

Internet Engineering Task Force (IETF)  
Request for Comments: 7365  
Category: Informational  
ISSN: 2070-1721

M. Lasserre  
F. Balus  
Alcatel-Lucent  
T. Morin  
Orange  
N. Bitar  
Verizon  
Y. Rekhter  
Juniper  
October 2014

## Framework for Data Center (DC) Network Virtualization

### Abstract

This document provides a framework for Data Center (DC) Network Virtualization over Layer 3 (NVO3) and defines a reference model along with logical components required to design a solution.

### Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see [Section 2 of RFC 5741](#).

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7365>.

## 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 .....	4
1.1. General Terminology .....	4
1.2. DC Network Architecture .....	7
2. Reference Models .....	8
2.1. Generic Reference Model .....	8
2.2. NVE Reference Model .....	10
2.3. NVE Service Types .....	11
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 .....	12
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 .....	14
3.1.4. Tunnel Overlays and Encapsulation Options .....	14
3.1.5. Control-Plane Components .....	14
3.1.5.1. Distributed vs. Centralized Control Plane .....	14
3.1.5.2. Auto-provisioning and Service Discovery .....	15
3.1.5.3. Address Advertisement and Tunnel Mapping .....	15
3.1.5.4. Overlay Tunneling .....	16
3.2. Multihoming .....	16
3.3. VM Mobility .....	17
4. Key Aspects of Overlay Networks .....	17
4.1. Pros and Cons .....	18
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 .....	20
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. Informative References .....	24
Acknowledgments .....	26
Authors' Addresses .....	26

## 1. Introduction

This document provides a framework for Data Center (DC) Network Virtualization over Layer 3 (NVO3) tunnels. This framework is intended to aid in standardizing protocols and mechanisms to support large-scale network virtualization for data centers.

[RFC7364] 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 Layer 2 (L2) or Layer 3 (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, or a Network Service Appliance, or it could 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 an 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 that NVO3 packets are tunneled over, 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 computation, memory, and storage services 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 Network Interface Card (NIC) 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:** 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 or 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 and iSCSI storage systems), and network devices (e.g., firewall, load-balancer, and 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:

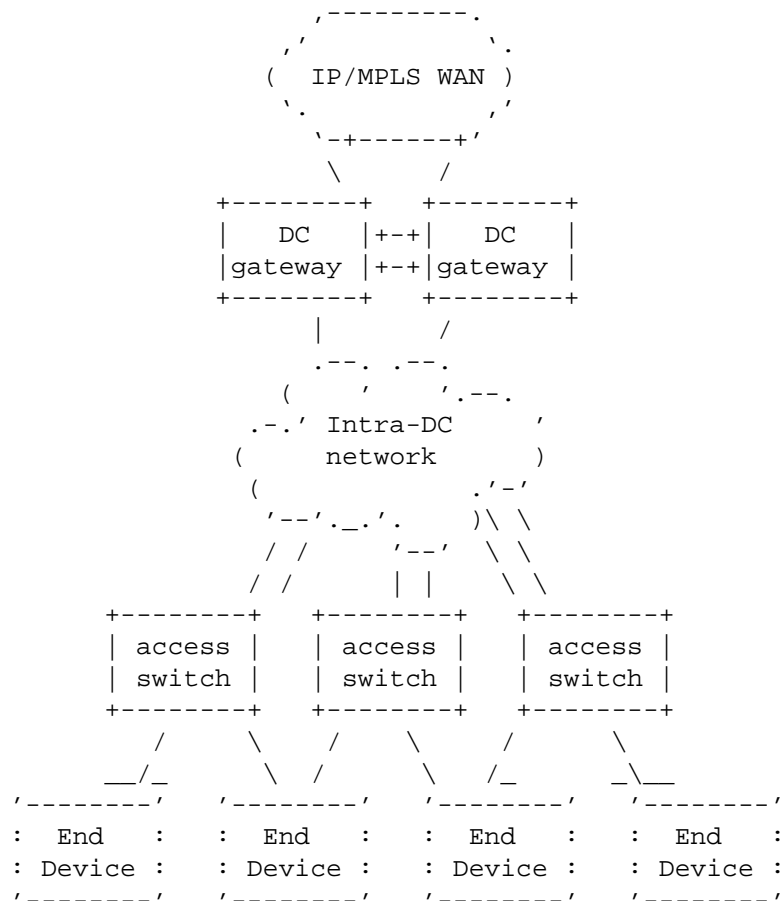


Figure 1: A Generic Architecture for Data Centers

An example of multi-tier DC network architecture is presented in Figure 1. It provides a view of the 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 Layer 2 tunneling over IP, for instance. Access switches are usually multihomed 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 End-of-Row (EoR) 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 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-homed or multihomed to access switches.

## 2. Reference Models

### 2.1. Generic Reference Model

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



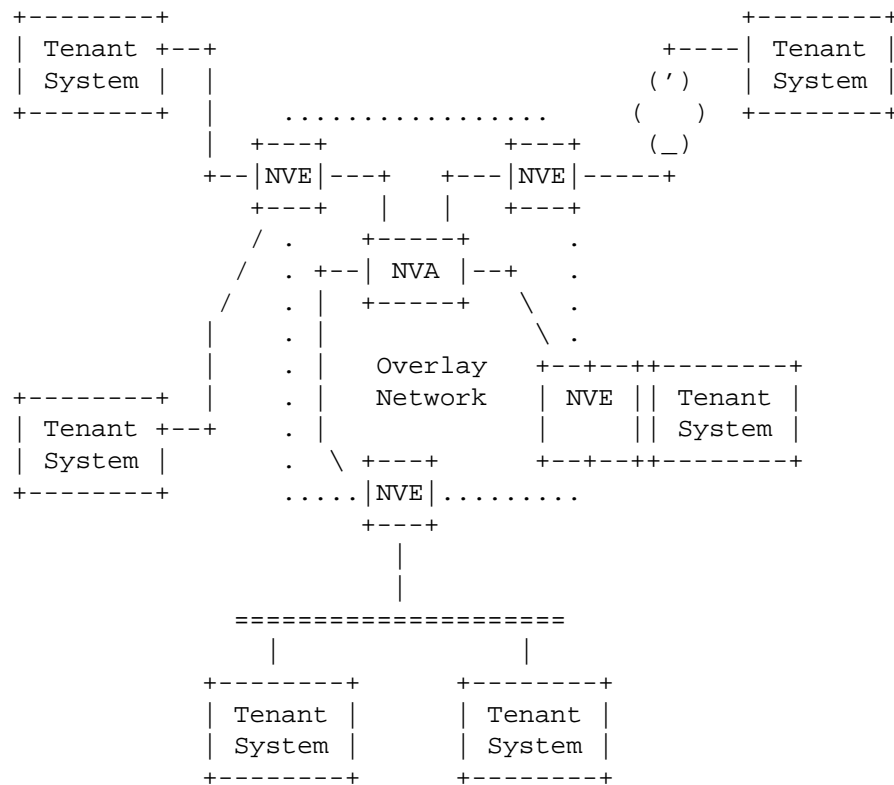


Figure 2: Generic Reference Model for DC Network Virtualization Overlays

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 data-plane 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 state for each tenant service, albeit some applications (e.g., multicast) may require control-plane or forwarding-plane information that pertains 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.).

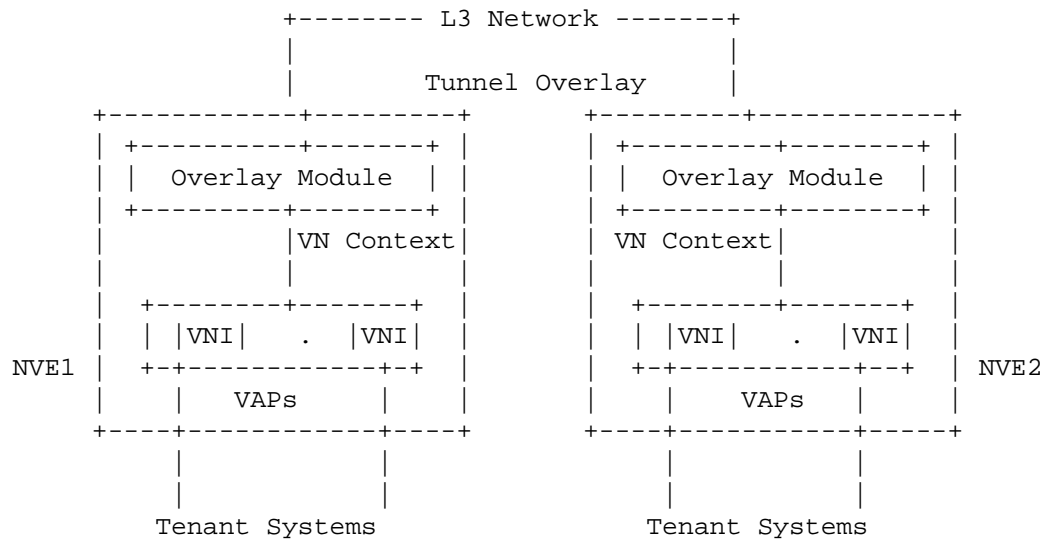


Figure 3: Generic NVE Reference Model

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 Virtual Private LAN Service (VPLS) [RFC4761][RFC4762] or Ethernet VPN [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 Media Access Control (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 IP VPN [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 Operations, Administration, and Maintenance (OAM) facilities are used.

With regard 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 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 the 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 and 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 encapsulation/decapsulation of frames from the 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: This is a 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 Service Instance IDs (I-SIDs) that identify virtual L2 domains when using IEEE 802.1Q 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 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., Generic Routing Encapsulation (GRE), the Layer 2 Tunneling Protocol (L2TP), and 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 and 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

Control- and management-plane entities 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 and 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 for 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 and service discovery could be accomplished by the same protocol. In this scenario, the auto-provisioning and 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 the Address Resolution Protocol (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 that 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 set up the tunnel path for some tunneling technologies. This applies to both unicast and multicast tunneling.

#### 3.2. Multihoming

Multihoming techniques can be used to increase the reliability of an NVO3 network. It is also important to ensure that the physical diversity in an NVO3 network is taken into account to avoid single points of failure.

Multihoming 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 techniques or MPLS re-rerouting capabilities as the means to re-route traffic upon failures.

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 L2 construct such as a VLAN, and it can be multihomed to various NVEs. An NVE may provide an L2



service to the end system or an L3 service. An NVE may be multihomed to a next layer in the DC at L2 or 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 an 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 multihomed 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 and 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 offers 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 provides tenant separation and separation of the tenant address spaces from the underlay address space.
- Overlay networks support of a large number of virtual network identifiers.

Overlay networks also create several challenges:

- Overlay networks typically have no control of underlay networks and lack underlay network information (e.g., underlay utilization):
  - o 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.
  - o Miscommunication or lack of coordination between overlay and underlay networks can lead to an inefficient usage of network resources.
  - o 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 data-plane learning or a control plane. For an L3 NVE, the NVE needs to be able to determine the 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; 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 Path MTU 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 middleboxes (e.g., firewalls) that disable ICMP for security reasons.

- Extended Path MTU Discovery techniques such as those 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
  - o Datapath (e.g., lookups, filtering, and encapsulation/decapsulation)
  - o Control-plane processing (e.g., routing, signaling, and OAM) and where specific control-plane functions should be enabled
- FIB/RIB size

- Multicast support
  - o Routing/signaling protocols
  - o Packet replication capability
  - o Multicast FIB
- Fragmentation support
- QoS support (e.g., marking, policing, and 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 general coordination in placing overlay demands on an underlay network, may be achieved by providing mechanisms to exchange performance and liveness information between the underlay and overlay(s) or by 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 underlay 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 secure access to NVEs and NVAs be ensured such that no unauthorized access is possible. As discussed in [Section 3.1.5.2](#), identification of Tenant Systems 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 NVE, 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 as discussed in [Section 4.2.6](#)) that can be exchanged between an NVO3 service and underlay networks based upon their agreed security requirements.

## 6. Informative References

- [EVPN] Sajassi, A., Aggarwal, R., Bitar, N., Isaac, A., and J. Uttaro, "BGP MPLS Based Ethernet VPN", Work in Progress, [draft-ietf-l2vpn-evpn-10](#), October 2014.
- [OPPSEC] Dukhovni, V. "Opportunistic Security: Some Protection Most of the Time", Work in Progress, [draft-dukhovni-opportunistic-security-04](#), August 2014.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", [RFC 1191](#), November 1990, <<http://www.rfc-editor.org/info/rfc1191>>.
- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", [RFC 1981](#), August 1996, <<http://www.rfc-editor.org/info/rfc1981>>.
- [RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", [RFC 2679](#), September 1999, <<http://www.rfc-editor.org/info/rfc2679>>.
- [RFC2680] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", [RFC 2680](#), September 1999, <<http://www.rfc-editor.org/info/rfc2680>>.
- [RFC3148] Mathis, M. and M. Allman, "A Framework for Defining Empirical Bulk Transfer Capacity Metrics", [RFC 3148](#), July 2001, <<http://www.rfc-editor.org/info/rfc3148>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", [RFC 3393](#), November 2002, <<http://www.rfc-editor.org/info/rfc3393>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", [RFC 4364](#), February 2006, <<http://www.rfc-editor.org/info/rfc4364>>.
- [RFC4761] Kompella, K., Ed., and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", [RFC 4761](#), January 2007, <<http://www.rfc-editor.org/info/rfc4761>>.
- [RFC4762] Lasserre, M., Ed., and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", [RFC 4762](#), January 2007, <<http://www.rfc-editor.org/info/rfc4762>>.



- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007, <<http://www.rfc-editor.org/info/rfc4821>>.
- [RFC6820] Narten, T., Karir, M., and I. Foo, "Address Resolution Problems in Large Data Center Networks", RFC 6820, January 2013, <<http://www.rfc-editor.org/info/rfc6820>>.
- [RFC7364] Narten, T., Ed., Gray, E., Ed., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", RFC 7364, October 2014, <<http://www.rfc-editor.org/info/rfc7364>>.

## Acknowledgments

In addition to the authors, the following people contributed to this document: Dimitrios Stiliadis, Rotem Salomonovitch, Lucy Yong, Thomas Narten, Larry Kreeger, and David Black.

## Authors' Addresses

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

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

Thomas Morin  
Orange  
EMail: thomas.morin@orange.com

Nabil Bitar  
Verizon  
50 Sylvan Road  
Waltham, MA 02145  
United States  
EMail: nabil.n.bitar@verizon.com

Yakov Rekhter  
Juniper  
EMail: yakov@juniper.net