM movement awareness

Virtual machines usually operate under the control of a server virtualization software residing on the physical compute server. The server virtualization software is commonly referred to as 'hypervisor'. The hypervisor is the container of the VM and provides shared compute/memory/storage and network connectivity to each VM that it hosts. Therefore the hypervisor or the virtualized server MUST be aware of VMs that it hosts. However it should not be aware of VMs that it doesn't host. When VMs hosted in different virtualization servers need to communicate each other, packets from one VM will be forwarded by a virtual switch within the virtualization server or the hypervisor to other VMs on another virtualization server. Since the virtual switch resides within the hypervisor or virtualization server, the rule on VM awareness applied to the hypervisor should apply to virtual switch too.

Unlike VM awareness, VM movement awareness is the capability of knowing the location update of the VM. For example, when a VM moves out of the hypervisor and goes to another host, the original hypervisor that hosts the VM is aware of VM movement or location changing but may not be able to keep track of the new location after the VM moves. Therefore one external party that maintains the mapping between the VM's identity and the VM's current location is needed which keeps track VM movements.

3.2. Why VM migration

VM migration refers to VM movement within a virtual environment in response to events, conditions or based on requirements. The events or conditions could be, for example, very high workloads experienced by the VMs or upgrades of the physical server or virtualization server, load balancing between virtualization servers. The requirements could be, for example, low power and low cost requirements or service continuity requirement. When a VM is moved without service disruption, we usually call this VM movement as VM mobility. However it is very difficult to provide transparent VM mobility support since it not only needs to keep connection uninterrupted but also needs to move the whole VM image from one place to another place, which may take a long down time (e.g., more than 400 ms) and can be noticed by the end user.

Fortunately, VMs may be migrated without VM mobility support. For example, a server manager or administrator can move a running virtual machine or application between different physical machines without disconnecting the client or application if the client or application

supports VM suspending and resuming operation or stopped at the source before the movement and restart at the destination after movement.

In some case when VM mobility is really needed, it is recommended that one copy of VM SHOULD be replicated from the source to the destination and during VM replication, thus the VM running on the source should not be affected. When VM replication to the destination completes and the VM on the destination restarts, the VM on the source can be stopped. However how the VM on the destination coordinates with the VM on the source to know whom the latter is communicating with is a challenging issue.

3.3. Who manages VM

To ensure the quality of applications (e.g., real-time applications) or provide the same service level agreement, the VM's state(i.e., the network attributes and policies associated with the VM) should be moved with the VM as well when the VM moves across participating virtualization hosts (e.g., virtualization server or hypervisor). These network attributes associated with VM should be enforced on the physical switch and the virtual switch corresponding to VM to avoid security and access threats. The hypervisor or the virtualization server may maintain the network attributes for each VM. However when VMs are moved from the previous server to the new server, the old server and the new server may have no means to find each other. Therefore one external party called VM network management system (e.g., Cloud Broker) is needed and should get involved to coordinate between the old server and the new server to establish the association between network attributes/policies and the VM's identity. If the VM management system does not span across data center and the VM is moved between data centers, the VM network management system in one data center may also need to coordinate with VM network management system in another data center.

3.4. VM Grouping

VM grouping significantly simplifies the administration tasks when managing large numbers of virtual machines, as new VMs are simply added to existing groups. With grouping, similar VMs can be grouped together and assigned with the same networking policies to all members of the group to ensure consistent allocation of resources and security measures to meet service level goals. Members of the group retain the group attributes wherever they are located or move within the virtual environment, providing a basis for dynamic policy assignments. VM groups can be maintained or distributed on the virtualization server or can be maintained on a centralized place, e.g., the VM management platform that manages all the virtualization

servers in the data center. VM groups maintained on each virtualization server may change at any time due to various VM operations (e.g., VM adding, VM removing, VM moving). Therefore VM groups need to be synchronized with the central VM management platform. Profiles containing network configurations such as VLAN, traffic shaping and ACLs for VM groups can be automatically synchronized to the central VM management platform as well. This way, consistent network policies can be enforced regardless of the VM's location.

3.5. What VM information should be managed

The ability to identify VMs within the physical hosts is very important. With the ability to identify each VM uniquely, the administrator can apply the same philosophy to VMs as used with physical servers. VLAN and QoS settings can be provisioned and ACL attributes can be set at a VM level with permit and deny actions based on layer 2 to layer 4 information. In the VM environment, a VM is usually identified by MAC or IP address and belongs to one tenant. Typically, one tenant may possess of one VM or a group of VMs in one virtual network or several groups of VMs distributed in multiple virtual networks. On the request of the tenant, a VM can be added, removed and moved by the virtualization server or the hypervisor. When the VM moves, the network attributes or configuration attributes associated with the VM should also be moved with the VM as well to ensure that the service level agreement and response times are met. These network attributes include access and tunnel policies and (L2 and/or L3) forwarding functions and should be moved with the VM information. We use Virtual Network Instance ID to represent those network attributes. One tenant has at least one Virtual Network ID. Therefore each tenant should at least include the following information:

- o vCenter Name or Identifier
- o vServer Name or Identifier
- o Virtual Network ID (VNID)
- o VLAN tag value
- o VM Group ID
- o VM MAC/IP address

Note that Tenant may have tenant ID which could be combination of these information.

3.6. Who Triggers or Controls VM Movements

VM can be moved within the virtual environment in response to events or conditions. An issue here is who triggers and controls VM movement? In a small scale or large scale data center, the server administrator is usually not aware of VM movement and may respond quickly to system fault or server overload and move a virtual machine or a group of VMs to different physical machines. However it is hard for the server administrator to response to dynamic VM movement and creation since he doesn't keep track of VM movements.

In large scale data centers, the server administrator may be more hesitated to utilize VM movements because of the time demands of managing the related networking requirements. Therefore automated solutions that safely create and move virtual machines and free VM network or Server administrators from their responsibilities is highly required.

The external party (i.e., the control or management plane function) is needed to play the role of server administrator and should support keeping track of VM movement and response quickly to dynamic VM creation and movement.

When one tenant moves from one place to another place, VM movement associated tenant should be informed to the control or management plane function. When one tenant requests to improve the quality of application and shorten the response time, the control or management function can trigger VM being moved to the server that is closer to the user.

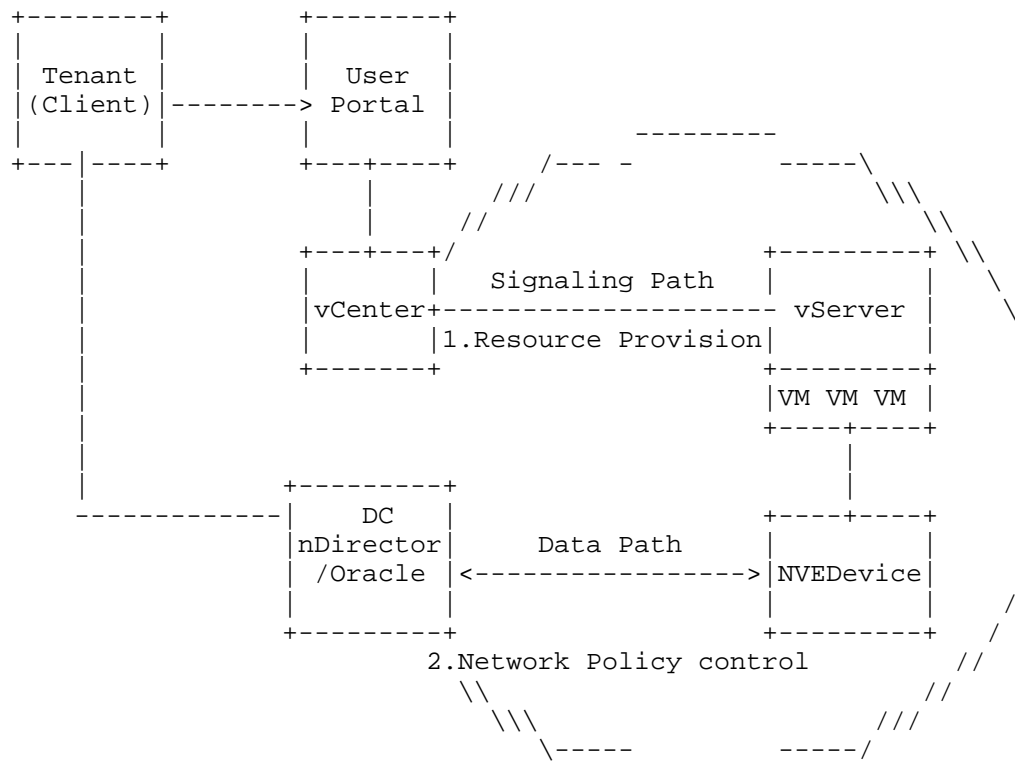
3.7. VM Monitoring

In order to sort out bad VMs, VM monitoring is very important. The VM monitor mechanism keeps track of the availability of VMs and their resource entitlements and utilization, e.g., CPU utilization, Disk and memory utilization, network utilization, network storage utilization,. It ensures that there is no overloading of resources whereby many service requests cannot be simultaneously fulfilled due to limited resource available. VM monitor is also useful for server administrations and report the status information of VMs or VM groups in each server to the VM management and provision system.

4. Use Cases

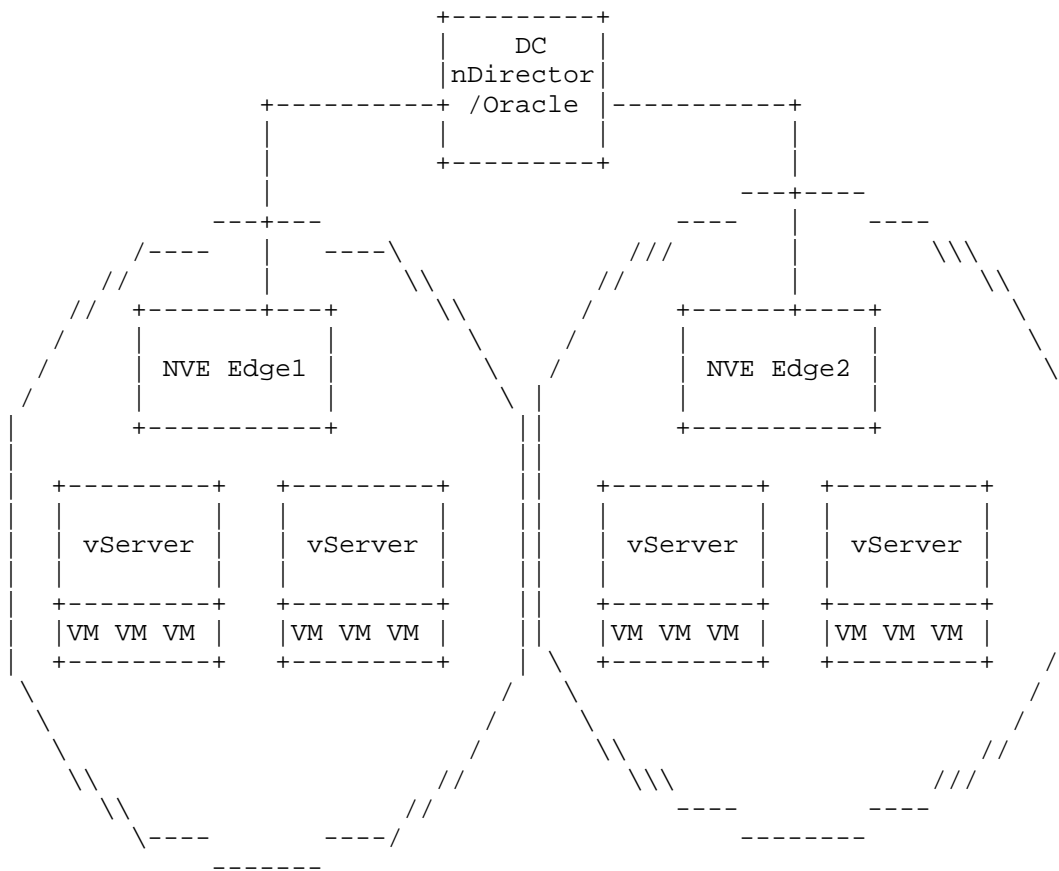
4.1. On Demand Network Provision Automation within the data center

The Tenant Alice is logging into user portal via her laptop and request playing Cloud gaming using VM she has already rented, the request is redirected to the provision system vCenter, the vCenter retrieves service configuration information and locate which vServer the VM belongs to and then Provision resources for VM running on that vServer. The vServer signals the VM operation parameter update to the NVE to which the VM is connecting. In turn, the NVE device interacts with the DC nDirector to configure policy and populate the forwarding table to each network element (e.g., ToR, DC GW), in the path from the Tenant End System to the NVE Device. In addition, the DC nDirector may also populate the mapping table to map the destination address (either L2 or L3) of a packet received from a VM into the corresponding destination address of the NVE device.



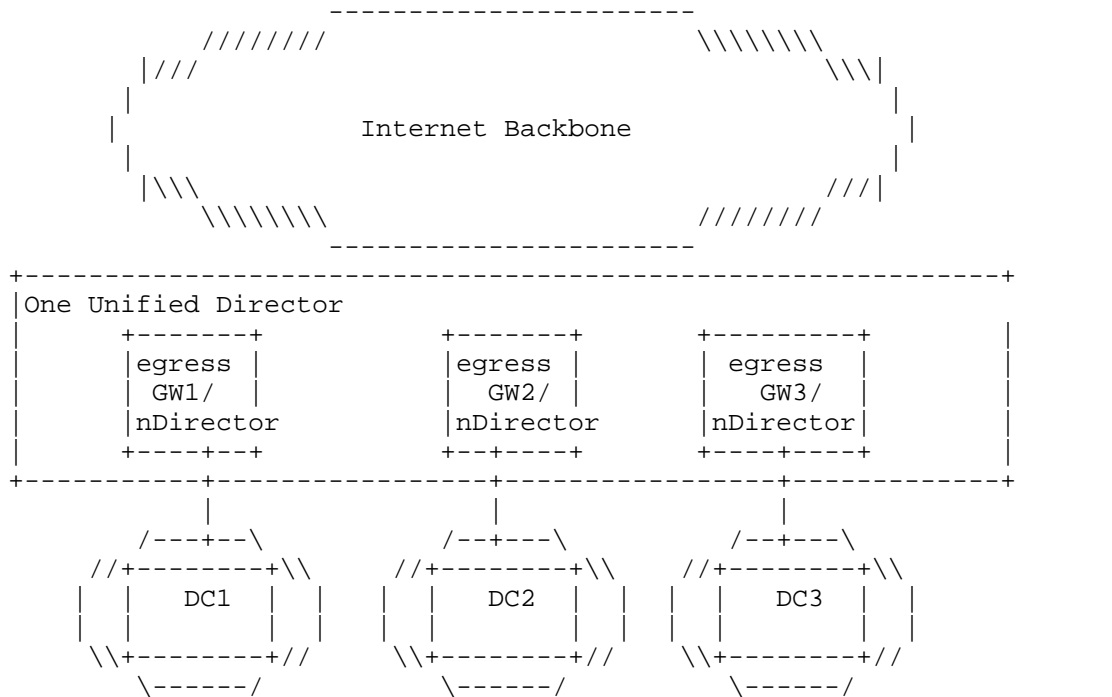
4.2. Large inter-data centers Layer 2 interconnection and data forwarding

When the tenant Alice using VM1 in data center1 communicates with tenant Bob using VM2 in data center2, the VM1 should already know layer2 identity of VM2, however the VM1 may not know which NVE Edge the VM2 is placed behind, in order to learn the location of the remote NVE Device associated with VM2, the mapping table is needed on the local NVE Device associated with VM1 which is used to map the final destination(i.e., the identity of VM2) to the destination address of the remote NVE device associated with VM2. In order to realize this, the nDirector should tell the local NVE device associated with VM1 about layer 3 location identifier of remote NVE device associated with VM2 and establish mapping between layer 2 VM2 identity and layer 3 identity of the remote NVE Edge associated with VM2. In addition, the nDirector may tell all the remote NVE devices associated with the VM which the VM1 is communicating with to establish the mapping between layer 2 VM1 identity and layer 3 identity of the local NVE Device associated with VM1. When this is done, the data packet from VM1 can be sent to the NVE device associated with VM1, the NVE Device associated with VM1 can identify layer 2 frame targeted for remote destination based on established mapping table, encapsulates it into IP packet and transmit it across layer 3 network. After the packet arrives at the remote NVE Edge, the remote NVE Edge device decapsulates layer 3 packet, take out layer 2 frame and forward it to ultimate destination VM2.



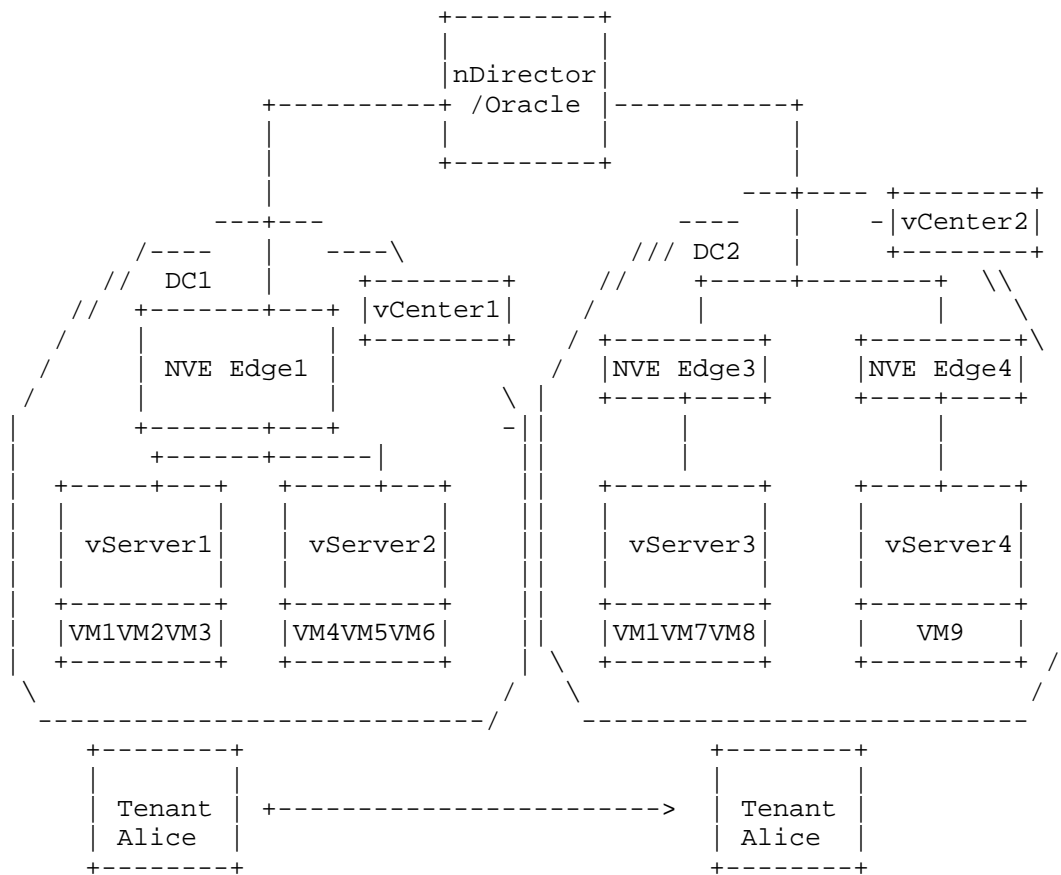
4.3. Enable multiple data centers present as one

In order to support more data centers interconnection and enable more efficient use of resources in each data center, multiple data centers may closely coordinate with each other to better load balancing capability and work as one large DC with the involvement of the nDirector that manages DCs, e.g., DC nDirector in each data center may coordinate with each other and form one common control plane.



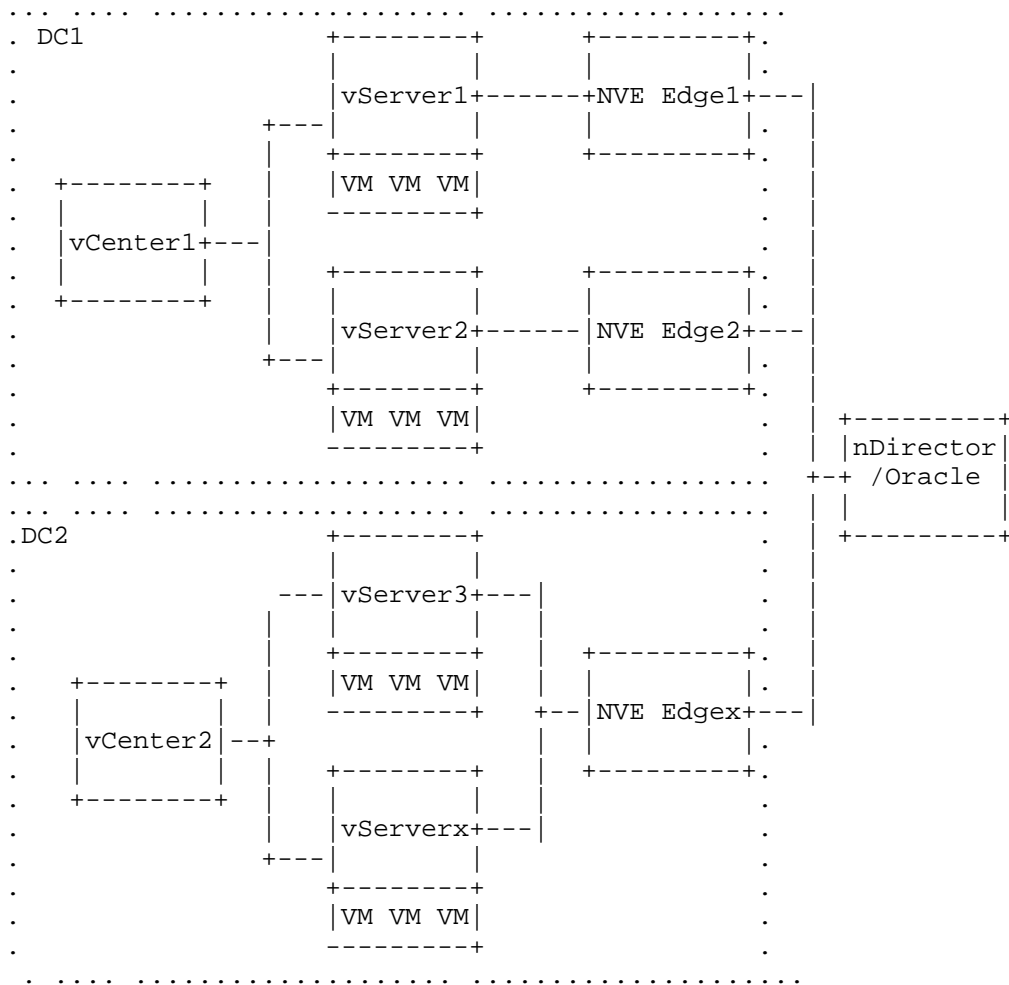
4.4. VM migration and mobility across data centers

The Tenant Alice is using VM1 in data center 1 to communicate with the tenant Bob who is using VM9 in data center 2. For business reason, the tenant Alice travels to the Bob's city where the data center 2 situates but still use VM1 in the data center 1 to communicate with the tenant Bob. In order to provide better user experience, the VM1 may be move from vServer 1 to the new vServer3 in the data center 2 which is more close to where the tenant Alice is located. The vCenter can get involved to interact with data center 1 and data center2 and help replicate and relocate VM1 to the new location. In the meanwhile ,when VM movement is done, the NVE device connecting to VM1 and associated with vServer 3 should interact with the nDirector to update mapping table maintained in the nDirector by the new NVE device location associated with VM1. In turn, the nDirector should update the mapping tables in all the NVE device associated with the VM which VM1 is communicating with.



5. General Network Virtualization Architecture

When Multiple virtual machines (VMs) created in one vServer, VM can be managed under this vServer. However vServer can not be isolated node since VM can be moved from one to another vServer under the same or different data center which is beyond the control of the vServer who create that VM. We envision the Network virtualization architecture to consist of vServers (virtualization servers), nDirector and vCenters (the aforementioned VM and vServer management platform) and NVE Edges. The vCenter is placed on the management plane within each data center and can be used to manage a large number of vServers in each data center. The vServer is connecting to NVE Edge in its own data center either directly or via a switched network (typically Ethernet). The nDirector is placed on the control plane and manage one or multiple data centers. When the nDirector manages multiple data centers, the nDirector should interact with all the NVE Edges in each data center to facilitate large inter-data center Layer 2 interconnection, VM migration and mobility across data centers and enabling multiple data centers work and present as one.



Network Virtualization Architecture

5.1. NVE (Network Virtualization Edge Function)

As defined in section 1.2 of [I.D-ietf-nvo3-framework], it is a network entity that sits on the edge of the NVO3 network and could be implemented as part of a virtual switch within a hypervisor, a physical switch or router, a Network Service Appliance (e.g., NAT/FW). When VM1 connecting to one NVE Edge want to communicate with the other VMs which are connecting to the other NVE Edges, the NVE Edge associated with VM1 should distribute the mapping between layer 2 identity of VM1 and NVE Edge associated with VM1 by the nDirector to all the NVE Edges associated with VMs which VM1 is communicating

with. In addition, the NVE Edge associated with VM1 either interact with the nDirector or learn from the other NVE Edges who is distributing mapping table through the nDirector to build mapping table between layer 2 identity of VMs which VM1 is communicating with the NVE Edge associated with VMs which VM1 is communicating with and based on such mapping table to forward the data packets.

5.2. vServer (virtualization Server)

The vServer is served as a platform for running virtual machines and is installed on the physical hardware in a virtualized environment and provide physical hardware resource dynamically to the virtual machines as needed. It is also referred to as "the virtualization server" or hypervisor. It may get instructions from provision systems (i.e., vCenters) to create, modify, terminate VM for each tenant. It may also interact with the NVE Edge to inform the NVE about the map or association between vserver, virtual machine and network connection. This interaction can also be used to release association between vServer and the NVE Edge.

5.3. vCenter (management plane function)

The vCenter is served as a platform for managing in one data center not only assignment of virtual machines to the vServer but also assignment of resources to the virtual machines and provide a single control point to the data center. It unifies the resources from individual vServer to be shared among virtual machines in the entire data center. It may interact with vServer to allocate virtual machines to the vServer and monitor performance of each vServer and each VM in the data center. The vCenter may maintain the map from vServer to Network connection which contain not only vServer configurations such as vServer name, vServer IP address port number but also VM configurations for each tenant end system associated with that vServer. When vCenter hierarchy is used, the root vCenter who has global view may interact with the child vCenter to decide which child vCenter is responsible for assigning the virtual machine to which vServer based on topological information and resource utilization information in each data center and local policy information.

5.4. nDirector (Control plane function)

The nDirector is implemented as part of DC Gateway and sits on top of the vCenter in each data center and is served as orchestrator layer to allow layer 2 interconnection and forwarding between data centers and enable multiple data centers to present as one. The nDirector may interact with the NVE Edge to populate forwarding table in the path from the NVE Edge Device to the Tenant End System and react to

the NVE request to assign network attributes such as VLAN,ACL, QoS parameters on all the network elements in the path from NVE device to the Tenant End System and manipulates the QoS control information in the path between the NVE Edges associated with VMs in communication. In addition, the nDirector may distribute mapping table between layer 2 identity of VM and the NVE Edge associate with that VM to all the other NVE Edges and maintain such mapping table in the nDirector.

6. vServer to vCenter management interface

6.1. VM Creation

vCenter requests vServer to create a new virtual machine and allocate the resource for its execution.

6.2. VM Termination

vCenter requests vServer to delete a virtual machine and clean up the underlying resources for that virtual machine.

6.3. VM Registration

When a VM is created for one tenant in the vServer, the vServer may create VM profile for this tenant containing VM identity, VNID, port, VID and registers the VM configuration associated with this tenant to the vCenter. Upon receiving such a registration request, vCenter should check if it has already established VM profile for the corresponding tenant: if yes, vCenter should update the existing VM profile for that tenant, otherwise vCenter should create a new VM profile for that tenant.

6.4. VM Unregistration

When a VM is removed for one tenant from the vServer, the vServer may remove VM profile for this tenant containing VM identity, VNID, port, VID and deregisters the VM configuration associated with that tenant to the vCenter. Upon receiving such a deregistration request, vCenter should check if it has already established VM profile for that tenant: if yes, vCenter should remove the existing VM profile for that tenant, otherwise other vCenter should report alert to the vServer.

6.5. VM Bulk Registration

When a large number of VMs are created in one vServer and share the same template, the vServer may create a profile for a group of these VMs and send a bulk registration request containing the group identifier and associated VM profile to the vCenter. Upon receiving such a bulk registration request, vCenter should create or update the profile for a group of these VMs.

6.6. VM Bulk Unregistration

When a large number of VMs are removed in one vServer and share the same template, the vServer may remove a profile for a group of these VMs and send a bulk unregistration request containing the group

identifier and associated VM profile to the vCenter. Upon receiving such a bulk registration request, vCenter should remove the profile for a group of these VMs.

6.7. VM Configuration Modification

vCenter requests vServer to update a virtual machine and reallocate the resource for its execution.

6.8. VM Profile Lookup/Discovery

When a VM1 in one vServer want to communicate with one VM2 in another vServer, the client associated with VM1 should check with vCenter based on VM2 identity to see if the profile for that VM2 already exists and which server maintains that VM configuration. If yes, vCenter should reply to the the client with the address or name of the vServer which the VM2 is situated in.

6.9. VM Relocation

When vCenter is triggered to move one VM or a group of VMs from one source vServer to another destination vServer, the vCenter should send a VM relocation request to both vServers and updates its profile to indicate the new vServer that maintains the VM configuration for that VM or a group of those VMs. The relocation request will trigger the VM image to be moved from the source vServer to the destination vServer.

6.10. VM Replication

One tenant moves between vServers or between data centers and may, as the internet user, want to access applications via the VM without service disruption. In order to achieve this, he can choose to access applications via the same VM without moving the VM when he moves. However, the VM he is using may be far away from where he stays. In order to provide better user experience, the tenant may request vCenter through the nDirector to move VM to the vServer that is more close to where he stays and keeps the service uninterrupted. In such case, the vCenter may interact with the vServer that hosts the original VM to chooses one vServer that is closer to the tenant and moves one copy of the VM image to the destination vServer.

6.11. VM Report

When one VM is created, moved, added, removed from the vServer, the VM monitor should be enabled to report the status information and resource availability of that VM to the vCenter. In this case, vCenter can know which server is overloaded, which server is unused

or least used.

7. nDirector to NVE Edge control interface

Signaling between the nDirector and NVE Device can be used to do three things:

- Enforce the network policy for each VM in the path from the NVE Edge associated with VM to the Tenant End System.

- Populate forwarding table in the path from the NVE Edge associated with VM to the Tenant End System in the data center.

- Populate mapping table in each NVE Edge that is in the virtual network across data centers under the control of the nDirector.

One could reuse existing protocols, among them NetConf, SNMP, RSVP, Radius, Diameter, to signal the messages between nDirector and NVE Edges. The nDirector need to know which NVE Edges belong to the same virtual network and then the distribute the routes between these NVE Edges to each NVE Edges belonging to the same virtual network. In additional the nDirector may interact with the NVE Edge and the associated overlay network in the data center in response to the provision request from the NVE Edge and populate forwarding table to the associated overlay Network elements in the data path from the Tenant End System to the NVE Edge and install network policy to the network elements in the data path between the Tenant End System and the NVE Edge. For details of Signaling control/forward plane information between network virtualization edges (NVEs) , please see [I.D-wu-nvo3-nve2nve].

8. vServer to NVE Edge control interface

Signaling between vServer and NVE Edge is used to establish mapping between the vServer who host VM and network connection which VM relies on. For more details signaling and operation, please see relevant NVO3 draft.

9. Security Considerations

Threats may arise when VMs move into a hostile VM environment, e.g., when the VM identity is exploited by adversaries to launch denial of service or Phishing attacks[Phishing]. Further details are to be explored in the future version of this document.

10. IANA Considerations

This document has no actions for IANA.

11. Contributors

Thank Xiaoming Fu for helping provide input to the initial draft of this document.

12. References

12.1. Normative References

- [I.D-ietf-nvo3-framework]
Lasserre, M., "Framework for DC Network Virtualization",
ID draft-ietf-nvo3-framework-00, September 2012.
- [I.D-wu-nvo3-nve2nve]
Wu, Q., "Signaling control/forward plane information
between network virtualization edges (NVEs)",
ID draft-wu-nvo3-nve2nve-00, 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", March 1997.

12.2. Informative References

- [I.D-kompella-nvo3-server2nve]
Kompella, K., "Using Signaling to Simplify Network
Virtualization Provisioning",
ID draft-kompella-nvo3-server2nve, July 2012.
- [Phishing]
"http://kea.hubpages.com/hub/What-is-Phishing".

Authors' Addresses

Roland Schott
Deutsche Telekom Laboratories
Deutsche-Telekom-Allee 7
Darmstadt 64295
Germany

Email: Roland.Schott@telekom.de

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: sunseawq@huawei.com

INTERNET-DRAFT
Intended Status: Informational
Expires: May 29, 2013

A. Ghanwani
Dell
November 30, 2012

Multicast Issues in Networks Using NVO3
draft-ghanwani-nvo3-mcast-issues-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/lid-abstracts.html>

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

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract

This memo discusses issues with supporting multicast traffic in a network that uses Network Virtualization using Overlays over Layer 3. It lists the various mechanisms that may be used for multicast and describes some of the considerations.

Table of Contents

1	Introduction	3
2	Multicast mechanisms in networks that use NVO3	3
2.1	No multicast support	3
2.2	Replication at the source NVE	4
2.3	Replication at a multicast service node	4
2.4	IP multicast in the underlay	5
2.5	Simultaneous use of more than one mechanism	6
3	Summary	6
4	Security Considerations	6
5	IANA Considerations	6
6	References	6
6.1	Normative References	6
6.2	Informative References	7
	Authors' Addresses	7

1 Introduction

Network virtualization using Overlays over Layer 3 (NVO3) is a technology that is used to address issues that arise in building large, multitenant data centers that make extensive use of server virtualization [PS].

This document is focused specifically on the problem of supporting multicast in networks that use NVO3. Because of the requirement of multi-destination delivery, multicast traffic poses some unique challenges.

The reader is assumed to be familiar with the terminology as defined in the NVO3 Framework document [FW].

2. Multicast mechanisms in networks that use NVO3

In NVO3 environments, traffic between NVEs is transported using a tunnel encapsulation such as VXLAN [VXLAN], NVGRE [NVGRE], STT [STT], etc.

Besides the need to support the Address Resolution Protocol (ARP) and Neighbor Discovery (ND), there are several applications that require the support of multicast and/or broadcast in data centers [DC-MC]. With NVO3, there are four possible ways that multicast may be handled in such networks.

1. No multicast support.
2. Replication at the source NVE.
3. Replication at a multicast service node.
4. IP Multicast in the underlay.

These mechanisms are briefly mentioned in the NVO3 Framework [FW] document. This document attempts to fill in some more details about the basic mechanisms underlying each of these mechanisms and discusses the issues and tradeoffs of each.

2.1 No multicast support

In this scenario, there is no support whatsoever for multicast traffic when using the overlay. This can only work if the following conditions are met:

1. All of the traffic is unicast.
2. An oracle is used at the NVE to determine the MAC address-to-NVE mapping and to determine the MAC address-to-IP address bindings. In other words, there is no data plane learning, and address resolution

requests via ARP/ND that are issued by the VMs must be resolved by the NVE that they are attached to.

With this approach, certain multicast/broadcast applications such as DHCP can be supported by use of a helper function in the NVE.

The main issues that need to be addressed with this mechanism are the handling of hosts for which a mapping does not already exist in the oracle. This issue can be particularly challenging if such end systems are reachable through more than one NVE.

2.2 Replication at the source NVE

With this method, the overlay attempts to provide a multicast service without requiring any specific support from the underlay, other than that of a unicast service. A multicast or broadcast transmission is achieved by replicating the packet at the source NVE, and making copies, one for each destination NVE that the multicast packet must be sent to.

For this mechanism to work, the source NVE must know, a priori, the IP addresses of all destination NVEs that need to receive the packet.

For example, in the case of an ARP broadcast or an ND multicast, the source NVE must know the IP addresses of all the remote NVEs where there are members of the tenant subnet in question.

The obvious drawback with this method is that we have multiple copies of the same packet that will traverse any common links that are along the path to each of the destination NVEs. If, for example, a tenant subnet is spread across 50 NVEs, the packet would have to be replicated 50 times at the source NVE. This also creates an issue with the forwarding performance of the NVE, especially if it is implemented in software.

Note that this method is similar to what was used in VPLS [VPLS] prior to extensive support of MPLS multicast [MPLS-MC].

2.3 Replication at a multicast service node

With this method, all multicast packets would be sent using a unicast tunnel encapsulation to a multicast service node. The multicast service node, in turn, would create multiple copies of the packet and would deliver a copy, using a unicast tunnel encapsulation, to each of the NVEs that are part of the multicast group for which the packet is intended.

This mechanism is similar to that used by the ATM Forum's LAN Emulation [LANE] specification [LANE].

Unlike the method described in Section 2.2, there is no performance impact at the ingress NVE, nor are there any issues with multiple copies of the same packet from the source NVE to the multicast service node. However there remain issues with multiple copies of the same packet on links that are common to the paths from the multicast service node to each of the egress NVEs. Additional issues that are introduced with this method include the availability of the multicast service node, methods to scale the services offered by the multicast service node, and the sub-optimality of the delivery paths.

Finally, the IP address of the source NVE must be preserved in packet copies created at the multicast service node if data plane learning is in use. This could create problems if IP source address reverse path forwarding (RPF) checks are in use.

2.4 IP multicast in the underlay

In this method, the underlay supports IP multicast and the ingress NVE encapsulates the packet with the appropriate IP multicast address in the tunnel encapsulation header for delivery to the desired set of NVEs. The protocol in the underlay could be any variant of Protocol Independent Multicast (PIM).

With this method, there are none of the issues with the methods described in Sections 2.2.

With PIM Sparse Mode (PIM-SM), the number of flows required would be $(n \cdot g)$, where n is the number of source NVEs that source packets for the group, and g is the number of groups. Bidirectional PIM (BIDIR-PIM) would offer better scalability with the number of flows required being g .

In the absence of any additional mechanism, e.g. using an oracle for address resolution, for optimal delivery, there would have to be a separate group for each tenant, plus a separate group for each multicast address (used for multicast applications) within a tenant. Additional considerations are that only the lower 23 bits of the IP address (regardless of whether IPv4 or IPv6 is in use) are mapped to the outer MAC address, and if there is equipment that prunes multicasts at Layer 2, there will be some aliasing. Finally, a mechanism to efficiently provision such addresses for each group would be required.

There are additional optimizations which are possible, but they come with their own restrictions. For example, a set of tenants may be restricted to some subset of NVEs and they could all share the same outer IP multicast group address. This however introduces a problem of sub-optimal delivery (even if a particular tenant within the group

of tenants doesn't have a presence on one of the NVEs which another one does, the former's multicast packets would still be delivered to that NVE). It also introduces an additional network management burden to optimize which tenants should be part of the same tenant group (based on the NVEs they share), which somewhat dilutes the value proposition of NVO3 which is to completely decouple the overlay and physical network design allowing complete freedom of placement of VMs anywhere within the data center.

2.5 Simultaneous use of more than one mechanism

While the mechanisms discussed in the previous section have been discussed individually, it is possible for implementations to rely on more than one of these. For example, the method of Section 2.1 could be used for minimizing ARP/ND, while at the same time, multicast applications may be supported by one, or a combination of, the other methods. For small multicast groups, the methods of source NVE replication or the use of a multicast service node may be attractive, while for larger multicast groups, the use of multicast in the underlay may be preferable.

3 Summary

This document has identified various mechanisms for supporting multicast in networks that use NVO3. It highlights the basics of each mechanism and some of the issues with them. As solutions are developed, the protocols would need to consider the use of these mechanisms and co-existence may be a consideration.

4 Security Considerations

This is an informational document, and as such, does not introduce any new security considerations beyond what may be present in proposed solutions.

5 IANA Considerations

This draft does not have any IANA considerations.

6 References

6.1 Normative References

- [PS] Lasserre, M. et al., "Framework for DC network virtualization", work in progress.
- [FW] Narten, T. et al., "Problem statement: Overlays for network virtualization", work in progress.

6.2 Informative References

- [VXLAN] Mahalingam, M. et al., "VXLAN: A framework for overlaying virtualized Layer 2 networks over Layer 3 networks", work in progress.
- [NVGRE] Sridharan, M. et al., "NVGRE: Network virtualization using Generic Routing Encapsulation", work in progress.
- [STT] Davie, B. and Gross J., "A stateless transport tunneling protocol for network virtualization", work in progress.
- [DC-MC] McBride M., and Lui, H., "Multicast in the data center overview", work in progress.
- [VPLS] Lasserre, M., and Kompella, V. (Eds), "Virtual Private LAN Service (VPLS) using Label Distribution Protocol (LDP) signaling", RFC 4762, January 2007.
- [MPLS-MC] Aggarwal, R. et al., "Multicast in VPLS", work in progress.
- [LANE] "LAN emulation over ATM", The ATM Forum, af-lane-0021.000, January 1995.

Authors' Addresses

Anoop Ghanwani
Dell
350 Holger Way
San Jose, CA 95134

Phone: +1-408-571-3228
Email: anoop@alumni.duke.edu

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-cp]. 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.

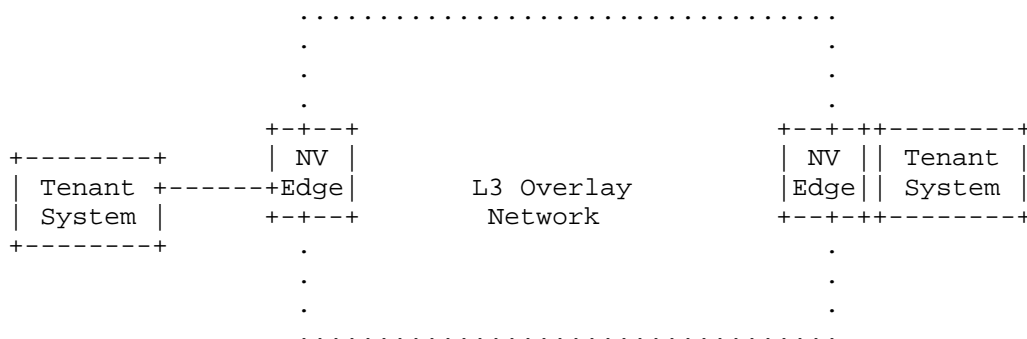
Virtual Network (VN): This is a virtual L2 or L3 domain that belongs to a tenant.

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:

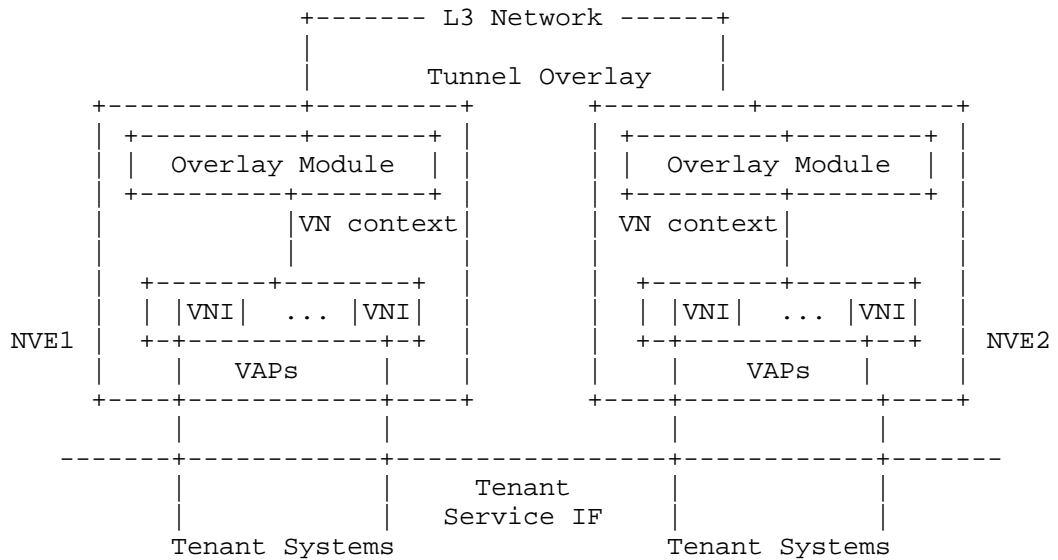


Figure 1 : Generic reference model for NV Edge

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:

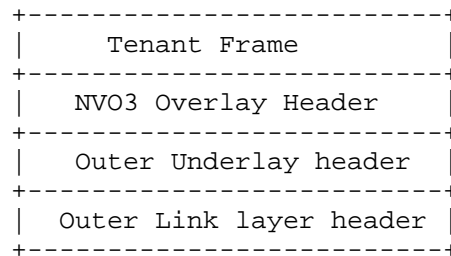


Figure 2 : NVO3 encapsulated frame

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:

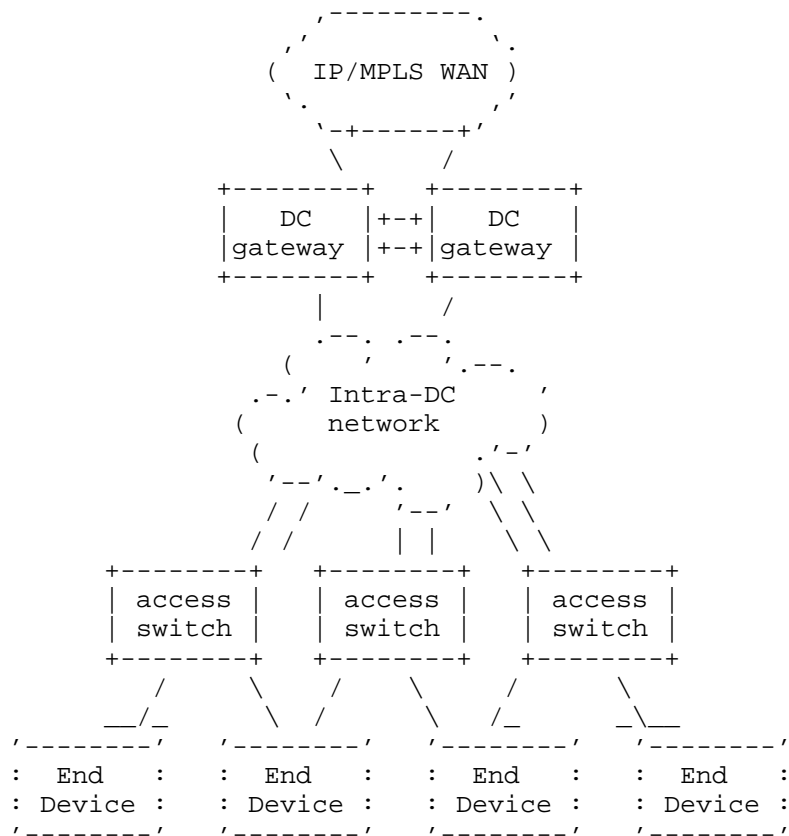


Figure 1 : A Generic Architecture for Data Centers

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.

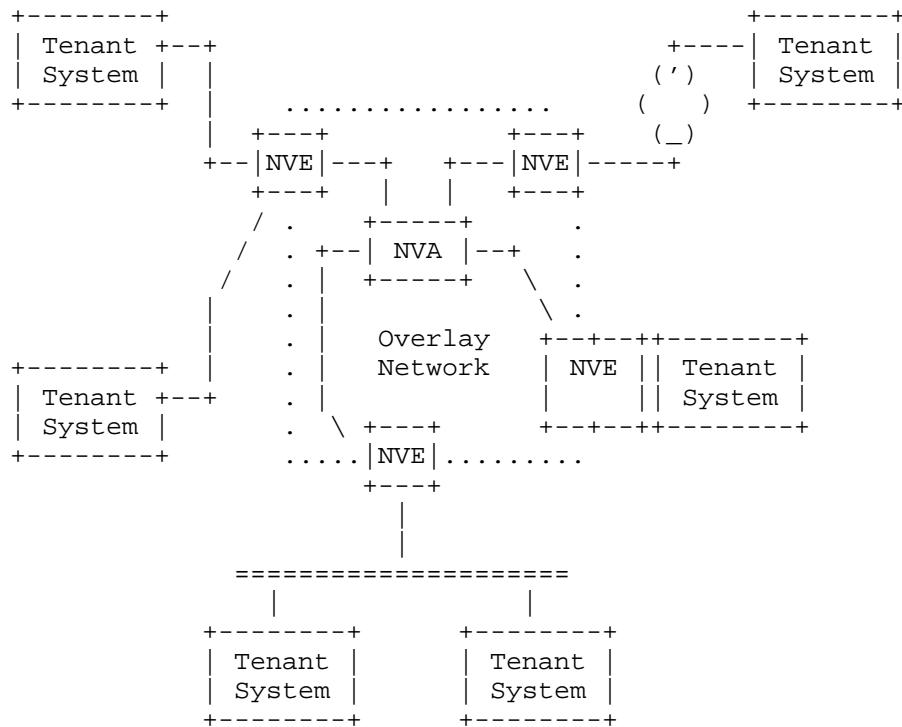


Figure 2 : Generic reference model for DC network virtualization overlay

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.).

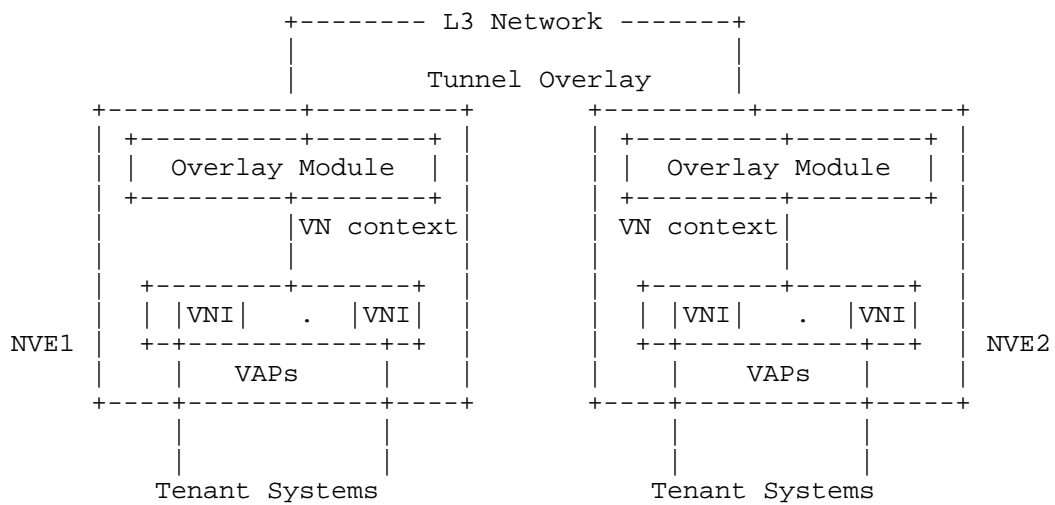


Figure 3 : Generic NVE reference model

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
Huawei
M. Toy
Comcast
A. Isaac
Bloomberg
V. Manral
Hewlett-Packard
L. Dunbar
Huawei

Expires: August 2013

February 15, 2013

Use Cases for DC Network Virtualization Overlays

draft-ietf-nvo3-use-case-00

Abstract

This draft describes the general NVO3 use cases. The work intention is to help validate the NVO3 framework and requirements as along with the development of the solutions.

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 August, 2013.

Copyright Notice

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

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

Conventions used in this document

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

Table of Contents

1. Introduction.....	3
2. Terminology.....	4
3. Basic Virtual Networks in a Data Center.....	4
4. Interconnecting DC Virtual Network and External Networks.....	6
4.1. DC Virtual Network Access via Internet.....	7
4.2. DC Virtual Network and WAN VPN Interconnection.....	7
5. DC Applications Using NVO3.....	9
5.1. Supporting Multi Technologies in a Data Center.....	10
5.2. Tenant Virtual Network with Bridging/Routing.....	10
5.3. Virtual Data Center (VDC).....	11
5.4. Federating NV03 Domains.....	13
6. OAM Considerations.....	13
7. Summary.....	13
8. Security Considerations.....	14
9. IANA Considerations.....	14
10. Acknowledgements.....	15
11. References.....	15
11.1. Normative References.....	15
11.2. Informative References.....	16
Authors' Addresses.....	16

1. Introduction

Compute Virtualization has dramatically and quickly changed IT industry in terms of efficiency, cost, and the speed in providing a new applications and/or services. However the problems in today's data center hinder the support of an elastic cloud service and dynamic virtual tenant networks [NVO3PRBM]. The goal of DC Network Virtualization Overlays, i.e. NVO3, is to decouple tenant system communication networking from DC physical networks and to allow one physical network infrastructure to provide: 1) traffic isolation among virtual networks over the same physical network; 2) independent address space in each virtual network and address isolation from the infrastructure's; 3) Flexible VM placement and move from one server to another without any physical network limitation. These characteristics will help address the issues in the data centers [NVO3PRBM].

Although NVO3 may enable a true virtual environment where VMs and network service appliances communicate, the NVO3 solution has to address the communication between a virtual network and one physical network. This is because 1) many traditional DCs exist and will not disappear any time soon; 2) a lot of DC applications serve to Internet and/or cooperation users on physical networks; 3) some applications like Big Data analytics which are CPU bound may not want the virtualization capability.

This document is to describe general NVO3 use cases that apply to various data center networks to ensure nvo3 framework and solutions can meet the demands. Three types of the use cases described here are:

- o A virtual network connects many tenant systems within a Data Center and form one L2 or L3 communication domain. A virtual network segregates its traffic from others and allows the VMs in the network moving from one server to another. The case may be used for DC internal applications that constitute the DC East-West traffic.
- o A DC provider offers a secure DC service to an enterprise customer and/or Internet users. In these cases, the enterprise customer may use a traditional VPN provided by a carrier or an IPsec tunnel over Internet connecting to an overlay virtual network offered by a Data Center provider. This is mainly constitutes DC North-South traffic.

- o A DC provider uses NVO3 to design a variety of cloud applications that make use of the network service appliance, virtual compute, storage, and networking. In this case, the NVO3 provides the virtual networking functions for the applications.

The document uses the architecture reference model and terminologies defined in [NVO3FRWK] to describe the use cases.

2. Terminology

This document uses the terminologies defined in [NVO3FRWK], [RFC4364]. Some additional terms used in the document are listed here.

CUG: Closed User Group

L2 VNI: L2 Virtual Network Instance

L3 VNI: L3 Virtual Network Instance

ARP: Address Resolution Protocol

CPE: Customer Premise Equipment

DNS: Domain Name Service

DMZ: DeMilitarized Zone

NAT: Network Address Translation

VNIF: Virtual Network Interconnection Interface

3. Basic Virtual Networks in a Data Center

A virtual network may exist within a DC. The network enables a communication among tenant systems (TSs) that are in a Closed User Group (CUG). A TS may be a physical server or virtual machine (VM) on a server. A virtual network has a unique virtual network identifier (may be local or global unique) for switches/routers to properly differentiate it from other virtual networks. The CUGs are formed so that proper policies can be applied when the TSs in one CUG communicate with the TSs in other CUGs.

Figure 1 depicts this case by using the framework model.[NVO3FRWK] NVE1 and NVE2 are two network virtual edges and each may exist on a server or ToR. Each NVE may be the member of one or more virtual networks. Each virtual network may be L2 or L3 basis. In this

illustration, three virtual networks with VN context Ta, Tn, and Tm are shown. The VN 'Ta' terminates on both NVE1 and NVE2; The VN 'Tn' terminates on NVE1 and the VN 'Tm' at NVE2 only. If an NVE is a member of a VN, one or more virtual network instances (VNI) (i.e. routing and forwarding table) exist on the NVE. Each NVE has one overlay module to perform frame encapsulation/decapsulation and tunneling initiation/termination. In this scenario, a tunnel between NVE1 and NVE2 is necessary for the virtual network Ta.

A TS attaches to a virtual network (VN) via a virtual access point (VAP) on an NVE. One TS may participate in one or more virtual networks via VAPs; one NVE may be configured with multiple VAPs for a VN. Furthermore if individual virtual networks use different address spaces, the TS participating in all of them will be configured with multiple addresses as well. A TS as a gateway is an example for this. In addition, multiple TSs may use one VAP to attach to a VN. For example, VMs are on a server and NVE is on ToR, then some VMs may attach to NVE via a VLAN.

A VNI on an NVE is a routing and forwarding table that caches and/or maintains the mapping of a tenant system and its attached NVE. The table entry may be updated by the control plane, data plane, management plane, or the combination of them.

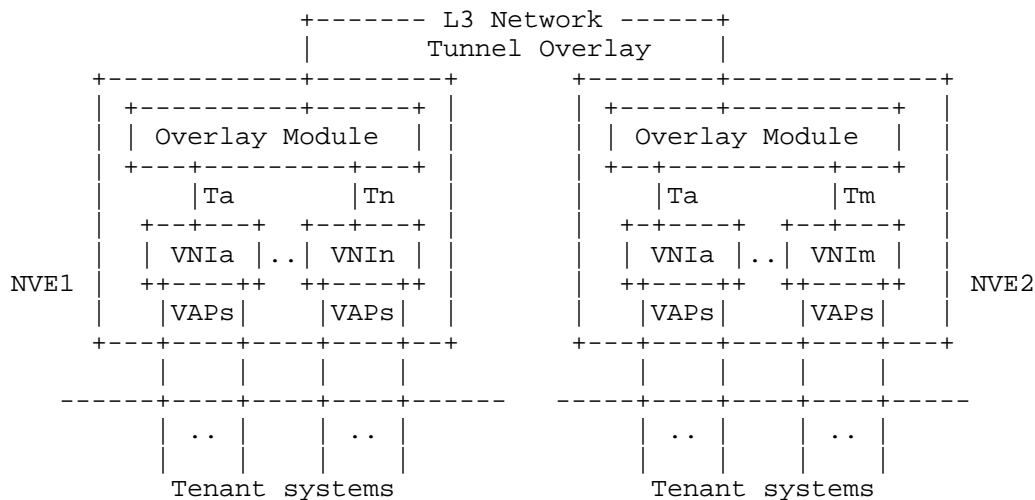


Figure 1 NVO3 for Tenant System Networking

One virtual network may have many NVE members and interconnect several thousands of TSs (as a matter of policy), the capability of supporting a lot of TSs per tenant instance and TS mobility is critical for NVO3 solution no matter where an NVE resides.

It is worth to mention two distinct cases here. The first is when TS and NVE are co-located on a same physical device, which means that the NVE is aware of the TS state at any time via internal API. The second is when TS and NVE are remotely connected, i.e. connected via a switched network or point-to-point link. In this case, a protocol is necessary for NVE to know TS state.

Note that if all NVEs are co-located with TSs in a CUG, the communication in the CUG is in a true virtual environment. If a TS connects to a NVE remotely, the communication from this TS to other TSs in the CUG is not in a true virtual environment. The packets to/from this TS are directly carried over a physical network, i.e. on the wire. This may require some necessary configuration on the physical network to facilitate the communication.

Individual virtual networks may use its own address space and the space is isolated from DC infrastructure. This eliminates the route reconfiguration in the DC underlying network when VMs move. Note that the NVO3 solutions still have to address VM move in the overlay network, i.e. the TS/NVE association change when a VM moves.

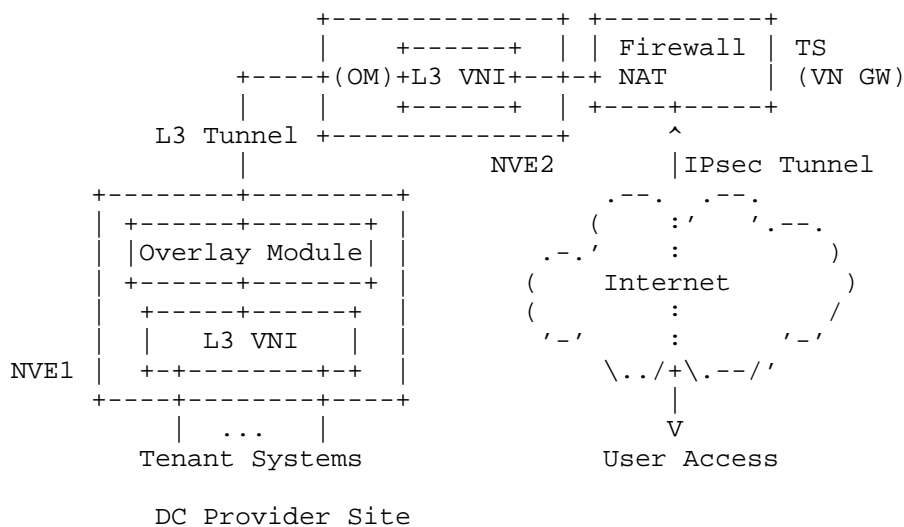
If a virtual network spans across multiple DC sites, one design is to allow the corresponding NVO3 instance seamlessly span across those sites without DC gateway routers' termination. In this case, the tunnel between a pair of NVEs may in turn be tunneled over other intermediate tunnels over the Internet or other WANs, or the intra DC and inter DC tunnels are stitched together to form an end-to-end tunnel between two NVEs in different DCs.

4. Interconnecting DC Virtual Network and External Networks

For customers (an enterprise or individuals) who want to utilize the DC provider's compute and storage resources to run their applications, they need to access their systems hosted in a DC through Carrier WANs or Internet. A DC provider may use an NVO3 virtual network for such customer to access their systems; then it, in turn, becomes the case of interconnecting DC virtual network and external networks. Two cases are described here.

4.1. DC Virtual Network Access via Internet

A user or an enterprise customer connects to a DC virtual network via Internet but securely. Figure 2 illustrates this case. An L3 virtual network is configured on NVE1 and NVE2 and two NVEs are connected via an L3 tunnel in the Data Center. A set of tenant systems attach to NVE1. The NVE2 connects to one (may be more) TS that runs the VN gateway and NAT applications (known as network service appliance). A user or customer can access their systems via Internet by using IPsec tunnel [RFC4301]. The encrypted tunnel is established between the VN GW and the user machine or CPE at enterprise location. The VN GW provides authentication scheme and encryption. Note that VN GW function may be performed by a network service appliance device or on a DC GW.



OM: Overlay Module;

Figure 2 DC Virtual Network Access via Internet

4.2. DC Virtual Network and WAN VPN Interconnection

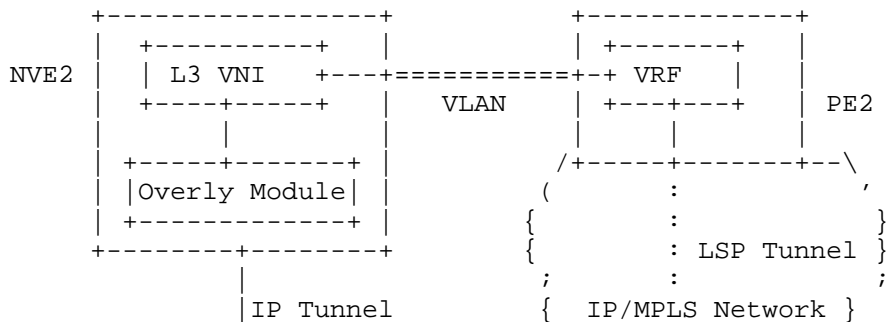
A DC Provider and Carrier may build a VN and VPN independently and interconnect the VN and VPN at the DC GW and PE for an enterprise customer. Figure 3 depicts this case in an L3 overlay (L2 overlay is the same). The DC provider constructs an L3 VN between the NVE1 on a server and the NVE2 on the DC GW in the DC site; the carrier

constructs an L3VPN between PE1 and PE2 in its IP/MPLS network. An Ethernet Interface physically connects the DC GW and PE2 devices. The local VLAN over the Ethernet interface [VRF-LITE] is configured to connect the L3VNI/NVE2 and VRF, which makes the interconnection between the L3 VN in the DC and the L3VPN in IP/MPLS network. An Ethernet Interface may be used between PE1 and CE to connect the L3VPN and enterprise physical networks.

This configuration allows the enterprise networks communicating to the tenant systems attached to the L3 VN without interfering with DC provider underlying physical networks and other overlay networks in the DC. The enterprise may use its own address space on the tenant systems attached to the L3 VN. The DC provider can manage the VMs and storage attached to the L3 VN for the enterprise customer. The enterprise customer can determine and run their applications on the VMs. From the L3 VN perspective, an end point in the enterprise location appears as the end point associating to the NVE2. The NVE2 on the DC GW has to perform both the GRE tunnel termination [RFC4797] and the local VLAN termination and forward the packets in between. The DC provider and Carrier negotiate the local VLAN ID used on the Ethernet interface.

This configuration makes the L3VPN over the WANs only has the reachability to the TS in the L3 VN. It does not have the reachability of DC physical networks and other VNs in the DC. However, the L3VPN has the reachability of enterprise networks. Note that both the DC provider and enterprise may have multiple network locations connecting to the L3VPN.

The eBGP protocol can be used between DC GW and PE2 for the route population in between. In fact, this is like the Option A in [RFC4364]. This configuration can work with any NVO3 solution. The eBGP, OSPF, or other can be used between PE1 and CE for the route population.



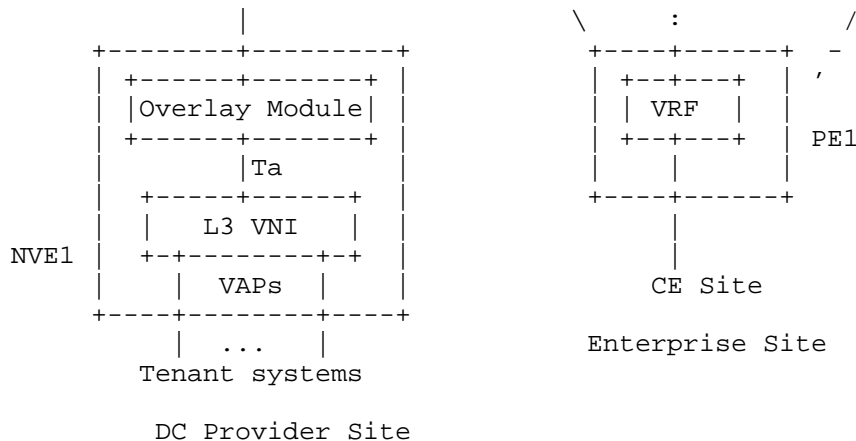


Figure 3 L3 VNI and L3VPN interconnection across multi networks

If an enterprise only has one location, it may use P2P VPWS [RFC4664] or L2TP [RFC5641] to connect one DC provider site. In this case, one edge connects to a physical network and another edge connects to an overlay network.

Various alternatives can be configured between DC GW and SP PE to achieve the same capability. Option B, C, or D in RFC4364 [RFC4364] can be used and the characteristics of each option are described there.

The interesting feature in this use case is that the L3 VN and compute resource are managed by the DC provider. The DC operator can place them at any location without notifying the enterprise and carrier because the DC physical network is completely isolated from the carrier and enterprise network. Furthermore, the DC operator may move the VMs assigned to the enterprise from one sever to another in the DC without the enterprise customer awareness, i.e. no impact on the enterprise 'live' applications running these resources. Such advanced feature brings some requirements for NVO3 [NVO3PRBM].

5. DC Applications Using NVO3

NVO3 brings DC operators the flexibility to design different applications in a true virtual environment (or nearly true) without worrying about physical network configuration in the Data Center. DC operators may build several virtual networks and interconnect them directly to form a tenant virtual network and implement the

communication rules, i.e. policy between different virtual networks; or may allocate some VMs to run tenant applications and some to run network service application such as Firewall and DNS for the tenant. Several use cases are given in this section.

5.1. Supporting Multi Technologies in a Data Center

Most likely servers deployed in a large data center are rolled in at different times and may have different capacities/features. Some servers may be virtualized, some may not; some may be equipped with virtual switches, some may not. For the ones equipped with hypervisor based virtual switches, some may support VxLAN [VXLAN] encapsulation, some may support NVGRE encapsulation [NVGRE], and some may not support any types of encapsulation. To construct a tenant virtual network among these servers and the ToRs, it may use two virtual networks and a gateway to allow different implementations working together. For example, one virtual network uses VxLAN encapsulation and another virtual network uses traditional VLAN.

The gateway entity, either on VMs or standalone one, participates in to both virtual networks, and maps the services and identifiers and changes the packet encapsulations.

5.2. Tenant Virtual Network with Bridging/Routing

A tenant virtual network may span across multiple Data Centers. DC operator may want to use L2VN within a DC and L3VN outside DCs for a tenant network. This is very similar to today's DC physical network configuration. L2 bridging has the simplicity and endpoint awareness while L3 routing has advantages in policy based routing, aggregation, and scalability. For this configuration, the virtual L2/L3 gateway function is necessary to interconnect L2VN and L3VN in each DC. Figure 4 illustrates this configuration.

Figure 4 depicts two DC sites. The site A constructs an L2VN that terminates on NVE1, NVE2, and GW1. An L3VN is configured between the GW1 at site A and the GW2 at site Z. An internal Virtual Network Interconnection Interface (VNIF) connects to L2VNI and L3VNI on GW1. Thus the GW1 is the members of the L2VN and L3VN. The L2VNI is the MAC/NVE mapping table and the L3VNI is IP prefix/NVE mapping table. Note that a VNI also has the mapping of TS and VAP at the local NVE. The site Z has the similar configuration. A packet coming to the GW1 from L2VN will be decapsulated and converted into an IP packet and then encapsulated and sent to the site Z. The Gateway uses ARP protocol to obtain MAC/IP address mapping.

Note that both the L2VN and L3VN in the figure are carried by the tunnels supported by the underlying networks which are not shown in the figure.

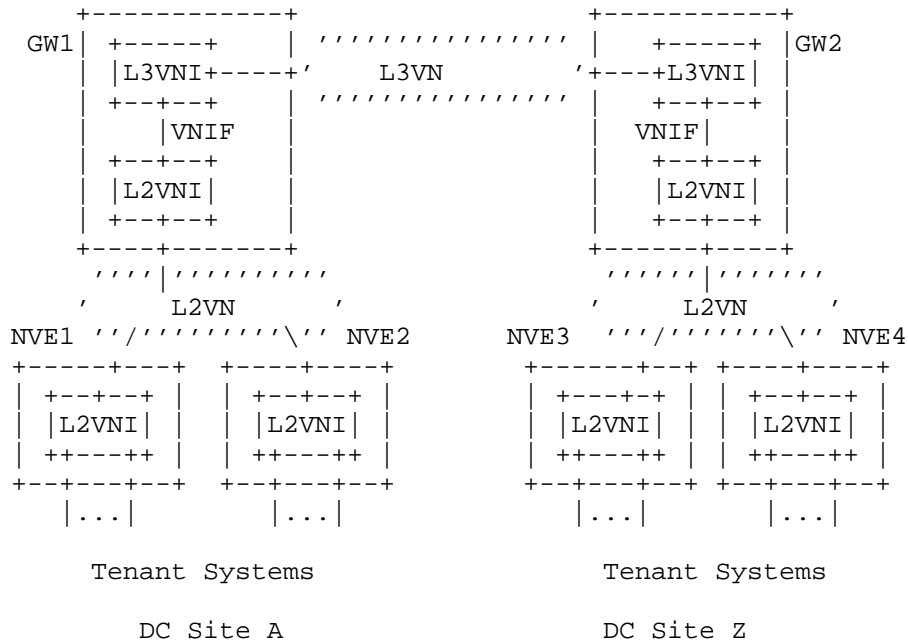


Figure 4 Tenant Virtual Network with Bridging/Routing

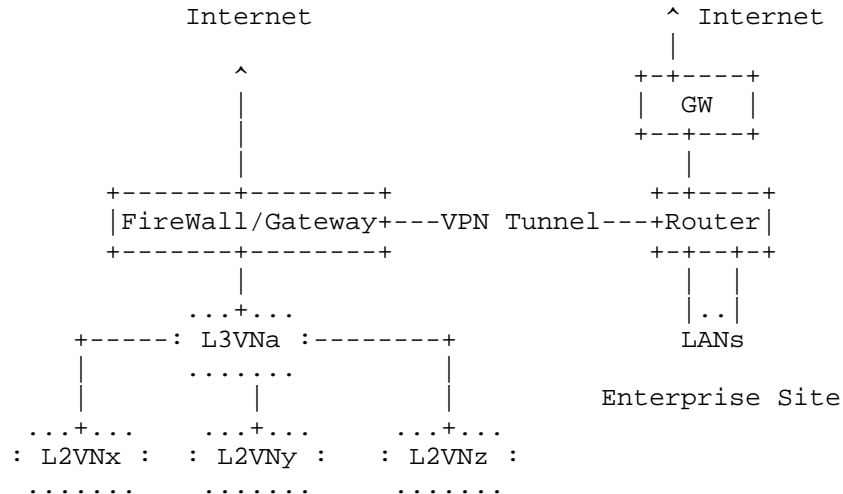
5.3. Virtual Data Center (VDC)

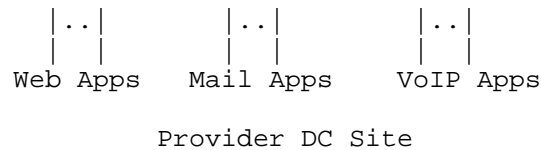
Enterprise DC's today may often use several routers, switches, and service devices to construct its internal network, DMZ, and external network access. A DC Provider may offer a virtual DC to an enterprise customer to run enterprise applications such as website/emails. Instead of using many hardware devices, with the overlay and virtualization technology of NVO3, DC operators can build them on top of a common network infrastructure for many customers and run service applications per customer basis. The service applications may include firewall, gateway, DNS, load balancer, NAT, etc.

Figure 5 below illustrates this scenario. For the simple illustration, it only shows the L3VN or L2VN as virtual and overlay routers or switches. In this case, DC operators construct several L2 VNs (L2VNx, L2VNy, L2VNz in Figure 5) to group the end tenant systems together per application basis, create an L3VNa for the internal routing. A server or VM runs firewall/gateway applications and connects to the L3VNa and Internet. A VPN tunnel is also built between the gateway and enterprise router. The design runs Enterprise Web/Mail/Voice applications at the provider DC site; lets the users at Enterprise site to access the applications via the VPN tunnel and Internet via a gateway at the Enterprise site; let Internet users access the applications via the gateway in the provider DC. The enterprise operators can also use the VPN tunnel or IPsec over Internet to access the vDC for the management purpose. The firewall/gateway provides application-level and packet-level gateway function and/or NAT function.

The Enterprise customer decides which applications are accessed by intranet only and which by both intranet and extranet; DC operators then design and configure the proper security policy and gateway function. DC operators may further set different QoS levels for the different applications for a customer.

This application requires the NVO3 solution to provide the DC operator an easy way to create NVEs and VNIs for any design and to quickly assign TSs to a VNI, easily place and configure policies on an NVE, and support VM mobility.





* firewall/gateway may run on a server or VMs

Figure 5 Virtual Data Center by Using NVO3

5.4. Federating NV03 Domains

Two general cases are 1) Federating AS managed by a single operator; 2) Federating AS managed by different Operators. The detail will be described in next version.

6. OAM Considerations

NVO3 brings the ability for a DC provider to segregate tenant traffic. A DC provider needs to manage and maintain NVO3 instances. Similarly, the tenant needs to be informed about tunnel failures impacting tenant applications.

Various OAM and SOAM tools and procedures are defined in [IEEE 802.1ag], [ITU-T Y.1731], [RFC4378], [RFC5880], [ITU-T Y.1564] for L2 and L3 networks, and for user, including continuity check, loopback, link trace, testing, alarms such as AIS/RDI, and on-demand and periodic measurements. These procedures may apply to tenant overlay networks and tenants not only for proactive maintenance, but also to ensure support of Service Level Agreements (SLAs).

As the tunnel traverses different networks, OAM messages need to be translated at the edge of each network to ensure end-to-end OAM.

It is important that failures at lower layers which do not affect NVo3 instance are to be suppressed.

7. Summary

The document describes some basic potential use cases of NVO3. The combination of these cases should give operators flexibility and capability to design more sophisticated cases for various purposes.

The key requirements for NVO3 are 1) traffic segregation; 2) supporting a large scale number of virtual networks in a common infrastructure; 3) supporting highly distributed virtual network

with sparse memberships 3) VM mobility 4) auto or easy to construct a NVE and its associated TS; 5) Security 6) NVO3 Management [NVO3PRBM].

Difference between other overlay network technologies and NVO3 is that the client edges of the NVO3 network are individual and virtualized hosts, not network sites or LANs. NVO3 enables these virtual hosts communicating in a true virtual environment without constraints in physical networks.

NVO3 allows individual tenant virtual networks to use their own address space and isolates the space from the network infrastructure. The approach not only segregates the traffic from multi tenants on a common infrastructure but also makes VM placement and move easier.

DC services may vary from infrastructure as a service (IaaS), platform as a service (PaaS), to software as a service (SaaS), in which the network virtual overlay is just a portion of an application service. NVO3 decouples the services from DC network infrastructure configuration.

NVO3's underlying network provides the tunneling between NVEs so that two NVEs appear as one hop to each other. Many tunneling technologies can serve this function. The tunneling may in turn be tunneled over other intermediate tunnels over the Internet or other WANs. It is also possible that intra DC and inter DC tunnels are stitched together to form an end-to-end tunnel between two NVEs.

A DC virtual network may be accessed via an external network in a secure way. Many existing technologies can help achieve this.

8. Security Considerations

Security is a concern. DC operators need to provide a tenant a secured virtual network, which means one tenant's traffic isolated from the other tenant's traffic and non-tenant's traffic; they also need to prevent DC underlying network from any tenant application attacking through the tenant virtual network or one tenant application attacking another tenant application via DC networks. For example, a tenant application attempts to generate a large volume of traffic to overload DC underlying network. The NVO3 solution has to address these issues.

9. IANA Considerations

This document does not request any action from IANA.

10. Acknowledgements

Authors like to thank Sue Hares, Young Lee, David Black, Pedro Marques, Mike McBride, David McDysan, Randy Bush, and Uma Chunduri for the review, comments, and suggestions.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.
- [IEEE 802.1ag] "Virtual Bridged Local Area Networks - Amendment 5: Connectivity Fault Management", December 2007.
- [ITU-T G.8013/Y.1731] OAM Functions and Mechanisms for Ethernet based Networks, 2011.
- [ITU-T Y.1564] "Ethernet service activation test methodology", 2011.
- [RFC4378] Allan, D., Nadeau, T., "A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM)", RFC4378, February 2006
- [RFC4301] Kent, S., "Security Architecture for the Internet Protocol", rfc4301, December 2005
- [RFC4664] Andersson, L., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", rfc4664, September 2006
- [RFC4797] Rekhter, Y., et al, "Use of Provider Edge to Provider Edge (PE-PE) Generic Routing Encapsulation (GRE) or IP in BGP/MPLS IP Virtual Private Networks", RFC4797, January 2007
- [RFC5641] McGill, N., "Layer 2 Tunneling Protocol Version 3 (L2TPv3) Extended Circuit Status Values", rfc5641, April 2009.
- [RFC5880] Katz, D. and Ward, D., "Bidirectional Forwarding Detection (BFD)", rfc5880, June 2010.

11.2. Informative References

- [NVGRE] Sridharan, M., "NVGRE: Network Virtualization using Generic Routing Encapsulation", draft-sridharan-virtualization-nvgre-01, work in progress.
- [NVO3PRBM] Narten, T., etc "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-02, work in progress.
- [NVO3FRWK] Lasserre, M., Motin, T., and etc, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-02, work in progress.
- [VRF-LITE] Cisco, "Configuring VRF-lite", <http://www.cisco.com>
- [VXLAN] Mahalingam, M., Dutt, D., etc "VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", draft-mahalingam-dutt-dcops-vxlan-02.txt, work in progress.

Authors' Addresses

Lucy Yong
Huawei Technologies,
4320 Legacy Dr.
Plano, Tx75025 US

Phone: +1-469-277-5837
Email: lucy.yong@huawei.com

Mehmet Toy
Comcast
1800 Bishops Gate Blvd.,
Mount Laurel, NJ 08054

Phone : +1-856-792-2801
E-mail : mehmet_toy@cable.comcast.com

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

Vishwas Manral
Hewlett-Packard Corp.
191111 Pruneridge Ave.

Cupertino, CA 95014

Phone: 408-447-1497

Email: vishwas.manral@hp.com

Linda Dunbar

Huawei Technologies,

4320 Legacy Dr.

Plano, Tx75025 US

Phone: +1-469-277-5840

Email: linda.dunbar@huawei.com

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

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

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

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

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

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

Table of Contents

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

1. Introduction

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

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

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

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

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

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

2. Terminology

This document uses the same terminology as found in the NV03 Framework document, [I-D.ietf-nvo3-framework]. Some of the terms defined in the Framework document have been repeated in this section for the convenience of the reader, along with additional terminology that is used by this document.

IMA: Information Mapping Authority.

[I-D.ietf-nvo3-overlay-problem-statement] uses the term "oracle" to describe this. It is a back-end system that is responsible for distributing and maintaining the mapping information for the entire overlay system. Note that the WG never reached consensus on what to call this architectural entity within the overlay system, so this term is subject to change.

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

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

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

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

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

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

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

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

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

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

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

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

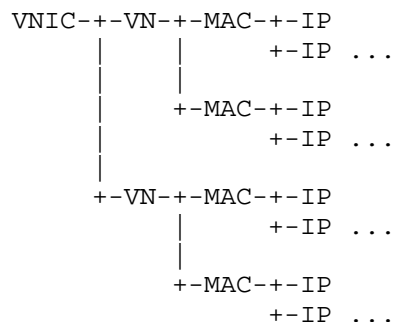
3. Entity Relationships

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

3.1. VNIC Containment Relationship

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

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



VNIC Containment Relationship

Figure 1

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

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

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

individual IP addresses.

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

VNIC--VN--MAC--IP

Figure 2

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

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

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

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

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

3.1.1. Layer 2 Virtual Network Service

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

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

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

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

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

3.1.2. Layer 3 Virtual Network Service

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

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

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

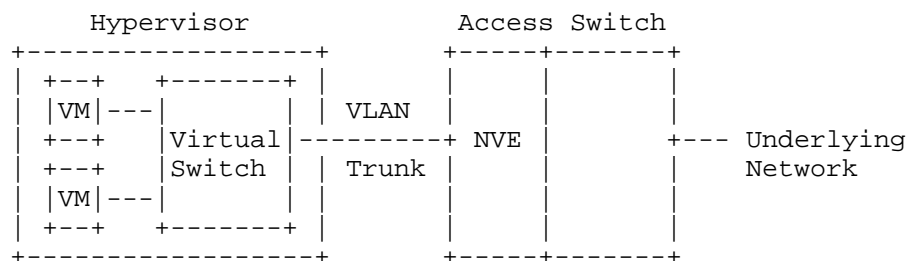
4. Hypervisor-to-NVE Control Plane Protocol Functionality

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

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

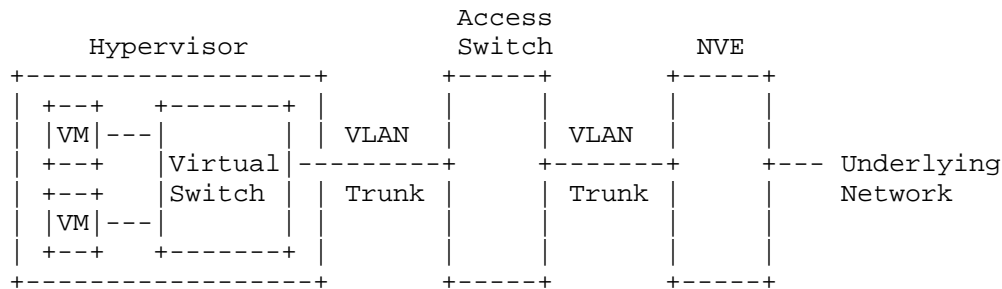
Alternatively, a Tenant System may use an externally connected NVE. An external NVE can provide an offload of the encapsulation / decapsulation function, network policy enforcement, as well as the VN Overlay protocol overheads. This offloading may provide performance improvements and/or resource savings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

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



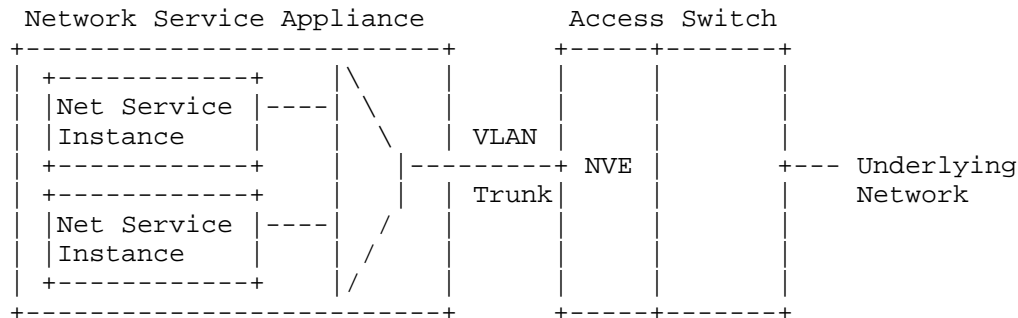
Hypervisor with an External NVE.

Figure 3



Hypervisor with an External NVE across an Ethernet Access Switch.

Figure 4



Physical Network Service Appliance with an External NVE.

Figure 5

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

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

adds the appropriate overlay encapsulation for that VN.

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

4.1. VN Connect/Disconnect

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

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

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

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

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

4.2. VNIC Address Association

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

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

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

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

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

4.3. VNIC Address Disassociation

When a VNIC within an End Device terminates function (due to events such as VNIC shutdown, Tenant System (TS) shutdown, or VM migration to another hypervisor), all addresses associated with that VNIC must be disassociated with the End Device on the connected NVE.

If the VNIC only has a single address associated with it, then this can be a single address disassociate message to the NVE. However, if the VNIC had hundreds of addresses associated with it, then the protocol with the NVE would be better optimized to simply disassociate the VNIC with the NVE, and the NVE can automatically disassociate all addresses that were associated with the VNIC.

Having TS addresses associated with a VNIC can also provide scalability benefits when the VM migrates between hypervisors that are connected to the same NVE. When a VM migrates to another hypervisor connected to the same NVE, if the NVE is aware of the migration, there is no need for all the addresses to be purged from NVE (and IMA) only to be immediately re-established again when the VM migration completes.

If the device containing the NVE is supporting many hypervisors, it may be quite likely that the VM migration will result in the VNICs still being associated with the same NVE, but simply on a different port. From the point of view of the IMA, nothing has changed and it would be inefficient to signal these changes to the IMA for no benefit. The NVE only needs to associate the addresses with a different port/tag pair.

It is possible for the NVE to handle a VM migration by using a timer to retain the VNIC addresses for a short time to see if the disassociated VNIC re-associates on another NVE port, but this could be better handled if the NVE knew the difference between a VNIC/VM shutdown and a VM migration. This leads to the next section.

4.4. VNIC Shutdown/Startup/Migration

As discussed above, the NVE can make optimizations if it knows which addresses are associated with which VNICs within an End Device and also is notified of state changes of that VNIC, specifically the difference between VNIC shutdown/startup and VNIC migration arrival/departure.

Upon VNIC shutdown, the NVE can immediately signal to the IMA that the bindings of the VNIC's addresses to the NVE's IP address can be removed.

Upon VNIC arrival, the NVE could either start a timer to hold the VNIC address bindings waiting to see if the VNIC arrives on a different port, or if there is a pre-arrival handshake with the NVE, then it will already know that the VNIC is going to be reassociated with the same NVE.

Upon VNIC arrival, the NVE knows that any addresses previously bound to the VNIC are still present and has no need to signal any change in address mappings to the IMA.

Note that if the IMA is also aware of VNIC address bindings, it can similarly participate efficiently in a VM migration that occurs across two different NVEs.

4.5. VN Profile

Once an NVE (embedded or external) receives a VN connect indication with a specified VN Name or ID, the NVE must determine the VN Context value to encapsulate packets with as well as other information that may be needed (e.g., QoS settings). The NVE serving that hypervisor needs a way to get a VN Context allocated or receive the already allocated VN Context for a given VN Name or ID (as well as any other information needed to transmit encapsulated packets). A protocol for an NVE to get this mapping may be a useful function, but would be the subject of work items 1 and 2 in [I-D.ietf-nvo3-overlay-problem-statement].

5. Security Considerations

Editor's Note: This is an initial start on the security considerations section; it will need to be expanded, and suggestions for material to add are welcome.

NVEs must ensure that only properly authorized Tenant Systems are allowed to join and become a part of any specific Virtual Network. In addition, NVEs will need appropriate mechanisms to ensure that any hypervisor wishing to use the services of an NVE are properly authorized to do so. One design point is whether the hypervisor should supply the NVE with necessary information (e.g., VM addresses, VN information, or other parameters) that the NVE uses directly, or whether the hypervisor should only supply a VN ID and an identifier for the associated VM (e.g., its MAC address), with the NVE using that information to obtain the information needed to validate the hypervisor-provided parameters or obtain related parameters in a secure manner.

6. Acknowledgements

Thanks to the following people for reviewing and providing feedback: Vipin Jain and Shyam Kapadia.

7. Informative References

[I-D.gu-nvo3-overlay-cp-arch]
Yingjie, G. and W. Hao, "Analysis of external assistance to NVE and consideration of architecture",
draft-gu-nvo3-overlay-cp-arch-00 (work in progress),
July 2012.

[I-D.gu-nvo3-tes-nve-mechanism]

Yingjie, G. and L. Yizhou, "The mechanism and signalling between TES and NVE", draft-gu-nvo3-tes-nve-mechanism-01 (work in progress), October 2012.

[I-D.ietf-nvo3-framework]

Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-02 (work in progress), February 2013.

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

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

[I-D.kompella-nvo3-server2nve]

Kompella, K., Rekhter, Y., and T. Morin, "Signaling Virtual Machine Activity to the Network Virtualization Edge", draft-kompella-nvo3-server2nve-01 (work in progress), October 2012.

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

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

Authors' Addresses

Lawrence Kreeger
Cisco

Email: kreeger@cisco.com

Thomas Narten
IBM

Email: narten@us.ibm.com

Internet-Draft NV03 Hypervisor-NVE Control Protocol Reqs February 2013

David Black
EMC

Email: david.black@emc.com

Network Virtualization Overlays Working
Group
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013

Q. Wu
Huawei
February 25, 2013

MAC address learning in NVO3 using ARP
draft-wu-nvo3-mac-learning-arp-01

Abstract

[I.D-ietf-nvo3-framework] discusses using Dynamic data plane learning or control plane protocol to build and maintain the mapping tables and deliver encapsulated packets to their proper destination. However, there is no relevant work to discuss how those capabilities can be realized in details at the NVEs. This document goes into details to discuss how MAC address learning works through data plane and control plane.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

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

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

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions used in this document	4
3. Overview of MAC address learning using ARP	5
4. Possible solutions for MAC learning using ARP Resolution	7
4.1. MAC learning using flooding without MAC hiding	7
4.2. MAC learning using NVE-oracle interaction	7
4.3. MAC learning using control plane operation and MAC hiding	8
4.4. MAC learning using control plane operation without MAC hiding	9
5. Discuss	11
6. IANA Considerations	12
7. Security Considerations	13
8. Normative References	14
Author's Address	15

1. Introduction

[I.D-ietf-nvo3-framework] discusses using Dynamic data plane learning or control plane protocol to build and maintain the mapping tables and deliver encapsulated packets to their proper destination. However, there is no relevant work to discuss how those capability can be realized in details at the NVEs. This document provides an overview of MAC address learning using ARP and discuss several MAC address learning methods by relying on ARP and mapping tables through data plane and control plane in NVO3. Comparing and evaluating these methods will be provided in the future version.

2. Conventions used in this document

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

3. Overview of MAC address learning using ARP

This document addresses how to build and maintain mapping table at the NVE associated with the tenant system through data plane learning or control plane.

Figure 1 shows the example architecture for MAC learning using ARP. This example architecture assumes that:

- o Tenant system A is connecting to VN by attaching to NVE X. Tenant System A knows IP address of Tenant System B but does not know MAC address of Tenant System B.
- o Tenant system B is connecting to VN by attaching to NVE Y. Tenant System B knows IP address of Tenant System A but does not know MAC address of Tenant System A.
- o NVE X associated with tenant system A doesn't know IP address and MAC address of tenant system B.
- o NVE Y associated with tenant system B doesn't know IP address and MAC address of tenant system A.

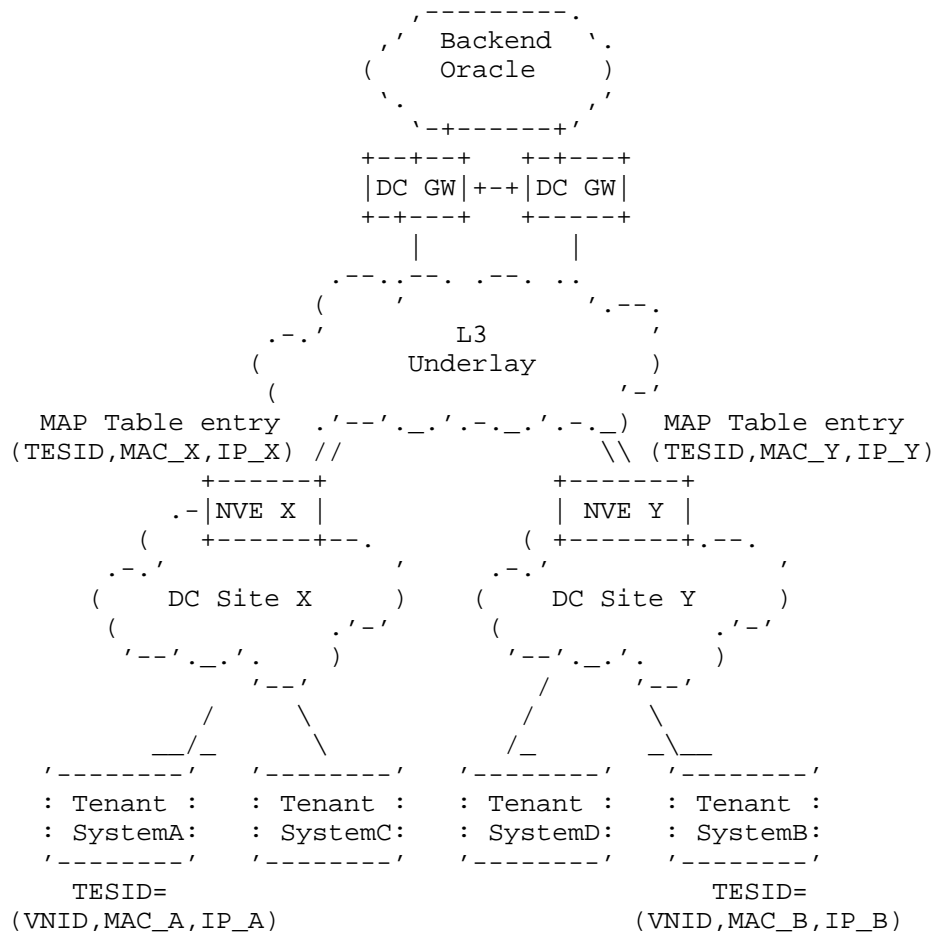


Figure 1: Figure 1 Example of MAC learning using ARP

4. Possible solutions for MAC learning using ARP Resolution

MAC addresses of the Tenant systems also can be learnt by NVE through data plane and control plane. The following section outlines several examples for MAC learning using ARP resolution.

4.1. MAC learning using flooding without MAC hiding

The packet flow and control plane operation for MAC learning are as follows:

- o Tenant system A sends a broadcast ARP message to discover the MAC address of Tenant system B. The message contains IP_B in the ARP message payload.
- o The ARP proxy in NVE X, receiving the ARP message, will flood it on the overlay network for TESID = <VNID,IP_B,*>.
- o The ARP message will be intercepted by NVE (i.e., NVE Y) which maintain mapping table matching TESID = <VNID,IP_B,*>. NVE Y, will forward the ARP message to tenant system B. Tenant System B send ARP reply to tenant system A containing the mapping TESID=<VNID,IP_B,MAC_B>.
- o NVE X intercept ARP reply message and populates the map-table with the received entry, then send it to Tenant System A that includes MAC_B and IP_B of Tenant System B.
- o Tenant system A learns MAC_B from the ARP rely message and can now send a packet to Tenant system B by including MAC_B, and IP_B, as destination addresses.

4.2. MAC learning using NVE-oracle interaction

The packet flow and control plane operation for MAC learning are as follows:

- o Tenant System A sends a broadcast ARP message to discover the MAC address of Tenant system B. The message contains IP_B in the ARP message payload.
- o NVE A, receiving the ARP message, but rather than flooding it on the overlay network sends a Map-Request to the backend Oracle that maintains mapping information for entire overlay network for TESID = <VNID,IP_B,*>.
- o The Map-Request is routed by the backend Oracle to NVE Y, that will send a Map-Reply back to NVE X containing the mapping

TESID=<VNID,IP_B,MAC_B>. Alternatively, depending on the Backend Oracle configuration, the backend Oracle may send directly a Map-Reply to NVE X.

- o NVE X populates the map-table with the received entry, and sends an ARP-Agent Reply to Tenant System A that includes MAC_B and IP_B.
- o Tenant system A learns MAC_B from the ARP message and can now send a packet to Tenant system B by including MAC_B, and IP_B, as destination addresses.

4.3. MAC learning using control plane operation and MAC hiding

MAC addresses of the Tenant systems also can be learnt by NVE through control plane.

When tenant system A is attached to NVE X, the mapping table TESID=<VNID,IP_A,MAC_A> should be populated at the local NVE A. In order to enable tenant system A to communicate with any tenant system that is not under NVE X, the mapping table should be distributed to all the remote NVEs that belong to the same VN even through there is no tenant system which communicates with tenant system A behind the remote NVE. In order to achieve this, NVE X should know the list of remote NVE that belong to the same VN as NVE X and distribute the mapping table to each remote NVE. Alternatively, backend Oracle may know a list of tenant systems that is in communication with tenant system A and which remote NVE these tenant systems are attached to. So NVE X distribute the mapping table to the backend Oracle which in turn determine which remote NVE should populate mapping table and distribute mapping table to the corresponding remote NVE. The packet flow for MAC learning in data plane are as follows:

- o Tenant system A sends a broadcast ARP message to discover the MAC address of Tenant system B. The message contains IP_B in the ARP message payload.
- o The ARP proxy in NVE X, will terminate the ARP message, and create a ARP reply message, set the inner destination MAC address in the inner Ethernet header and sender MAC address in the payload of ARP reply message to NVE X's MAC address then send it back to tenant system A. Therefore ARP message is restricted within layer 2 network behind NVE X and will not be flooded to the entire overlay network at the outsider of NVE X.
- o Tenant system A learns MAC_B from the ARP rely message and send a packet to Tenant system B by including MAC_X, and IP_B, as destination addresses.

- o NVE X, will intercept the packet from tenant system A and perform a lookup operation in its map table for the destination TESID=<VNID, IP_B> and determine which tunnel the packet needs to be sent to. Then NVE X encapsulate the packet from tenant system A into tunnel header with NVE Y IP_Y as the destination address NVE X IP_X as the source address and transmit it across overlay network.
- o NVE Y decapsulates the tunnel packet from NVE X and take out the packet from tenant system A and send to the tenant system B.

4.4. MAC learning using control plane operation without MAC hiding

MAC addresses of the Tenant systems also can be learnt by NVE through control plane.

When tenant system A is attached to NVE X, the mapping table TESID=<VNID,IP_A,MAC_A> should be populated at the local NVE A. In order to enable tenant system A to communicate with any tenant system that is not under NVE X, the mapping table should be distributed to all the remote NVEs that belong to the same VN even through there is no tenant system which communicate with tenant system A behind the remote NVE. In order to achieve this, NVE X should know the list of remote NVE that belong to the same VN as NVE X and distribute the mapping table to each remote NVE. Alternatively, backend Oracle may know a list of tenant systems that is in communication with tenant system A and which remote NVE these tenant systems are attached to. So NVE X distribute the mapping table to the backend Oracle which in turn determine which remote NVE should populate mapping table and distribute mapping table to the corresponding remote NVE. The packet flow for MAC learning in data plane are as follows:

- o Tenant system A sends a broadcast ARP message to discover the MAC address of Tenant system B. The message contains IP_B in the ARP message payload.
- o The ARP proxy in NVE X, will terminate the ARP message, and look up the MAC_B in the local mapping table send the ARP reply message to tenant system A that includes MAC_B and IP_B. Therefore ARP message is restricted within layer 2 network behind NVE X and will not be flooded to the entire overlay network at the outsider of NVE X.
- o Tenant system A learns MAC_B from the ARP rely message and send a packet to Tenant system B by including MAC_B, and IP_B, as destination addresses.

- o NVE X, will intercept the packet from tenant system A and perform a lookup operation in its map table for the destination TESID=<VNID, IP_B> and determine which tunnel the packet needs to be sent to. Then NVE X encapsulate the packet from tenant system A into tunnel header with NVE Y IP_Y as the destination address NVE X IP_X as the source address and transmit it across overlay network.
- o NVE Y decapsulates the tunnel packet from NVE X and take out the packet from tenant system A and send to the tenant system B.

5. Discuss

TBC.

6. IANA Considerations

This document has no actions for IANA.

7. Security Considerations

TBC.

8. Normative References

[I.D-ietf-nvo3-framework]

Lasserre, M., "Framework for DC Network Virtualization",
ID draft-ietf-nvo3-framework-00, September 2012.

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

Author's Address

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Network Virtualization Overlays Working
Group
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2013

Q. Wu
Huawei
February 25, 2013

Signaling control/forward plane information between network
virtualization edges (NVEs)
draft-wu-nvo3-nve2nve-01

Abstract

This document discusses how to provide control plane and forward plane information to the NVE associated with the tenant system for enabling interconnect between Tenant Systems that belong to specific tenant network.

Status of this Memo

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

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

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

This Internet-Draft will expire on August 29, 2013.

Copyright Notice

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

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

described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Conventions used in this document	4
3. Solution Overview	5
4. Mapping tables at the NVE	7
5. Key functions for signaling control/forwarding info to NVEs	8
5.1. Create and Update tenant Virtual Network (VN)	8
5.2. Associate the NVE and tenant system with VN context	8
5.3. Populate mapping tables at the local NVE	9
5.4. Distribute the mapping table to remote NVEs in the VN	9
5.5. The mapping table update at the NVE when VM moves or connection fails	9
5.6. The VN context re-association at the NVE when VM moves	10
6. IANA Considerations	11
7. Security Considerations	12
8. References	13
8.1. Normative References	13
8.2. Informative References	13
Author's Address	14

1. Introduction

In [I.D-ietf-nvo3-framework], one control component is defined to provide the capability for Address advertisement and tunnel mapping. In [I.D-fw-nvo3-server2vcenter], the control interface between NVE and interconnection functionality is defined to provide the capability:

- o Enforce the network policy for each VM in the path from the NVE Edge associated with VM to the Tenant End System.
- o Populate forwarding table in the path from the NVE Edge associated with VM to the Tenant End System in the data center.
- o Populate mapping table in each NVE Edge that is in the virtual network across data centers under the control of the Director.

However, there is no relevant work to discuss how those capability can be realized at the NVEs. This document goes into details to discuss how to provide control plane and forward plane information to the NVE associated with the tenant system for enabling interconnect between Tenant Systems that belong to specific tenant network.

2. Conventions used in this document

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

Site :

If multiple tenant systems connect to the VN through one NVE, the collection of these tenant systems and the NVE associated with these tenant systems are referred to as a site or virtualization network subnet.

3. Solution Overview

This document addresses how to provide control plane and forward plane information to the NVE associated with the tenant system for enabling interconnect between Tenant Systems that belong to specific tenant network.

Figure 1 shows the example architecture for interconnection between tenant systems. This example architecture assumes that:

- o One tenant system or a NVE may belong to one tenant VN or several tenant VNs, e.g., VMa and NVE Edge4 belong to both VN2 and VN3.
- o If one tenant system belongs to multiple tenant VNs, it may connect to each tenant VN by being attached to one NVE or multiple NVEs, e.g., VM1 connect to VN1 by being attached to NVE Edge 1.
- o One site may belong to one tenant VN or several tenant VN, e.g., Site 2 belong to both VN2 and VN3.
- o if one tenant system in one VN want to communicate with one tenant system in another VN, the interconnection functionality should get involved to setup tunnel between the interconnection functionality and the NVEs associated with the tenant system.

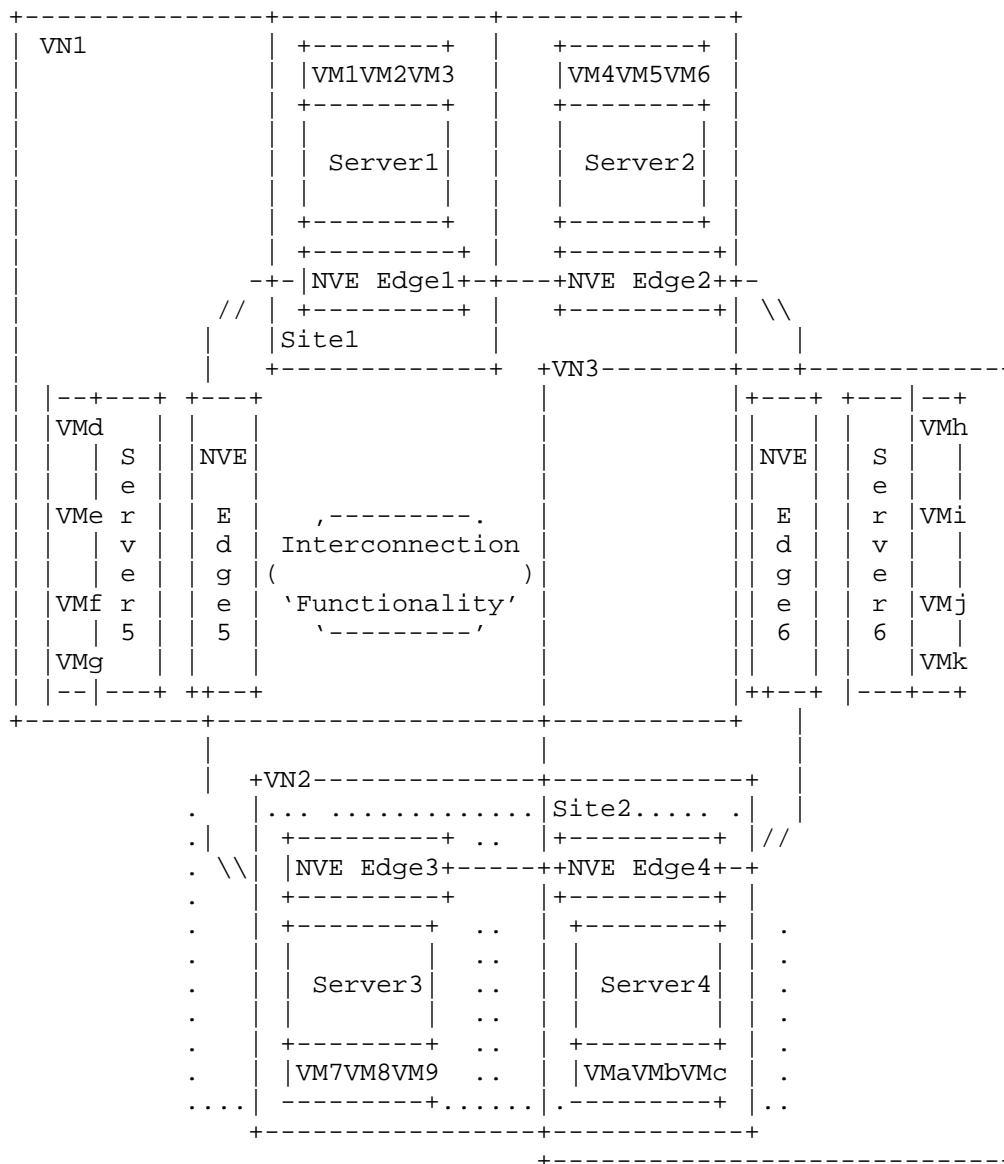


Figure 1: Figure 1.

4. Mapping tables at the NVE

Every NVE pair(local NVE and remote NVE) associated with the tenant system MUST maintain a mapping table entry for each currently attached tenant system. Each mapping table entry conceptually contains the following fields:

- o The tunnel interface identifier (tunnel-if-id) of the tunnel between the remote NVE and the local NVE where the tenant system is currently attached. The tunnel interface identifier is acquired during the tunnel creation.
- o The MAC address of the attached tenant end system. This MAC address is obtained from auto-discovery protocol between Tenant System and its local NVE.
- o The IP address of the attached tenant system. This IP address is obtained from auto-discovery protocol between Tenant System and its local NVE.
- o The IP address of the local NVE associated with the tenant system.
- o The Identifier of VN context-VNID. This Identifier is obtained from auto-discovery protocol between Tenant System and its local NVE.

5. Key functions for signaling control/forwarding info to NVEs

5.1. Create and Update tenant Virtual Network (VN)

The tenant virtualization network(VN) is a collection of tenant systems, Network Virtualization Edges (NVE)(s) and end systems that are interconnected with each other. The tenant VN also consists of a set of sites where each can send traffic directly to the other.

In order to create or update a tenant VN, when a Tenant System is attached to a local NVE, the tenant system should inform the attached local NVE which VN the tenant system belong to.

- o If the tenant system are the first participant in the VN through the local NVE, the tenant system and associated local NVE should be firstly added to the VN and the mapping table should be setup at the local NVE for each attached tenant system.
- o If both the tenant system and the local NVE are not on the VN, the tenant system and associated local should be firstly added to the VN and then the mapping table associated with this tenant system should be setup at the local NVE and distributed to the other remote NVEs that belong to the same VN.
- o If the local NVE is on the same tenant VN as the tenant system associated with the local NVE, only the tenant system needs to be added into the VN, i.e., the local NVE only needs to distribute mapping table at the local NVE to the other remote NVEs that belong to the same tenant VN.
- o If the local NVE is not on the same tenant VN as the tenant system associated with that local NVE, the local NVE should firstly be added into the VN and then distributes the new mapping table at the local NVE to the other remote NVEs that belong to the same tenant VN.
- o If one tenant system is the last participant connecting to the VN through local NVE, when this tenant system leave the VN, the local NVE associated with this tenant system should be removed from the VN. The mapping table associated with this tenant system should be removed from the local NVE associated with this tenant system.

5.2. Associate the NVE and tenant system with VN context

The VN context includes a set of configuration attributes defining access and tunnel policies and (L2 and/or L3) forwarding functions. When a Tenant System is attached to a local NVE, a VN network instance should be allocated to the local NVE. The tenant system

should be associated with the specific VN context using virtual Network Instance(VNI). The tenant system should also inform the attached local NVE which VN context the tenant system belong to. Therefore the VN context can be bound with the data path from the tenant system to the local NVE and the tunnel from local NVE associated with the tenant system and all the remote NVEs that belong to the same VN as the local NVE. For the data path from the tenant system and the local NVE, the network policy can be installed on the underlying switched network and forwarding tables also can be populated to each network elements in the underlying network based on the specific VNI associated with the tenant system. For the tunnel from local NVE to the remote NVEs, the traffic engineering information can be applied to each tunnel based on VNI associated with the tenant system.

5.3. Populate mapping tables at the local NVE

In some cases, two tenant systems may be attached to the same local NVE. In order to allow the NVE to locally route traffic between two tenant systems that are attached to the same NVE, the mapping table that maps a final destination address to the proper tunnel should be populated at the local NVE.

In some cases, two tenant systems may connect to the different VNs through the same interconnection functionality, in order to allow two tenant systems communication between two VNs, the mapping table that maps a final destination address to the proper tunnel should be populated in both NVE associated with two communicated tenant system and the interconnection functionality associated corresponding NVE.

5.4. Distribute the mapping table to remote NVEs in the VN

When the packet sent from one tenant system arrives at the ingress NVE associated with that tenant system, in order to determine which tunnel the packet needs to be sent to, the mapping table that maps a final destination address to the proper tunnel should also be distributed to all the remote NVEs in the VN using a control plane protocol or dynamic data plane learning. The mapping table may be advertised directly to other remote NVEs that belong to the same VN or firstly advertised to the centralized controller that maintain global view of NVEs that belong to the same VN and then let the centralized controller distribute the mapping tables to all the relevant remote NVEs that belong to the same VN.

5.5. The mapping table update at the NVE when VM moves or connection fails

In some cases, one tenant system may be detached from one NVE and

move to another NVE. In such cases, the mapping table should be removed from the NVE to which the tenant system was previously attached and the new mapping table should be created at the new NVE to which the tenant system is currently attached. Such mapping table should be updated at each remote NVE associated with the tenant system and the new NVE.

In some cases, one tenant system may fail to connect to the VN through the NVE. In such cases, the mapping table should be removed from the NVE to which the tenant system is currently attached. In addition, the mapping table should be updated at each remote NVE in the same VN through which the tenant system is communicating with the destination tenant system.

5.6. The VN context re-association at the NVE when VM moves

In some cases, one tenant system may be detached from one NVE and move to another NVE. In such cases, the VN context should be moved from the NVE to which the tenant system was previously attached to the new NVE to which the tenant system is currently attached. In order to achieve this, the per tenant system VN context can be maintained at the centralized database and be retrieved at the new place based on the VN Identifier (VNID).

6. IANA Considerations

This document has no actions for IANA.

7. Security Considerations

TBC.

8. References

8.1. Normative References

- [I.D-ietf-nvo3-framework]
Lasserre, M., "Framework for DC Network Virtualization",
ID draft-ietf-nvo3-framework-00, September 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", March 1997.

8.2. Informative References

- [I.D-fw-nvo3-server2vcenter]
Wu, Q. and R. Scott, "Network Virtualization
Architecture", ID draft-fw-nvo3-server2vcenter-01,
January 2013.

Author's Address

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Network working group
Internet Draft
Category: Informational

L. Yong
L. Dunbar
Huawei

Expires: March 2013

December 11, 2012

NVO3 Framework and Data Plane Requirement Addition
draft-yong-nvo3-frwk-dpreq-addition-00

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on March 11, 2013.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Abstract

This document describes some additional functions and requirements for NVO3 framework [NVO3FRWK] and data plane requirements [DPREQ]. These additions are necessary in supporting VM communication and mobility.

Conventions used in this document

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

Table of Contents

1. Introduction.....	3
2. New NVE Service Type.....	3
2.1. Why a New NVE Service Type.....	3
2.2. L2-3 NVE Providing IP Routing/Bridging-like Service (Framework Addition).....	4
2.3. L2-3 VNI (Data Plane Requirement Addition).....	4
3. Tenant System Mobility.....	5
3.1. Background.....	5
3.2. NVE Functions for TS Mobility (Framework Addition).....	6
3.2.1. Tenant System Mobility.....	6
3.2.2. Tenant Multicast Traffic.....	6
3.2.3. The Policy Associated With TS.....	7
3.3. Tenant System Mobility (Data Plane Requirement Addition).....	7
4. Security Considerations.....	8
5. IANA Considerations.....	8
6. Acknowledgements.....	8
7. References.....	8
7.1. Normative References.....	8
7.2. Informative References.....	8

1. Introduction

NVO3 framework [NVO3FRWK] and data plane requirement [DPREQ] documents specify the network virtualization overlay framework and data plane requirements, which aims on an architecture to support the network virtualization overlay in DC [NVO3PRBM]. The main application of NVO3 is to support multi-tenant networks on a common infrastructure, where a tenant virtual network may contain one or more subnets [HYPERV]. However, current framework specifies two NVE service types. Neither of them naturally supports the communication among the VMs when some VMs are on the same subnet and other on different. Second, one of the key aspects of NVO3 is to support the virtual machines (VM) mobility. However, neither document mentions VM mobility nor specifies any function and/or requirements in supporting VM mobility. This document addresses the two additions.

To use the terminologies specified in the framework document, this document refers VM mobility as to Tenant System mobility or TS mobility.

2. New NVE Service Type

2.1. Why a New NVE Service Type

A virtual machine on a server behaves like a physical server to an application or guest OS on it. This means that any frame from/to a virtual machine is an Ethernet frame, just as a frame from/to a physical server.

L2 NVE service type specified in the framework [NVO3FRWK] provides Ethernet LAN like service where multiple Tenant Systems appear to interconnected by an LAN environment over a set of L3 tunnels. However, from the host (physical servers or VMs) perspective, only the hosts on the same subnet can communicate in an LAN network. This implies that L2 NVE service type only applies to a single subnet.

L3 NVE service type [NVO3FRWK] provides a virtualized IP routing and forwarding like IETF IP VPN. The IP VPN emulates a route domain and provides forwarding and routing among TSes that are the same and/or different subnets. IETF IP VPN has the assumption that the Layer 3 MUST be implemented between a PE and a CE, which means between an NVE and a TS in this context. This assumption does not fit to the case where an NVE attached by the multiple TSes that are on the same subnet where the TSes uses bridging mechanism for the communication.

To support TSeS, regardless on the same or different subnets, communicating in an L2 environment, this document suggests adding a new L2-3 NVE Service Type. Suggested Text for the framework and data plane requirement documents is in section 2.2 and section 2.3, respectively.

2.2. L2-3 NVE Providing IP Routing/Bridging-like Service (Framework Addition)

L2-3 NVE is similar to IRB function on a router [CIRB] device today. It supports the TSeS attached to the NVE (locally or remotely) to communicate with each other when they are in a same route domain, i.e. a tenant virtual network. The NVE provides per tenant virtual switching and routing instance with address isolation and L3 tunnel encapsulation across the core. The L2-3 NVE supports the bridging among TSeS that are on the same subnet and the routing among TSeS that are on the different subnets.

2.3. L2-3 VNI (Data Plane Requirement Addition)

L2-3 VNIs MUST provide virtualized IP routing and bridging. L2-3 VNI MUST support per-tenant forwarding instance with IP and MAC address isolation and L3 tunneling for interconnecting instances of the same VNI on NVEs. L2-3 VNI MUST perform the virtual bridging for the Tenant Systems that are on the same subnet and the IP routing for the Tenant Systems that are on the different subnets. L2-3 VNI MUST support L2/3 gateway function.

L2-3 VNI MUST NOT change Tenant System communication mechanism in a route domain, i.e. a tenant virtual network, and not violate Tenant Systems communication rules. Tenant System communication rules are if Tenant Systems are on the same subnet, they are bridged directly; if Tenant Systems are on different subnets, they MUST communicate through a router. A tenant system uses the ARP/ND protocol to discover other tenant system MAC addresses if they are on the same subnet; a tenant system sends a packet to a known gateway if the destination of the packet is on different subnet from the sender TS; a tenant system uses ARP/ND protocol to find the gateway MAC address.

Forwarding table entries provide mapping information between MAC/IP and L3 Tunnel destination addresses. Such entries MAY be populated by a control or management plane.

The L2-3 VNI MUST support the ARP protocol at virtual access points (VAPs) and a default VGW MAC address.

In the case of L2-3 VNI, when the packet is forwarded from one subnet to another subnet, inner TTL field and outer TTL field process MUST be the same as described in L3 VNI section.

When tenant multicast is supported, L2-3 VNI SHOULD also be possible to select whether the NVE provides optimized multicast trees inside the VNI for individual tenant multicast groups or whether the default VNI multicast tree is used, where all the NVEs of the corresponding VNI are members, is used.

3. Tenant System Mobility

3.1. Background

NVO3 generic reference model specifies that a Tenant System can be attached to an NVE locally or remotely. The local means that a TS and the NVE are resident in the same device, e.g. server. The remote means a TS attached to the NVE via a point-to-point connection or a switched network, e.g. Ethernet.

When an NVE is local, the state of Tenant System can be provided without protocol assistance. This implies that when Tenant System state changes, the NVE is immediately aware of the changes. When an NVE is remote, the state of the Tenant System needs to be exchanged via a data or control plane protocol, or via a management entity.

VM mobility further requires support of hot and cold move [VMMOVE]. In the hot move, the moving is seamless to the application that runs on the moved TS, which implies that the existing connectivity between the moved TS and other TSes that the moved TS communicates with MUST be maintained while the TS is moved regardless if these TSes are on the same or different subnets.

When a TS and NVE are resident in the same device, the TS moves from one NVE, NVE1 to another NVE, NVE2. NVE2 instantly knows the TS address, state, and etc. However other NVEs that other TSes attach to and have the connectivity with the moved TS MUST be also aware of the TS new location, i.e. NVE2 location, and NVE2 MUST be also aware of these NVE locations in order to maintain the connectivity.

When a TS and NVE are remotely attached, TS moving only applies when a TS attaches to the NVE via a switched network, i.e. L2 physical and/or virtual network.[VMMOVE] In addition of the actions in the local NVE case (mentioned above), when an NVE is remote, the state of the Tenant System needs to be exchanged via a data or control plane protocol, or via a management entity.

Other two cases are a TS moved away from a local NVE and to a remote NVE and vice versa.

To support TS mobility, this document suggests adding a new section in the NVO3 framework and data plane requirement documents and the suggested text is in section 3.2 and 3.3, respectively.

3.2. NVE Functions for TS Mobility (Framework Addition)

3.2.1. Tenant System Mobility

If an NVE (say ingress NVE) is responsible to notify other NVEs (egress NVEs) regarding a new moved TS attaching to it. If the ingress NVE is not yet on the tenant virtual network that the moved TS belongs to, the NVE MUST establish the membership to the virtual network first and create a virtual access point (VAP) to associate to the virtual network. The NVE MUST send a notification about the TS to other egress NVEs that has the same membership. This can be done via data plane or control plane. Upon receiving the notification from an ingress NVE, an egress NVE has to update its VNIs that are associate to the same membership. If an NVE is remote, the VNI MUST send the new TS address notification to the access networks via the virtual access points (VAPs).

Note that if the ingress NVE is L2-3 NVE, and if it is not yet on the same tenant virtual network subnet as the moved TS belongs to, the NVE MUST establish the membership to the virtual subnet network first and create a VAP to associate with it. The NVE MUST send a notification about the TS to other egress NVEs that has the same membership. If an NVE is remote, they MUST only send the notification to the access networks that are on the same tenant virtual subnet as the moved TS is on.

Note that, when a TS moves away from an NVE and it is the last TS attached to the NVE belong to the tenant virtual network, the NVE MAY delete the membership of the tenant virtual network.

3.2.2. Tenant Multicast Traffic

If a tenant application on a set of TSes needs to send broadcast or multicast traffic among them, the NVE multicast and broadcast capability can facilitate such forwarding [NVO3FRWK]. To support VM mobility, when one of the TSes is moved from one NVE (say NVE1) to another (say NVE2) in hot mode, the NVE2 has to know which multicast groups that the TS is associated with. If NVE2 is local, such information can be available to the NVE2 via some API; if NVE2 is remote, such information can be available to the NVE2 via data plane,

control plane, or management entity [NVO3FRWK]. If NVE2 is need to learn Tenant Multicast Groups that a moved TS is on, the NVEs MUST be able to send a query message to the moved TS; The TS response which groups it is on.

Once NVE2 knows which multicast groups that the new attached TS is associated with, NVE2 MUST bind itself to these multicast groups if it is not on yet. Furthermore, NVE2 MAY (if not yet) have to bind the overlay multicast groups to one or more underlying multicast tree if it uses the underlay multicast trees to delivery overlay multicast traffic. An NVE MUST provide these capabilities, if it supports tenant multicast traffic, to ensure tenant application seamlessly running while a Tenant System is moved.

Similarly, when NVE1 knows a TS moved away and being the last one on the tenant virtual network, NVE1 MAY unbind itself to the corresponding multicast group. Furthermore, if this is the last multicast group on the NVE1, NVE1 MAY unbind the multicast group to the shared multicast tree if used.

3.2.3. The Policy Associated With TS

An NVE provides the policy based forwarding and routing [NVO3FRWK] [DPREQ]. When a TS is moved from one NVE (say NVE1) to another (say NVE2), the NVE2 has to apply to the same set of policy to the TS as well. If TS related policies are specified in the TS service profile that is moved along with the TS, and the file can be passed to the NVE2 via API if the TS is locally attached to or via a data plane or control plane protocol, or a management entity if remotely attached to. An NVE MUST be able to automatically install these policies at the VAP that a new TS attaches to. NVE1 MUST automatically delete the policies that are applied to the moved TS only.

3.3. Tenant System Mobility (Data Plane Requirement Addition)

If the data plane learning is used to populate the forwarding table[DPREQ], an NVE (local or remote) MUST be able to send a notification message to all the NVEs that are the membership of the tenant virtual network that the TS belongs to, e.g. ARP gratuitous message. The notification MUST contain the TS address and tenant VN ID. Upon receiving the notification message, an NVE MUST update the corresponding VNI indicated in the NVO3 overlay header. If the receiving NVE is remote, the NVE MUST send a notification to the local access networks that is on the same subnet as of one indicated in the NVO3 overlay header via the VAPs.

4. Security Considerations

When a Tenant System is moved from one NVE to another, automatic virtual network membership creation on an NVE may leave some security concern. Either certain authentication is needed for an NVE to accept a new TS or management entity assisted process is used to ensure the security.

Supporting TS mobility brings a new challenge for NVO3 is discussed in [NVO3PRBM].

5. IANA Considerations

The document does not require any IANA action.

6. Acknowledgements

Thank Weiguo Hao for the review and input to the draft.

7. References

7.1. Normative References

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

7.2. Informative References

[DPREQ] Bitar, N., and etc, "NVO3 Data Plane Requirement", draft-bl-nvo3-dataplane-requirements-03.txt, November 2012

[CIRB] Cisco, "Understanding and Configuring VLAN Routing and Bridging on a Router Using the IRB Feature", Doc. ID 17054

[HYPERV] Microsoft, "Hyper-V Network Virtualization Packet Flow", September 2012

[NVO3FRWK] LASSERRE, M., Motin, T., and etc, "Framework for DC Network Virtualization", draft-ietf-nvo3-framework-01, October 2012

[NVO3PRBM] Narten, T., and etc "Problem Statement: Overlays for Network Virtualization", draft-ietf-nvo3-overlay-problem-statement-01, October 2012

[VMMOVE] Rakhter, Y., and etc, "Network-related VM Mobility Issue", draft-rekhter-nvo3-vm-mobility-issues-03.txt, Sept. 2012

Authors' Addresses

Lucy Yong
Huawei USA
5340 Legacy Drive
Plano, TX 75025
U.S.A

Phone: 469-277-5837
Email: lucy.yong@huawei.com

Linda Dunbar
Huawei USA
5340 Legacy Drive
Plano, TX 75025
U.S.A

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

