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.

Table of Contents

## 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:

- vCenter Name or Identifier
- vServer Name or Identifier
- Virtual Network ID (VNID)
- VLAN tag value
- VM Group ID
- 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.

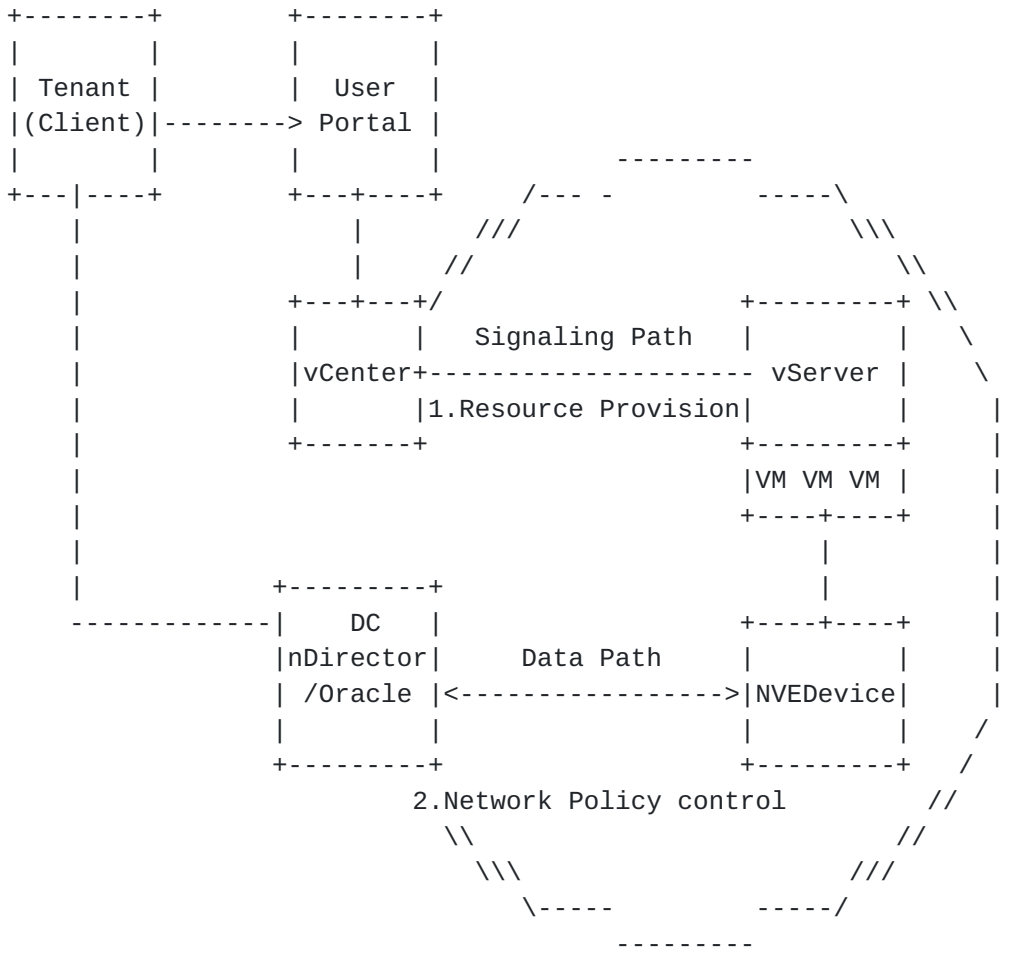
### 3.7. VM Monitoring

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

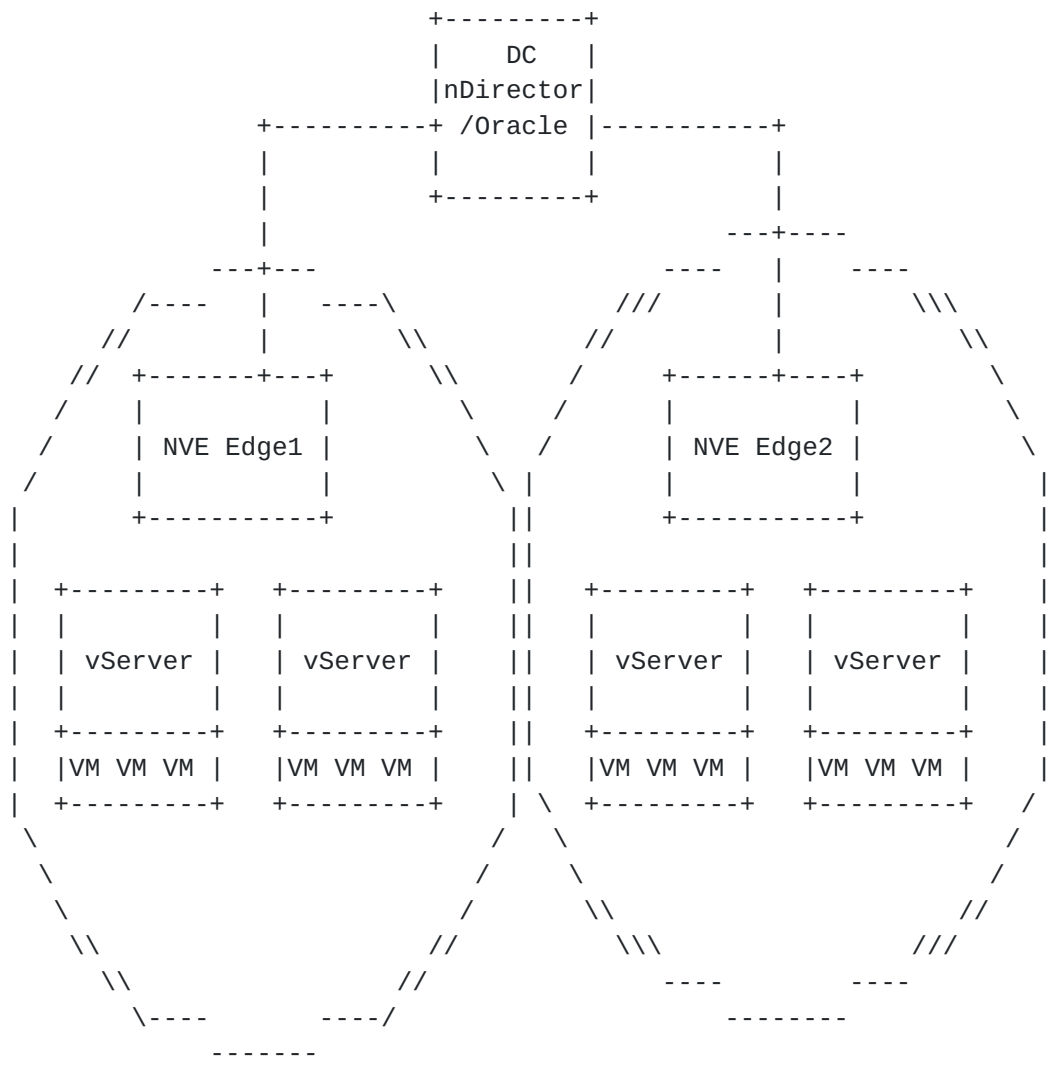
### 4.1. On Demand Network Provision Automation within the data center

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.



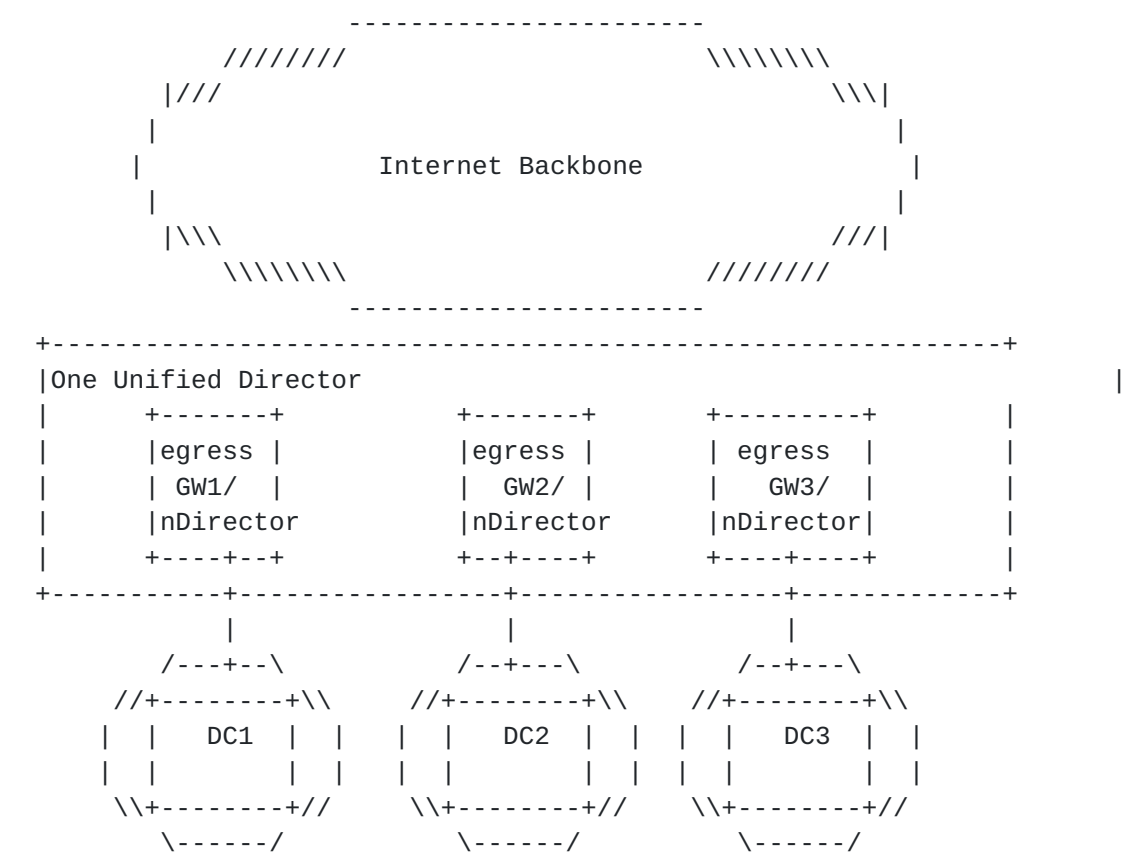
### 4.2. Large inter-data centers Layer 2 interconnection and data forwarding

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 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 remove NVE Edge device decapsulates layer 3 packet, take out layer 2 frame and forward it to ultimate destination VM2.



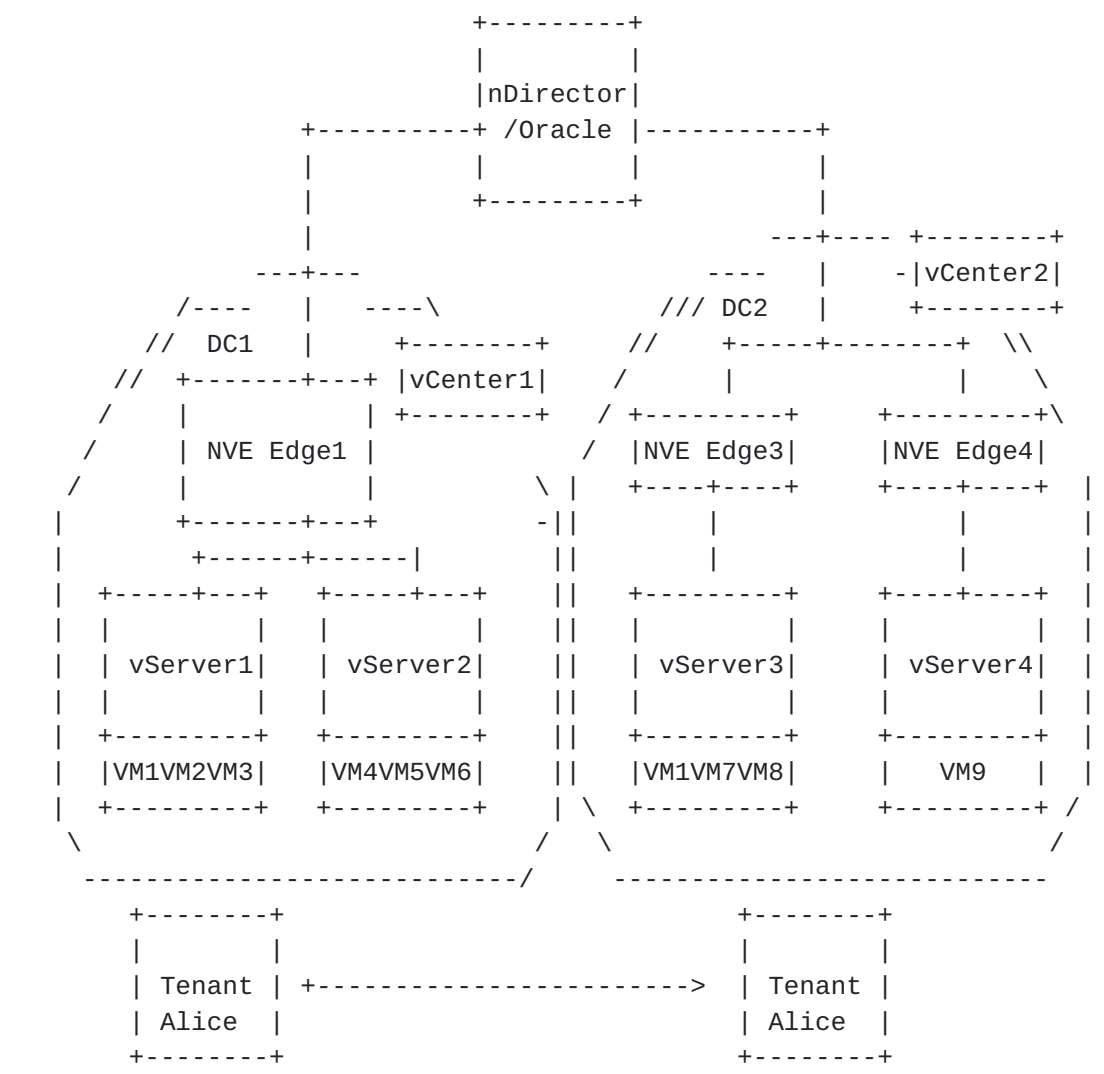

## 4.3. Enable multiple data centers present as one

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.



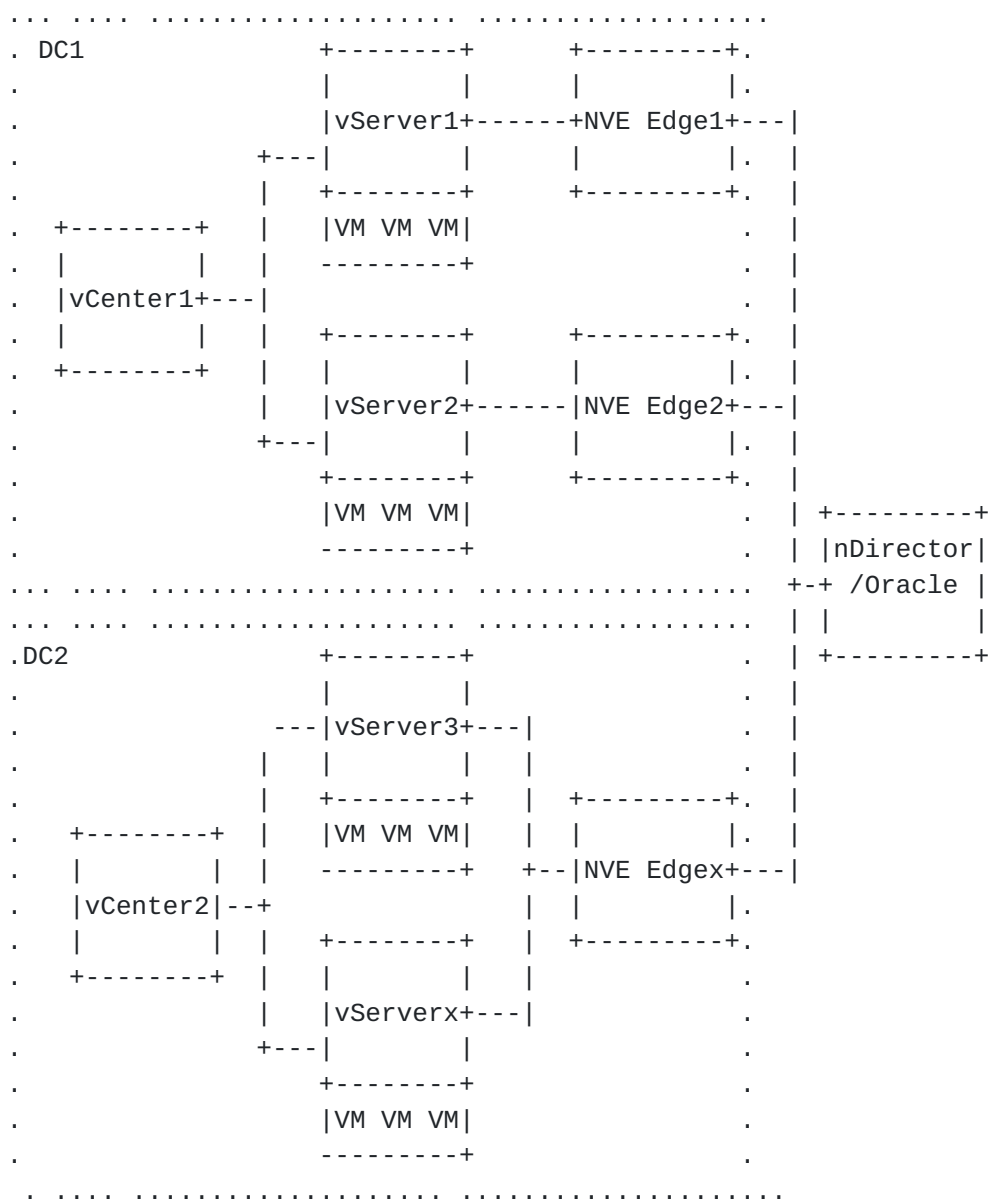

## 4.4. VM migration and mobility across data centers

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.


nDirector
Oracle
DC1
DC2
vCenter1
vCenter2
NVE Edge1
NVE Edge3
NVE Edge4
vServer1
vServer2
vServer3
vServer4
VM1VM2VM3
VM4VM5VM6
VM1VM7VM8
VM9
Tenant Alice
Tenant Alice

## 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 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 vSever 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 vSever 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