

# Reference Design: VMware® NSX for vSphere (NSX) Network Virtualization Design Guide

---

## Table of Contents

1	Overview.....	4
2	Introduction to Network Virtualization.....	8
2.1	Overview of NSX Network Virtualization Solution.....	8
2.1.1	Data Plane .....	8
2.1.1.i	NSX Logical Networking Components .....	9
2.1.1.ii	NSX Services Platform .....	9
2.1.2	Control Plane .....	11
2.1.3	Management Plane and Consumption Platforms.....	11
3	NSX Functional Components .....	13
3.1.1	NSX Manager.....	13
3.1.2	Controller Cluster .....	15
3.1.3	VXLAN Primer.....	18
3.1.4	ESXi Hypervisors with VDS .....	21
3.1.5	NSX Edge Services Gateway .....	22
3.1.6	Transport Zone.....	24
3.1.7	NSX Distributed Firewall (DFW).....	25
4	NSX Functional Services.....	33
4.1	Multi-Tier Application Deployment Example .....	33
4.2	Logical Switching .....	33
4.2.1	Replication Modes for Multi-Destination Traffic.....	35
4.2.2	Populating the Controller Tables .....	41
4.2.3	Unicast Traffic (Virtual to Virtual Communication) .....	43
4.2.4	Unicast Traffic (Virtual to Physical Communication) .....	45
4.3	Logical Routing .....	49
4.3.1	Logical Routing Components .....	50
4.3.2	Routing Capabilities in NSX .....	56

4.3.3	OSPF and NSX Connectivity Options: .....	59
4.3.4	BGP and NSX Connectivity Options: .....	64
4.3.5	Enterprise Routing Topology.....	69
4.4	Logical Firewalling and Security Services .....	74
4.4.1	Network Isolation.....	74
4.4.2	Network Segmentation.....	75
4.4.3	Taking Advantage of Abstraction .....	77
4.4.4	Advanced Security Service Insertion, Chaining and Steering.....	77
4.4.5	Consistent Visibility and Security Across Physical and Virtual.....	79
4.4.6	Introduction to Service Composer.....	79
4.4.7	Micro-Segmentation with NSX DFW and Implementation .....	89
4.5	Logical Load Balancing.....	94
4.6	Virtual Private Network (VPN) Services.....	98
4.6.1	L2 VPN .....	98
4.6.2	L3 VPN .....	100
5	NSX Design Considerations .....	101
5.1	Topology Independent Design with NSX .....	101
5.2	VLAN Connectivity with NSX .....	103
5.3	NSX Deployment Considerations .....	107
5.3.1	Cluster Types and Characteristics .....	108
5.3.2	vCenter Design with NSX.....	110
5.3.3	VDS Design in an NSX Domain .....	112
5.3.4	VDS Uplinks Connectivity NSX Design Considerations .....	113
5.3.5	ESXi Host Traffic Types .....	117
5.3.6	Edge Design and Deployment Considerations .....	122
5.3.7	NSX Edge Deployment Considerations .....	123
5.3.8	DC Cluster Configurations & Sizing with NSX .....	143
5.4	Design Consideration for NSX Security Services .....	150
5.4.1	Preparing Security Services for Datacenter .....	151
5.4.2	Determining Policy Model .....	155
5.4.3	Consideration for creating Groups and Policies .....	158
5.4.4	Deployment Models .....	164
6	Conclusion.....	167

## Intended Audience

This document is targeted toward virtualization and network architects interested in deploying VMware® NSX network virtualization solution in a vSphere environment.

### Revision History:

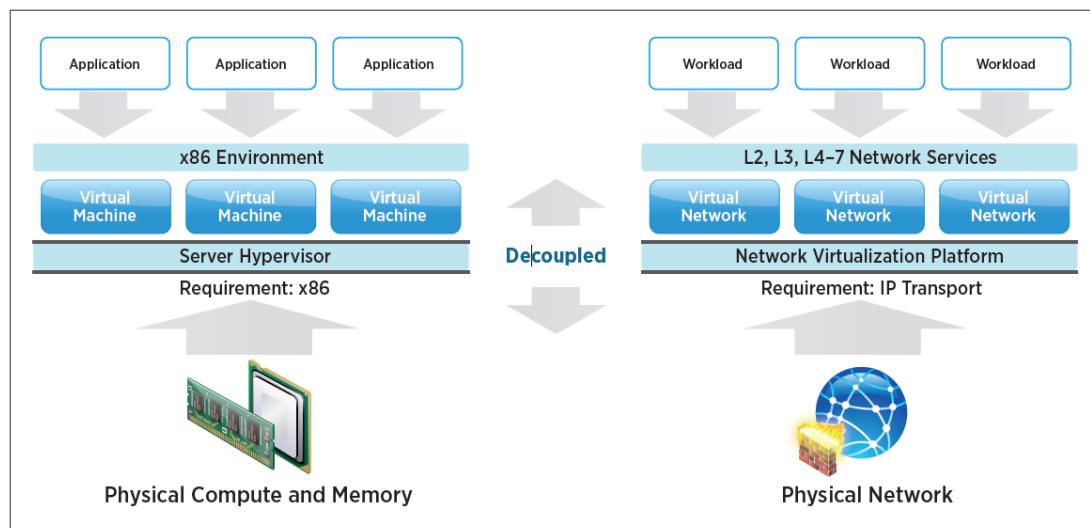
Version	Updates	Comments
2.1	None	First Release
3.0	NSX 6.2 Release & vSphere 6.0	Relevant Functional Update
	Edge Cluster Design	Cluster Sizing & Capacity
	Routing Design	Routing Protocol Interaction, Timers & Defaults
	Security Services	Service Composer and Policy Design
	Other	Feedback, technical clarity, refined recommendations
3.1	Your Feedback	Welcome

## 1 Overview

IT organizations have gained significant benefits as a direct result of server virtualization. Tangible advantages of server consolidation include reduced physical complexity, increased operational efficiency, and simplified dynamic re-purposing of underlying resources. These technology solutions have delivered on their promise of helping IT to quickly and optimally meet the needs of increasingly dynamic business applications.

VMware's Software Defined Data Center (SDDC) architecture moves beyond the server, extending virtualization technologies across the entire physical data center infrastructure. VMware NSX, the network virtualization platform, is a key product in the SDDC architecture. With VMware NSX, virtualization now delivers for networking what it has already delivered for compute. Traditional server virtualization programmatically creates, snapshots, deletes, and restores virtual machines (VMs); similarly, network virtualization with VMware NSX programmatically creates, snapshots, deletes, and restores software-based virtual networks. The result is a completely transformative approach to networking, enabling orders of magnitude better agility and economics while also vastly simplifying the operational model for the underlying physical network.

NSX is a completely non-disruptive solution which can be deployed on any IP network from any vendor – both existing traditional networking models and next generation fabric architectures. The physical network infrastructure already in place is all that is required to deploy a software-defined data center with NSX.

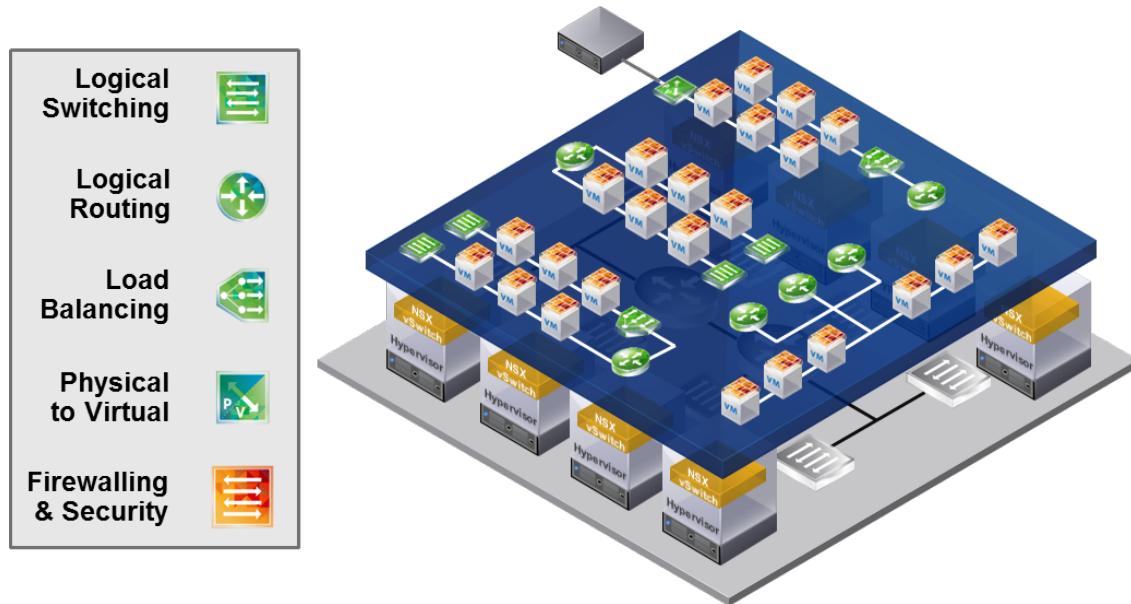


**Figure 1 - Server and Network Virtualization Analogy**

Figure 1 draws an analogy between compute and network virtualization. With server virtualization, a software abstraction layer (i.e., server hypervisor) reproduces the familiar attributes of an x86 physical server (e.g., CPU, RAM, Disk, NIC) in software. This allows components to be programmatically

assembled in any arbitrary combination to produce a unique VM in a matter of seconds.

With network virtualization, the functional equivalent of a “network hypervisor” reproduces layer 2 to layer 7 networking services (e.g., switching, routing, firewalling, and load balancing) in software. These services can then be programmatically assembled in any arbitrary combination, producing unique, isolated virtual networks in a matter of seconds.



**Figure 2: Network Virtualization Abstraction Layer and Underlying Infrastructure**

Where VMs are independent of the underlying x86 platform and allow IT to treat physical hosts as a pool of compute capacity, virtual networks are independent of the underlying IP network hardware. IT can thus treat the physical network as a pool of transport capacity that can be consumed and repurposed on demand. This abstraction is illustrated in Figure 2. Unlike legacy architectures, virtual networks can be provisioned, changed, stored, deleted, and restored programmatically without reconfiguring the underlying physical hardware or topology. By matching the capabilities and benefits derived from familiar server and storage virtualization solutions, this transformative approach to networking unleashes the full potential of the software-defined data center.

With VMware NSX, existing networks are immediately ready to deploy a next-generation software defined data center. This paper will highlight the range of functionality provided by the VMware NSX for vSphere architecture, exploring design factors to consider to fully leverage and optimize existing network investments.

## NSX Primary Use Cases

Customers are using NSX to drive business benefits as shown in the figure below. The main themes for NSX deployments are Security, IT automation and Application Continuity.

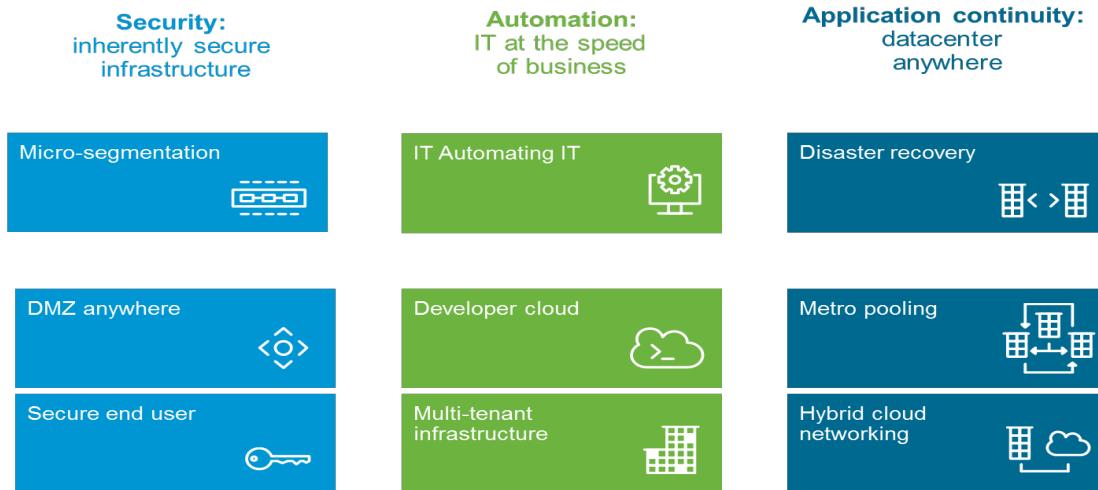


Figure 3: NSX Use Cases

- **Security:**

NSX can be used to create a secure infrastructure, which can create a zero-trust security model. Every virtualized workload can be protected with a full stateful firewall engine at a very granular level. Security can be based on constructs such as MAC, IP, ports, vCenter objects and tags, active directory groups, etc. Intelligent dynamic security grouping can drive the security posture within the infrastructure.

NSX can be used in conjunction with 3<sup>rd</sup> party security vendors such as Palo Alto Networks, Checkpoint, Fortinet, or McAfee to provide a complete DMZ like security solution within a cloud infrastructure.

NSX has been deployed widely to secure virtual desktops to secure some of the most vulnerable workloads, which reside in the data center to prohibit desktop-to-desktop hacking.

- **Automation:**

VMware NSX provides a full RESTful API to consume networking, security and services, which can be used to drive automation within the infrastructure. IT admins can reduce the tasks and cycles required to provision workloads within the datacenter using NSX.

NSX is integrated out of the box with automation tools such as vRealize automation, which can provide customers with a one-click deployment option for an entire application, which includes the compute, storage, network, security and L4-L7 services.

Developers can use NSX with the OpenStack platform. NSX provides a neutron plugin that can be used to deploy applications and topologies via OpenStack

- **Application Continuity:**

NSX provides a way to easily extend networking and security up to eight vCenters either within or across data center. In conjunction with vSphere 6.0 customers can easily vMotion a virtual machine across long distances and NSX will ensure that the network is consistent across the sites and ensure that the firewall rules are consistent. This essentially maintains the same view across sites.

NSX Cross vCenter Networking can help build active – active data centers. Customers are using NSX today with VMware Site Recovery Manager to provide disaster recovery solutions. NSX can extend the network across data centers and even to the cloud to enable seamless networking and security.

The use cases outlined above are a key reason why customers are investing in NSX. NSX is uniquely positioned to solve these challenges as it can bring networking and security closest to the workload itself and carry the policies along with the workload.

## 2 Introduction to Network Virtualization

### 2.1 Overview of NSX Network Virtualization Solution

An NSX deployment consists of a data plane, control plane, and management plane, as shown in Figure 4.

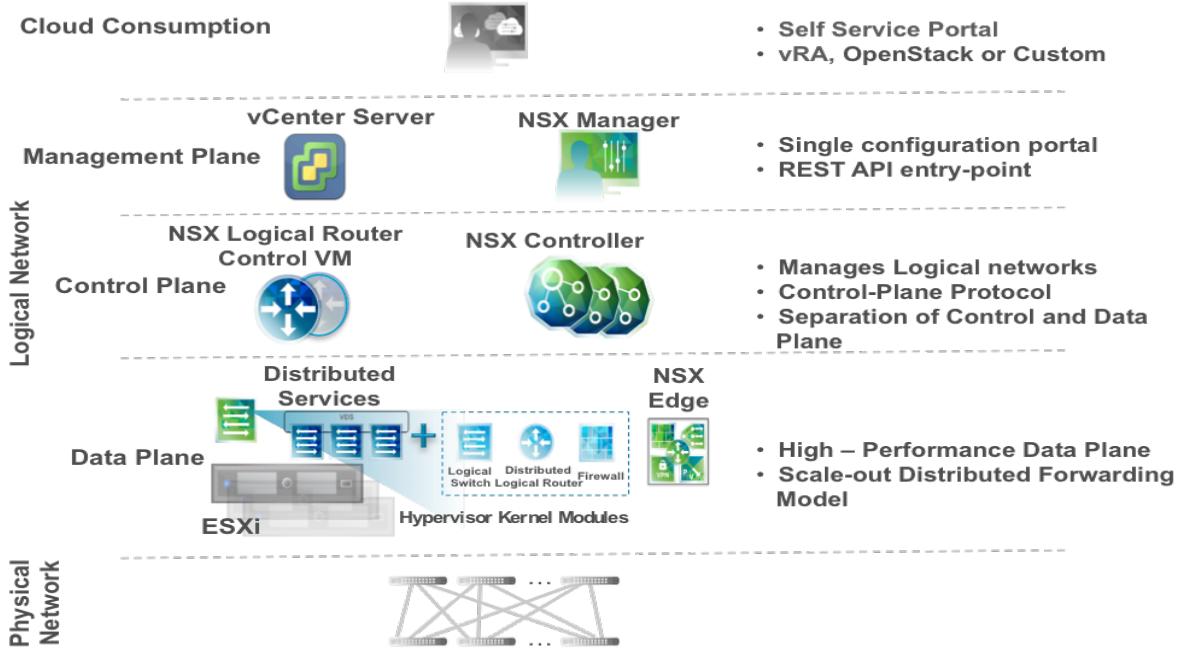


Figure 4 - NSX Components

The NSX architecture has built in separation of data, control, and management layers. The NSX components that maps to each layer and each layer's architectural properties are shown in above Figure 4. This separation allows the architecture to grow and scale without impacting workload. Each layer and its specific overview are described below.

#### 2.1.1 Data Plane

The NSX data plane is implemented by the NSX vSwitch. The vSwitch in NSX for vSphere is based on the VDS with additional components added to enable rich services. The add-on NSX components include kernel modules distributed as VMware installation bundles (VIBs). These modules run within the hypervisor kernel, providing services including distributed routing, distributed firewall, and VXLAN to VLAN bridging.

The NSX VDS abstracts the physical network, providing access-level switching in the hypervisor. This is central to network virtualization as it enables logical networks that are independent of physical constructs (e.g., VLANs).

The NSX vSwitch enables support for overlay networking with the use of the VXLAN protocol and centralized network configuration. Overlay networking with NSX enables the following capabilities:

- Creation of a flexible logical layer 2 (L2) overlay over existing IP networks on existing physical infrastructure.
- Agile provisioning of communication – both east–west and north–south – while maintaining isolation between tenants.
- Application workloads and VMs that are agnostic of the overlay network, operating as if they were connected to a physical network.
- Massive scalability of hypervisors.

Operational features – port mirroring, NetFlow/IPFIX, Traceflow, configuration backup and restore, Network Health Check, QoS, and LACP – provide a comprehensive toolkit for traffic management, monitoring, and troubleshooting within a virtual network.

The data plane also consists of gateway devices that can provide communication from the logical networking space to the physical network (e.g., VXLAN to VLAN). This functionality can happen at either L2 (NSX bridging) or at L3 (NSX routing).

### i. NSX Logical Networking Components

NSX provides a faithful reproduction of network & security services in software.



Figure 5 - NSX Logical Network Services

**Switching:** Logical switching enables extension of a L2 segment / IP subnet anywhere in the fabric independent of the physical network design.

**Routing:** Routing between IP subnets can be done in the logical space without traffic leaving the hypervisor; routing is performed directly in the hypervisor kernel with minimal CPU / memory overhead. This distributed logical routing (DLR) provides an optimal data path for traffic within the virtual infrastructure (i.e., east-west communication). Additionally, the NSX Edge provides an ideal centralized point for seamless integration with the physical network infrastructure to handle communication with the external network (i.e., north-south communication) with ECMP-based routing.

**Connectivity to physical networks:** L2 and L3 gateway functions are supported within NSX to provide communication between workloads deployed in logical and physical spaces.

### ii. NSX Services Platform

NSX is a services platform that enables both security services and distributed firewall.

These services are either a built-in part of NSX or available from 3<sup>rd</sup> party vendors. Existing physical devices – examples include physical load balancers, firewalls, syslog collectors, and monitoring devices – can also be integrated with NSX. All services can be orchestrated via the NSX consumption model (explained in section 2.1.3 below).

#### 2.1.1.1 Networking and Edge Services

NSX provides built-in networking and edge services including logical load balancing, L2/L3 VPN services, edge firewalling, and DHCP/NAT.



Figure 6 - NSX Logical Network Services

**Edge Firewall:** Edge firewall services are part of the NSX Edge Services Gateway (ESG). The Edge firewall provides essential perimeter firewall protection which can be used in addition to a physical perimeter firewall. The ESG-based firewall is useful in developing PCI zones, multi-tenant environments, or dev-ops style connectivity without forcing the inter-tenant or inter-zone traffic onto the physical network.

**VPN:** L2 VPN, IPSEC VPN, and SSL VPN services to enable L2 and L3 VPN services. The VPN services provide critical use-case of interconnecting remote datacenters and users access.

**Logical Load-balancing:** L4-L7 load balancing with support for SSL termination. The load-balancer comes in two different form factors supporting in-line as well as proxy mode configurations. The load-balancer provides critical use case in virtualized environment, which enables devops style functionalities supporting variety of workload in topological independent manner.

**DHCP & NAT Services:** Support for DHCP servers and DHCP forwarding mechanisms; NAT services.

NSX also provides an extensible platform that can be used for deployment and configuration of 3<sup>rd</sup> party vendor services. Examples include virtual form factor load balancers (e.g., F5 BIG-IP LTM) and network monitoring appliances (e.g., Gigamon - GigaVUE-VM).

Integration of these services is simple with existing physical appliances such as physical load balancers and IPAM/DHCP server solutions.

#### 2.1.1.2 Security Services and Distributed Firewall

NSX includes built-in security services – distributed firewalling for east-west L2-L4 traffic, edge firewalling for north-south traffic, and SpoofGuard for validation of IP/MAC identity.

**Distributed Firewall** – Security enforcement is done directly at the kernel and vNIC level. This enables highly scalable firewall rule enforcement by avoiding bottlenecks on physical appliances. The firewall is distributed in kernel, minimizing CPU overhead while enabling line-rate performance.

NSX also provides an extensible framework, allowing security vendors to provide an umbrella of security services. Popular offerings include anti-virus/anti-malware/anti-bot solutions, L7 firewalling, IPS/IDS (host and network based) services, file integrity monitoring, and vulnerability management of guest VMs.

### 2.1.2 Control Plane

The NSX controller is a key part of the NSX control plane. In a vSphere environment with the vSphere Distributed Switch (VDS), the controller enables multicast free VXLAN and control plane programming of elements such as the Distributed Logical Routing (DLR).

Stability and reliability of data transport are central concerns in networking. The NSX controller is a part of the control plane; it is logically separated from all data plane traffic. To further enhance high availability and scalability, NSX controller nodes are deployed in a cluster of odd number instances.

In addition to controller, the control VM, provides the routing control plane that allows the local forwarding in ESXi and allows dynamic routing between ESXi and north-south routing provided by Edge VM. It is critical to understand that data plane traffic never traverses the control plane component.

### 2.1.3 Management Plane and Consumption Platforms

The NSX manager is the management plane for the NSX eco-system. NSX manager provides configuration and orchestration of:

- Logical networking components – logical switching and routing
- Networking and Edge services
- Security services and distributed firewall

Edge services and security services can be provided by either built-in components of NSX Manager or by integrated 3<sup>rd</sup> party vendors. NSX manager allows seamless orchestration of both built-in and external services.

All security services, whether built-in or 3<sup>rd</sup> party, are deployed and configured by the NSX management plane. The management plane provides a single window for viewing services availability. It also facilitates policy based service chaining, context sharing, and inter-service events handling. This simplifies the auditing of the security posture, streamlining application of identity-based controls. (e.g., AD, mobility profiles).

Consumption of built-in NSX features or integrated 3<sup>rd</sup> party vendor services is available through the vSphere Web UI. NSX manager also provides REST API

entry-points to automate consumption. This flexible architecture allows for automation of all configuration and monitoring aspects via any cloud management platform, security vendor platform, or automation framework.

Multiple out of the box (OOTB) integrations are currently available. Examples of VMware and 3<sup>rd</sup> party offerings are provided below.

VMware SDDC Product Suite:

- VMware vRealize Automation (vRA)
- VMware Log Insight (LI)
- VMware vRealize Operations Manager (vROps)
- VMware Integrated OpenStack (VIO)

3<sup>rd</sup> Party Integration:

- Arkin Visibility and Operations Platform
- Tufin Orchestration Suite for Firewall Management

### 3 NSX Functional Components

NSX logical networks leverage two types of access layer entities – the hypervisor access layer and the gateway access layer. The hypervisor access layer represents the point of attachment to the logical networks for virtual endpoints (e.g., VM, service-point). The gateway access layer provides L2 and L3 connectivity into the logical space for devices deployed in the physical network infrastructure.

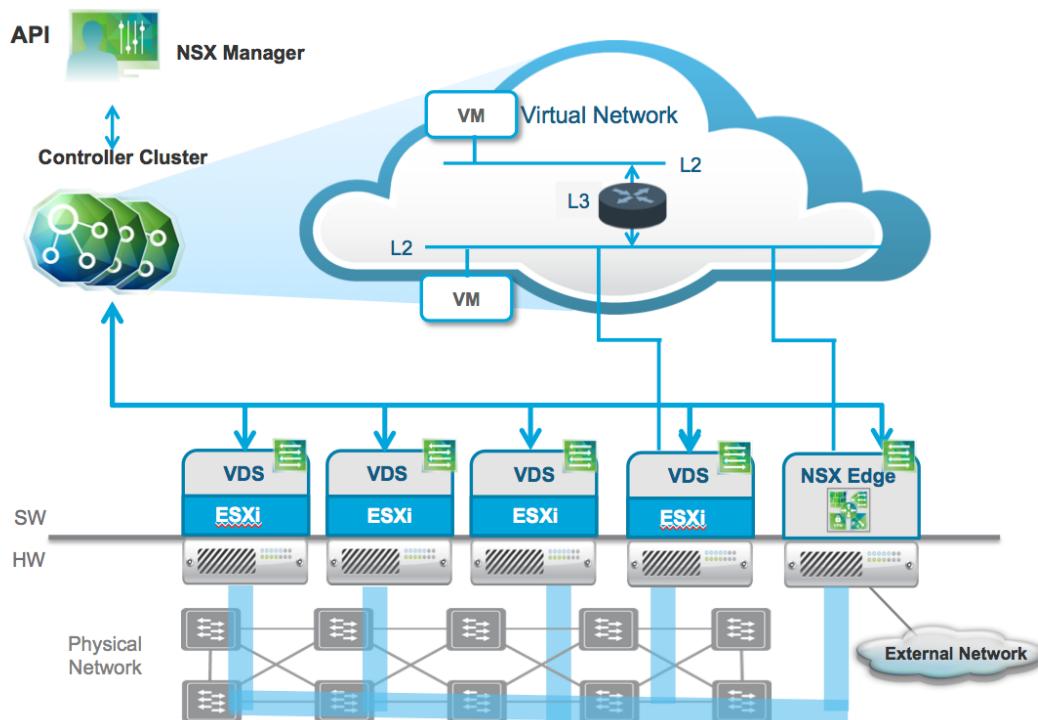


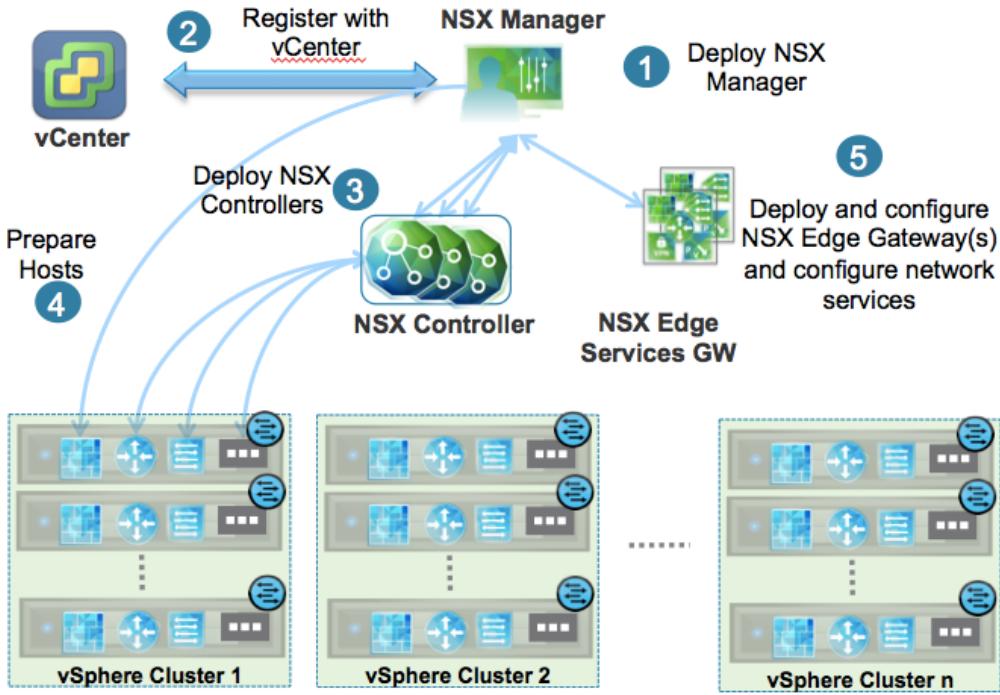
Figure 7 – NSX Functional Components

The NSX platform consists of multiple components, responsible for platform management, traffic control, and service delivery. The following sections detail their functional and operational specifics.

#### 3.1.1 NSX Manager

The NSX manager is the management plane virtual appliance. It serves as the entry point for REST API for NSX, which helps automate deployment and management of the logical networks.

In the NSX for vSphere architecture, the NSX manager is tightly connected to the vCenter Server managing the compute infrastructure. There is a 1:1 relationship between the NSX manager and vCenter. The NSX manager provides the networking and security plugin for the vCenter Web UI that enables administrators to configure and control NSX functionality.



**Figure 8 - NSX Manager Plugin Inside vSphere Web Client**

As highlighted in Figure 8, the NSX Manager is responsible for the deployment of the controller clusters and ESXi host preparation. The host preparation process installs various vSphere Installation Bundles (VIBs) to enable VXLAN, distributed routing, distributed firewall and a user world agent for control plane communications. The NSX manager is also responsible for the deployment and configuration of the NSX Edge services gateways and associated network services (load balancing, firewalling, NAT, etc.). This functionality will be described in greater detail in following sections.

The NSX manager also ensures security of the control plane communication of the NSX architecture. It creates self-signed certificates for the nodes of the controller cluster and ESXi hosts that should be allowed to join the NSX domain. The NSX manager installs those certificates to the ESXi hosts and the NSX controllers over a secure channel. Mutual authentication of NSX entities occurs by verifying the certificates. Once this mutual authentication is completed, control plane communication is encrypted.

SSL is disabled by default in NSX software release 6.0. In order to ensure confidentiality of the control plane communication, it is recommended to enable SSL. This can be accomplished through an API call. SSL is enabled by default from the 6.1 release onward.

Since the NSX Manager is a virtual machine and IP-based device, it is recommended to leverage standard vSphere functionalities (e.g., vSphere HA) to ensure that the NSX Manager can be dynamically moved should its ESXi hosts encounter a failure. Note that such a failure scenario would only impact the NSX

management plane; the already deployed logical networks would continue to operate seamlessly.

An NSX manager outage may affect only specific functionalities such as identity based firewall or flow monitoring collection.

NSX manager data (e.g., system configuration, events, audit log tables) can be backed up at any time by performing an on-demand backup from the NSX Manager GUI. It is also possible to schedule periodic backups to be performed (e.g., hourly, daily or weekly). Restoring a backup is only possible on a freshly deployed NSX manager appliance that can access one of the previously backed up instances.

The NSX manager requires IP connectivity to vCenter, controller, NSX Edge resources, and ESXi hosts. NSX manager typically resides in the same subnet (VLAN) as vCenter and communicates over the management network. This is not a strict requirement; NSX manager supports inter-subnet IP communication where design constraints require subnet separation from vCenter (e.g., security policy, multi-domain management).

The NSX manager vCPU and memory requirement planning is dependent of NSX release as shown below in Table 1.

NSX Release	vCPU	Memory	OS Disk
<b>6.1</b>	4	12 GB	60 GB
<b>6.2 Default</b>	4	16 GB	60 GB
<b>6.2 Large Scale</b>	8	24 GB	60 GB

Table 1 – NSX Manager Configuration Option

### 3.1.2 Controller Cluster

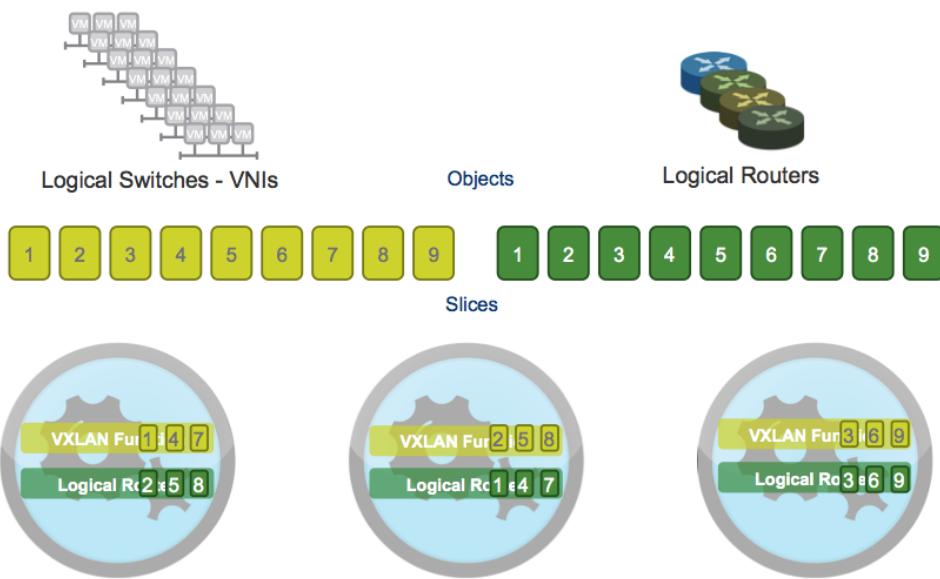
The controller cluster in the NSX platform is the control plane component responsible for managing the hypervisor switching and routing modules. The controller cluster consists of controller nodes that manage specific logical switches. The use of controller cluster in managing VXLAN based logical switches eliminates the need for multicast configuration at the physical layer for VXLAN overlay.

The NSX controller supports an ARP suppression mechanism, reducing the need to flood ARP broadcast requests across an L2 network domain where virtual machines are connected. The different VXLAN replication mode and the ARP

suppression mechanism will be discussed in more detail in the “Logical Switching” section.

For resiliency and performance, production deployments of controller VM should be in three distinct hosts. The NSX controller cluster represents a scale-out distributed system, where each controller node is assigned a set of roles that define the type of tasks the node can implement.

In order to increase the scalability characteristics of the NSX architecture, a slicing mechanism is utilized to ensure that all the controller nodes can be active at any given time.



**Figure 9 – Slicing Controller Cluster Node Roles**

Figure 9 illustrates the distribution of roles and responsibilities between different cluster nodes. This demonstrates how distinct controller nodes act as master for given entities such as logical switching, logical routing and other services. Each node in the controller cluster is identified by a unique IP address. When an ESXi host establishes a control-plane connection with one member of the cluster, a full list of IP addresses for the other members is passed down to the host. This enables establishment of communication channels with all members of the controller cluster, allowing the ESXi host to know at any given time which specific node is responsible for any given logical network.

In the case of failure of a controller node, the slices owned by that node are reassigned to the remaining members of the cluster. In order for this mechanism to be resilient and deterministic, one of the controller nodes is elected as a master for each role. The master is responsible for allocating slices to individual controller nodes, determining when a node has failed, and reallocating the slices to the other nodes. The master also informs the ESXi hosts about the failure of the cluster node so that they can update their internal node ownership mapping.

The election of the master for each role requires a majority vote of all active and inactive nodes in the cluster. This is the primary reason why a controller cluster must always be deployed with an odd number of nodes.

<b>Number of NSX Nodes in Cluster</b>	2	3
<b>Majority Number</b>	2	2
<b>Number of Nodes that can fail</b>	0	1

Figure 10 - Controller Nodes Majority Numbers

Figure 10 highlights the different majority number scenarios depending on the number of available controller nodes. In a distributed environment, node majority is required. During the failure of one the node, with only two nodes working in parallel, the majority number is maintained. If one of those two nodes were to fail or inter-node communication is lost (i.e., dual-active scenario), neither would continue to function properly. For this reason, NSX supports controller clusters with a minimum configuration of three nodes. In the case of second node failure the cluster will have only one node. In this condition controller reverts to read-only mode. In this mode, existing configuration should continue to work however any new modification to the configuration is not allowed.

NSX controller nodes are deployed as virtual appliances from the NSX manager UI. Each appliance communicates via a distinct IP address. While often located in the same subnet as the NSX manager, this is not a hard requirement. Each appliance must strictly adhere to the specifications in Table 2.

Per VM Configurations				
Controller VMs	vCPU	Reservation	Memory	OS Disk
3	4	2048 MHz	4 GB	20 GB

Table 2 – Controller Capacity Requirements

It is recommended to spread the deployment of cluster nodes across separate ESXi hosts. This helps ensure that the failure of a single host does not cause the loss of a majority number in the cluster. NSX does not natively enforce this design practice; leverage the native vSphere anti-affinity rules to avoid deploying more than one controller node on the same ESXi server. For more information on

how to create a VM-to-VM anti-affinity rule, please refer to the following KB article: <http://tinyurl.com/nzgv6hq>.

### 3.1.3 VXLAN Primer

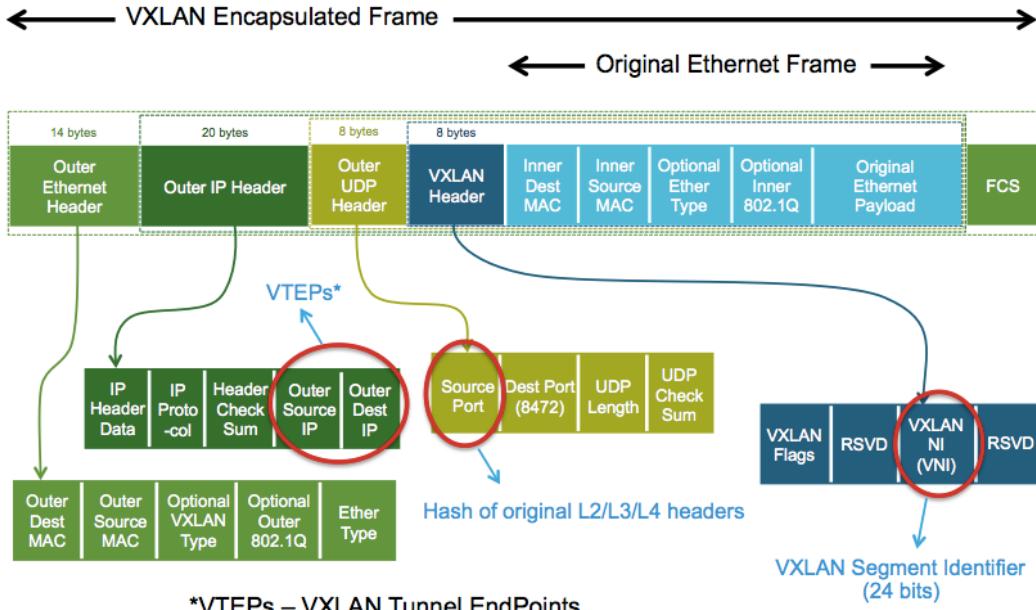
The deployment of overlay technologies has become very popular because of their capabilities in decoupling connectivity in the logical space from the physical network infrastructure. Devices connected to logical networks can leverage the entire set of network functions previously highlighted in Figure 5, independent of the underlying physical infrastructure configuration. The physical network effectively becomes a backplane used to transport overlay traffic.

This decoupling effect help solve many challenges traditional data center deployments are currently facing:

- **Agile/Rapid Application Deployment:** Traditional networking design is a bottleneck, preventing the rollout of new application at the pace that business is demanding. Overhead required to provision the network infrastructure in support of a new application often is counted in days if not weeks.
- **Workload Mobility:** Compute virtualization enables mobility of virtual workloads across different physical servers connected to the data center network. In traditional data center designs, this requires extension of L2 domains (VLANs) across the entire data center network infrastructure. This affects the overall network scalability and potentially jeopardizes the resiliency of the design.
- **Large Scale Multi-Tenancy:** The use of VLANs as a means of creating isolated networks limits the maximum number of tenants that can be supported (i.e., 4094 VLANs). While this value may currently be sufficient for typical enterprise deployments, it is becoming a serious bottleneck for many cloud providers.

Virtual Extensible LAN (VXLAN) has become the “de-facto” standard overlay technology and is embraced by multiple vendors; VMware in conjunction with Arista, Broadcom, Cisco, Citrix, Red Hat, and others developed it. Deploying VXLAN is key to building logical networks that provide L2 adjacency between workloads without the issues and scalability concerns found in traditional L2 technologies.

As shown in Figure 11, VXLAN is an overlay technology encapsulating the original Ethernet frames generated by workloads – virtual or physical – connected to the same logical layer 2 segment, usually named Logical Switch (LS).



\*VTEPs – VXLAN Tunnel EndPoints

**Figure 11 - VXLAN Encapsulation**

Figure 11 details the VXLAN packet format. Additional thoughts on the protocol include:

- VXLAN is a L2 over L3 (L2oL3) encapsulation technology. The original Ethernet frame generated by a workload is encapsulated with external VXLAN, UDP, IP and Ethernet headers to ensure it can be transported across the network infrastructure interconnecting the VXLAN endpoints (e.g., ESXi hosts).
- Scaling beyond the 4094 VLAN limitation on traditional switches has been solved by leveraging a 24-bit identifier, named VXLAN Network Identifier (VNI), which is associated to each L2 segment created in logical space. This value is carried inside the VXLAN Header and is normally associated to an IP subnet, similarly to what traditionally happens with VLANs. Intra-IP subnet communication happens between devices connected to the same virtual network/logical switch.

---

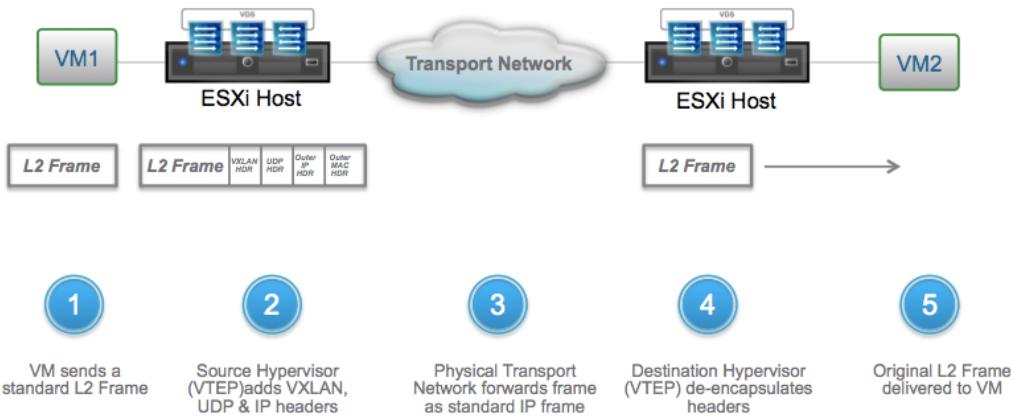
Note – The terms “VXLAN segment”, “Virtual Network” (VN) and “Logical Switch” (LS) all refer to the logical layer 2 domain created in the logical network space and will be used interchangeably in this document.

---

- Hashing of the L2/L3/L4 headers present in the original Ethernet frame is performed to derive the source port value for the external UDP header. This is important to ensure load balancing of VXLAN traffic across equal cost paths available inside the transport network infrastructure.
- NSX uses 8472 as destination port value for the external UDP header. This differs from the IANA assigned number for VXLAN that is 4789, as described in RFC 7348 - <http://tools.ietf.org/html/rfc7348#page-19>. This value can be modified in an NSX deployment via a REST API call. In order

to avoid a data-plane outage, ensure that all the ESXi hosts are running an NSX release and that the configuration in the physical network (e.g., access-list, firewall policies) is properly updated.

- The source and destination IP addresses used in the external IP header uniquely identify the ESXi hosts originating and terminating the VXLAN encapsulation of frames. Those are usually referred to as VXLAN Tunnel Endpoints (VTEPs).
- Encapsulating the original Ethernet frame into a UDP packet increases the size of the IP packet. For this reason, increasing the MTU to a minimum of 1600 bytes is recommended for all interfaces in the physical infrastructure that will carry the frame. The MTU for the VDS uplinks of the ESXi hosts performing VXLAN encapsulation is automatically increased when preparing the host for VXLAN (from the NSX Manager UI).



**Figure 12 - L2 Communication Leveraging VXLAN Encapsulation**

Figure 12 describes the high level steps required to establish L2 communication between virtual machines connected to different ESXi hosts leveraging the VXLAN overlay functionalities.

- VM1 originates a frame destined for the VM2 part of the same L2 logical segment/IP subnet.
- The source ESXi host identifies the ESXi host (VTEP) where the VM2 is connected and encapsulates the frame before sending it into the transport network.
- The transport network is only required to enable IP communication between the source and destination VTEPs.
- The destination ESXi host receives the VXLAN frame, decapsulates it, and identifies the L2 segment it belongs to, leveraging the VNI value inserted in the VXLAN header by the source ESXi host.
- The frame is delivered to VM2.

More detailed information on how L2 communication can be implemented when leveraging VXLAN will be provided in the Logical Switching section, together with a discussion about specific VXLAN control and data plane enhancements provided by NSX.

### 3.1.4 ESXi Hypervisors with VDS

As previously described, the VDS is a building block for the overall NSX architecture. VDS is now available on all VMware ESXi hypervisors, so its control and data plane interactions are central to the entire NSX architecture.

For more information on VDS vSwitch and best practices for its deployment in a vSphere environment please refer to the following paper:

<http://www.vmware.com/files/pdf/techpaper/vsphere-distributed-switch-best-practices.pdf>

#### 3.1.4.1 User Space and Kernel Space

As shown in Figure 13, each ESXi host has a user space and a kernel space.

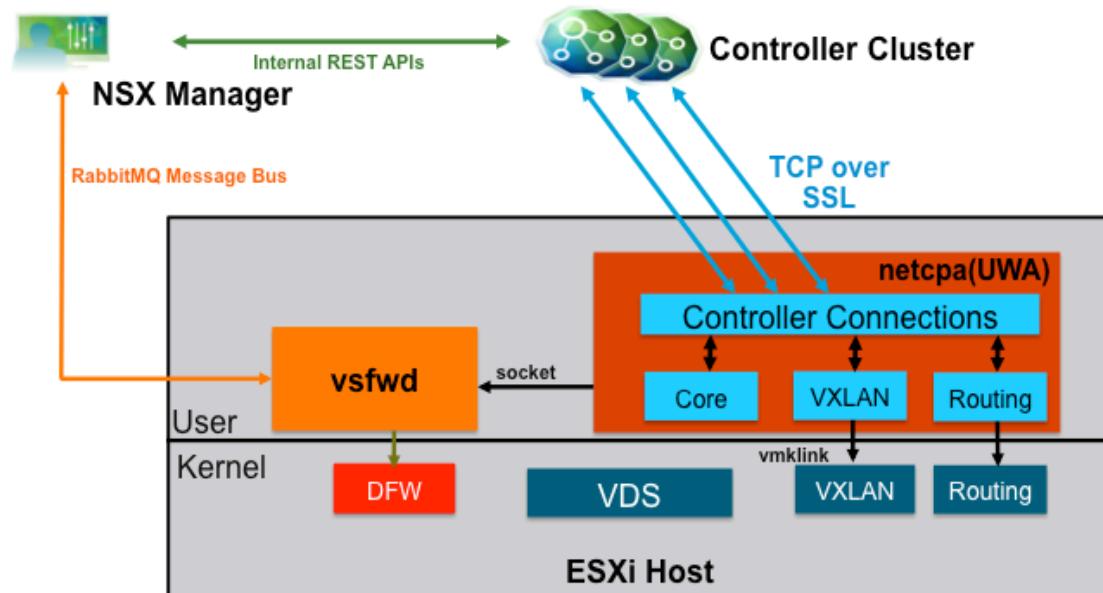


Figure 13 - ESXi Host User and Kernel Space Diagram

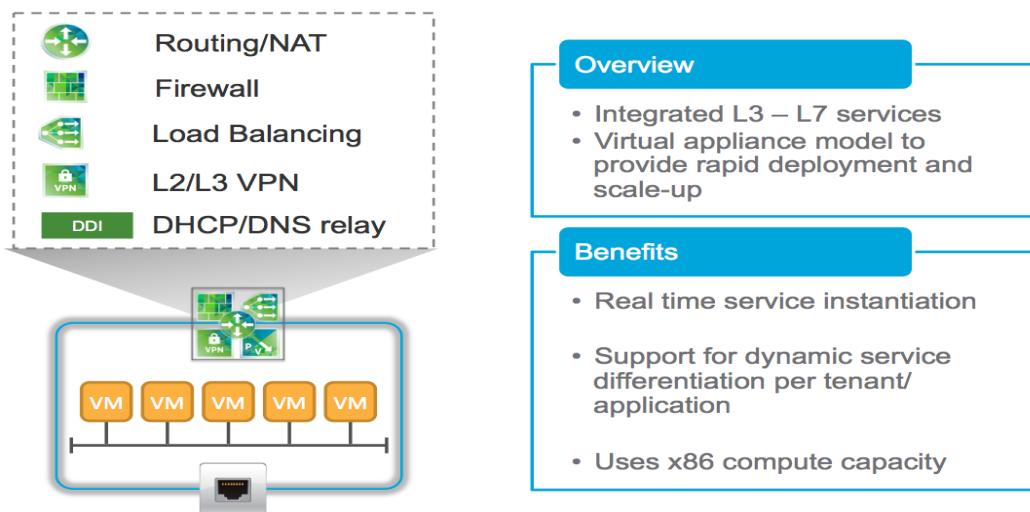
The user space consists of software components that provide the control plane communication path to the NSX manager and the controller cluster nodes.

A RabbitMQ message bus is leveraged for communication between the vsfwd (RMQ client) and RMQ server process hosted on the NSX manager. The message bus is used by the NSX manager to send various information to the ESXi hosts, including policy rules that need to be programmed on the distributed firewall in the kernel, private keys and host certificates to authenticate the communication between hosts and controllers, controller node IP addresses, and requests to create/delete distributed logical router instances.

The user world agent process (netcpa) establishes TCP over SSL communication channels to the controller cluster nodes. Controller nodes leverage this control-plane channel with the ESXi hypervisors to populate local tables (e.g., MAC address table, ARP table, and VTEP table) to keep track of where the workloads are connected in the deployed logical networks.

### 3.1.5 NSX Edge Services Gateway

The NSX Edge is a multi-function, multi-use VM appliance for network virtualization.



**Figure 14 - Services Provided by NSX Edge**

Figure 14 highlights the logical services provided by the NSX Edge. Its deployment varies based on its use, places in the topology, elastic performance requirements, and stateful services such as load balancer, firewall, VPN, and SSL. Edge VM supports two distinct modes of operation. The first one is active-standby in which all services are available. The second mode is ECMP mode, which provides high bandwidth (up to eight Edge VM supporting up to 80 GB traffic per DLR) and faster convergence. In ECMP mode, only routing service is available. Stateful services cannot be supported due to asymmetric routing inherent in ECMP-based forwarding. Each service is described briefly below, with additional detailed design choices discussed in “[Edge Design and Deployment Considerations](#)” section.

**Routing:** The NSX Edge provides centralized on-ramp/off-ramp routing between the logical networks deployed in the NSX domain and the external physical network infrastructure. The NSX Edge supports various dynamic routing protocols (e.g., OSPF, iBGP, eBGP) and can also leveraging static routing. The routing capability supports two models, active-standby stateful services and ECMP. The routing functionality is covered in greater detail in the “Logical Routing” section.

**Network Address Translation (NAT):** Network Address Translation enables dev-ops topology that can be enabled on-demand. The NAT is an integral part of load balancer functionality. NAT can be performed for traffic flowing through the Edge. Both source and destination NAT are supported.

**Firewall:** The NSX Edge supports stateful firewalling capabilities, complementing the Distributed Firewall (DFW) enabled in the kernel of the ESXi hosts. While the DFW is primarily utilized to enforce security policies for communication between workloads connected to logical networks (i.e., east-west traffic), the NSX Edge firewall is mainly filters communications between the logical space and the external physical network (i.e., north-south traffic).

**Load Balancing:** The NSX Edge can perform load-balancing services for server farms of workloads deployed in the logical space. The load balancing functionalities natively supported in the Edge cover most of the typical requirements found in real-life deployments.

**L2 and L3 Virtual Private Networks (VPNs):** The NSX Edge provides both L2 and L3 VPN capabilities. L2 VPN is usually positioned to extend L2 domains between geographically dispersed datacenters in support of use cases including compute resources bursting and hybrid cloud deployments. Common use cases for L3 VPNs include IPSec site-to-site connectivity and SSL VPN connectivity for access to private networks behind the NSX Edge.

**DHCP, DNS and IP Address Management (DDI):** The NSX Edge supports DNS relay functionalities and acts as a DHCP server, providing IP addresses, default gateway, DNS servers, and search domains parameters to workloads connected to the logical networks. NSX release 6.1 introduces support for DHCP Relay on the NSX Edge, allowing a centralized and remotely located DHCP server to provide IP addresses to the workloads across the logical networks.

The NSX manager offers deployment-specific form factors to be used depending on the functionalities that need to be enabled. These are detailed in Figure 15.



Edge Services Gateway Form	vCPU	Memory MB	Specific Usage
X-Large	6	8192	Suitable for L7 High Performance LB
Quad-Large	4	1024	Suitable for high performance ECMP and FW deployment
Large	2	1024	Small DC & Single Service
Compact	1	512	Small Deployments or Single Service use or PoC

Figure 15 - NSX Edge Form Factors

The NSX Edge form factor can be changed from after its initial deployments via UI or API calls, though this will cause a small service outage. The sizing

consideration for various deployment types along with failover and resilient design choices can be found in the “NSX Design Considerations” section.

### 3.1.6 Transport Zone

A Transport Zone defines a collection of ESXi hosts that can communicate with each other across a physical network infrastructure. This communication happens over one or more interfaces defined as VXLAN Tunnel Endpoints (VTEPs).

A Transport Zone extends across one or more ESXi clusters and commonly defines a span of logical switches. The relationship existing between the Logical Switch, VDS, and Transport Zone is central to this concept.

A VDS may span multiple ESXi hosts, and it is possible to dynamically add or remove single ESXi hosts from a specific VDS. In a practical NSX deployment, it is very likely that multiple VDS are defined in a given NSX Domain. Figure 16 shows a scenario where a “Compute-VDS” spans across all the ESXi hosts belonging to compute clusters, and a separate “Edge-VDS” extends across ESXi hosts in the edge clusters.

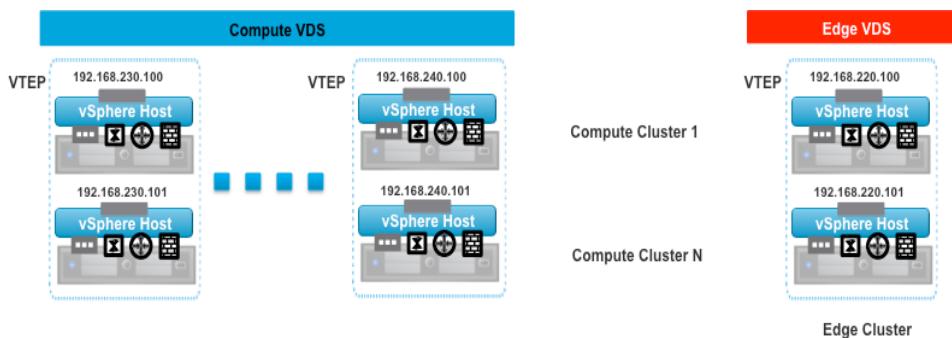
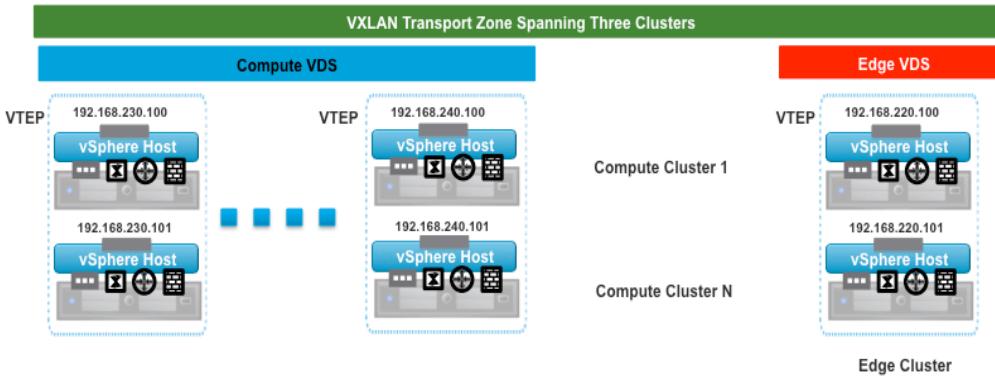


Figure 16 - Separate VDS spanning ESXi Clusters

A Logical Switch can extend across multiple VDS. Referring to the previous example, a given Logical Switch can provide connectivity for VMs that are connected to the Compute Clusters or to the Edge Cluster. A Logical Switch is always created as part of a specific Transport Zone. This implies that normally the Transport Zone extends across all the ESXi clusters and defines the extension of a Logical Switch, as highlighted in Figure 17.



**Figure 17 - Transport Zone Spanning VDS and ESXi Clusters**

More considerations and recommendations on VDS design in an NSX domain can be found in the “VDS Design in an NSX Domain” section.

### 3.1.7 NSX Distributed Firewall (DFW)

The NSX DFW provides L2-L4 stateful firewall services to any workload in the NSX environment. DFW runs in the kernel space and provides near line rate network traffic protection. DFW performance and throughput scale linearly through addition of new ESXi hosts.

DFW is activated as soon as the host preparation process is completed. If a VM does not require DFW service, it can be added in the exclusion list functionality. By default, NSX Manager, NSX Controllers, and Edge services gateways are automatically excluded from DFW function.

One DFW instance is created per VM vNIC; for example, if a new VM with 3 vNICs is created, 3 instances of DFW will be allocated to this VM. Configuration of these DFW instances can be identical or different based on the “apply to” setting. When a DFW rule is created, the user can select a Point of Enforcement (PEP) for this rule, with options varying from Logical Switch to vNIC. By default, “apply to” is not selected, so DFW rules are propagated down to all the ESXi hosts which are part of the NSX domain that have been prepared for enforcement and applied to all the connected virtual machines.

DFW policy rules can be written in 2 ways, using L2 rules (Ethernet) or L3/L4 rules.

- L2 rules are mapped to L2 in the OSI model; only MAC addresses can be used in the source and destination fields and only L2 protocols can be used in the service fields (e.g., ARP).
- L3/L4 rules are mapped to L3/L4 in the OSI model; policy rules can be written using IP addresses and TCP/UDP ports.

It is important to remember that L2 rules are always enforced before L3/L4 rules. If the L2 default policy rule is modified to ‘block’, and then all L3/L4 traffic will be blocked as well by DFW (e.g., pings would fail).

While DFW is an NSX component designed to protect workload-to-workload network traffic, either virtual-to-virtual or virtual-to-physical (i.e., east-west traffic),

since its policy enforcement is applied to the vNICs of the VMs, it could also be used to prevent communication between the VMs and the external physical network infrastructure. DFW is fully complementary with NSX Edge services gateway that provides centralized firewall capability. ESG is typically used to protect north-south traffic and as such is the first entry point to the Software-Defined Data Center.

The distinct application of DFW and ESG is depicted in the Figure 18.

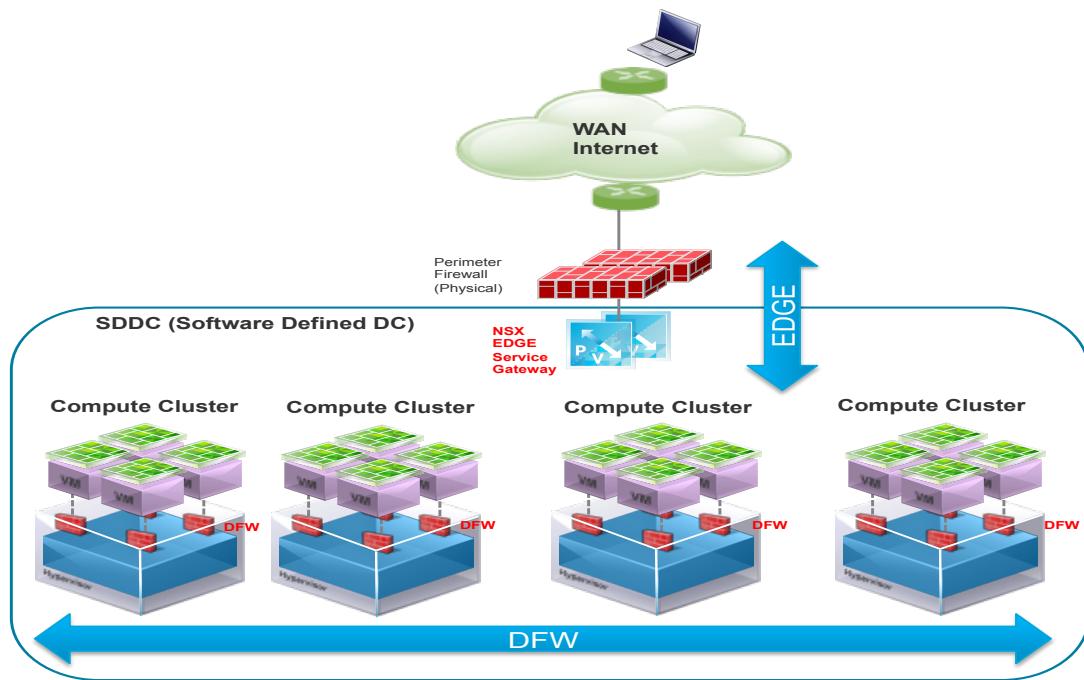


Figure 18 – DFW and Edge Service Gateway Traffic Protection

NSX DFW operates at the VM vNIC level; therefore, a VM is always protected irrespective of topology. VMs can be connected to a VDS VLAN-backed port-group or to a Logical Switch (i.e., VXLAN-backed port-group). ESG Firewall can also be used to protect workloads sitting on physical servers and appliances (e.g., NAS).

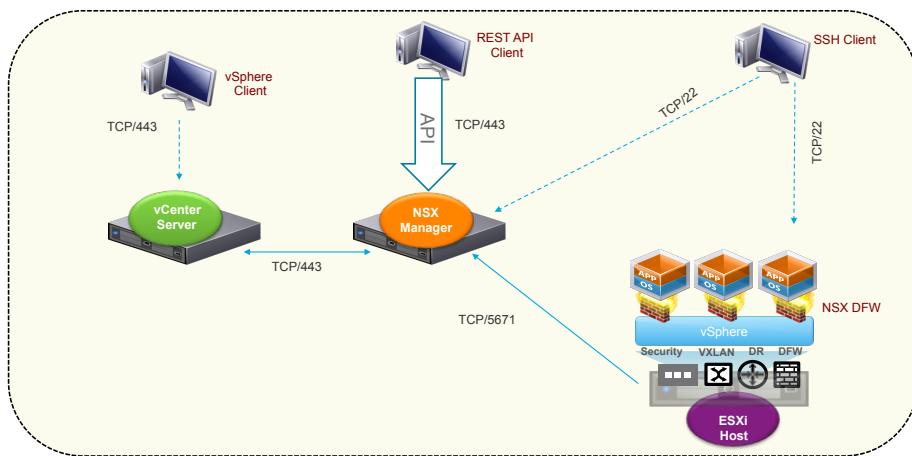
The DFW system architecture is based on 3 distinct entities, each with a clearly defined role.

- **vCenter Server:** vCenter Server is the management plane of the solution. DFW policy rules are created in the vSphere Web client. Any vCenter container can be used in the source/destination field of the policy rule: cluster, VDS port-group, Logical Switch, VM, vNIC, Resource Pool, etc.
- **NSX Manager:** NSX manager is the control plane of the solution. It receives rules from the vCenter Server and stores them in its central database. NSX manager then pushes DFW rules down to all ESXi hosts that have been prepared for enforcement. Backup of DFW security policy rule table is performed each time the table is modified and published. NSX

manager can also receive DFW rules directly from REST API calls in deployments where a cloud management system is used for security automation.

- **ESXi Host:** ESXi host is the data plane of the solution. DFW rules are received from the NSX manager and translated into kernel space for real-time execution. VM network traffic is inspected and enforced per ESXi host. As an example, VM1 is located on ESXi host 1 and sends packets to VM2 that is located on ESXi host 2. Policy enforcement is done on ESXi host 1 for egress traffic when packets leave VM1 and then on ESXi host 2 for ingress traffic destined to VM2.

The SSH client depicted in Figure 19 can access the ESXi host CLI to perform specific DFW related troubleshooting.



**Figure 19 – NSX DFW System Architecture and Communications Channels**

When vCenter containers are used in the DFW policy rules, VMtools must be installed on the guest VMs. VMtools has visibility of the IP address of the VM – whether dynamically provided through DHCP or statically configured on the VM by the administrator – and provides this information down to the DFW engine that operates based on MAC, IP, and TCP/UDP port fields.

If the DFW policy rules use only IP information (e.g., host IP, subnet IP, or IP sets) then the presence of VMtools on guest VM is not required.

---

**Note:** With NSX release 6.2, VMtools is not mandatory. See the “Deploying Distributed Firewall” section for further details.

---

The DFW function is activated when a host is prepared for enforcement. During this operation, a kernel VIB is loaded into the hypervisor. This VIB is known as the VMware Internetworking Service Insertion Platform (**VSIP**).

VSIP is responsible for all data plane traffic protection – it is the DFW engine by itself – and runs at near line-rate speed.

A DFW instance is created per VM vNIC. This instance is located between the VM and the Virtual Switch (i.e., VDS port-group VLAN-backed or Logical Switch). DVfilter slot 2 is allocated for the DFW instance. All ingress and egress packets to and from this VM must pass through the DFW.

A set of daemons called **vsfwd** runs permanently on the ESXi host and performs the following tasks:

- Interact with NSX Manager to retrieve DFW policy rules.
- Gather DFW statistics information and send them to the NSX manager.
- Send audit logs information to the NSX manager.

The communication path between the vCenter Server and the ESXi host uses the vpxa process on the ESXi host. This is only used for vSphere related purposes, including VM creation, storage modification, and NSX manager IP address distribution. This communication is not used for general DFW operation.

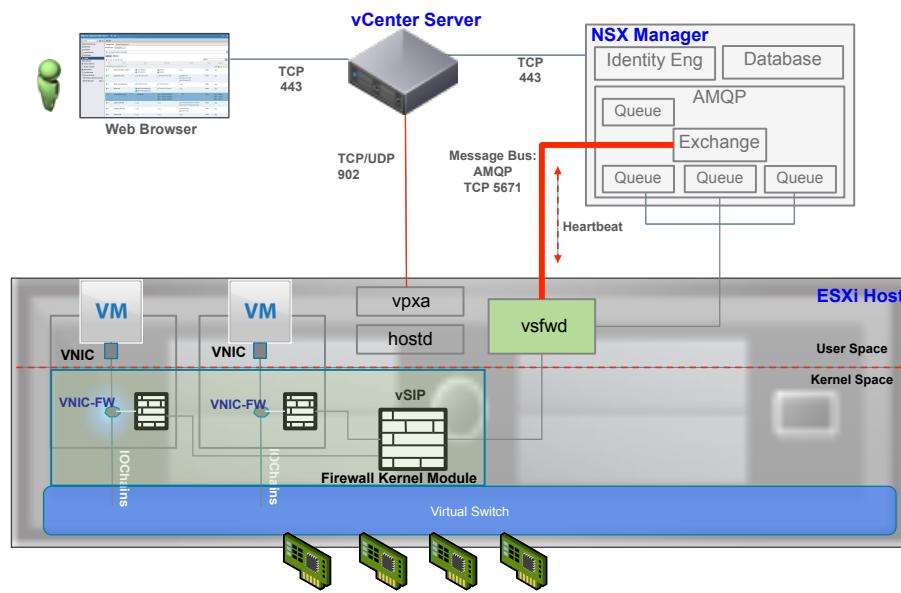
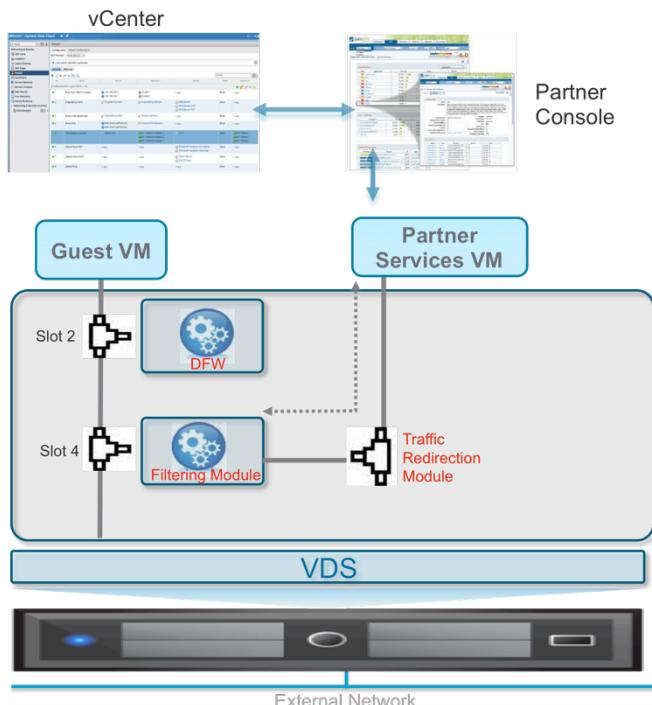


Figure 20 – NSX DFW Components Details

The VSIP kernel module does not simply enhance than DFW functionality. This service insertion platform adds complementary services like SpoofGuard and traffic redirection from third party partners including Palo Alto Networks, CheckPoint, Fortinet, Intel Security, Symantec, RAPID7, and Tend Micro).

SpoofGuard protects against IP spoofing by maintaining a reference table of VM name and IP address, populating it with information retrieved from VMtools during VM's initial boot up. SpoofGuard is inactive by default and must be explicitly enabled per Logical Switch or VDS port-group. When a VM IP address change is detected, traffic from/to this VM can be blocked by the DFW until an NSX administrator approves this new IP address.



**Figure 21 – Traffic Redirection to Third-Party Partner Services**

Traffic redirection to third party vendor provides the capability to steer a particular type of traffic to a selected partner services VM. For instance, web traffic from the Internet to a set of Apache or Tomcat servers can be redirected to an L4-L7 deep packet inspection firewall for advanced protection.

Traffic redirection is defined under Service Composer/Security Policy for NSX version 6.0 or under Partner Security Services tab of the DFW menu in NSX version 6.1.

The Partner Security Services tab provides a powerful and easy to use user interface to define what type of traffic needs to be redirected to which partner services. It follows the same policy definition construct as DFW, providing the same options for source field, destination field and services field. The only difference is in the action field; instead of Block/Allow/Reject, a user can select between redirect/no redirect followed by a partner list. Any partner that has been registered with NSX can be successfully deployed on the platform. Additionally, log options can be enabled for this traffic redirection rule.

The DFW instance on an ESXi host contains 2 separate tables. The rule table is used to store all policy rules, while the connection tracker table caches flow entries for rules with permit actions. One DFW instance is permitted per VM vNIC.

A specific flow is identified by the 5-tuple information consisting source IP address, destination IP address, protocols, L4 source port, and L4 destination port fields. By default, DFW does not perform a lookup on L4 source port, but it can be configured to do so by defining a specific policy rule.

DFW rules are enforced as follows:

- Rules are processed in top-to-bottom ordering.
- Each packet is checked against the top rule in the rule table before moving down the subsequent rules in the table.
- The first rule in the table that matches the traffic parameters is enforced. No subsequent rules can be enforced as the search is then terminated for that packet.

Because of this behavior, it is always recommended to put the most granular policies at the top of the rule table. This will ensure they will be enforced before more specific rules.

The DFW default policy rule, located at the bottom of the rule table, is a “catch-all” rule; packets not matching any other rules will be enforced by the default rule. After the host preparation operation, the DFW default rule is set to ‘allow’ action. This ensures that VM-to-VM communication is not broken during staging or migration phases. It is a best practice to then change this default rule to ‘block’ action and enforce access control through a positive control model (i.e., only traffic defined in the firewall policy is allowed onto the network).

Figure 22 steps through policy rule lookup and packet flow:

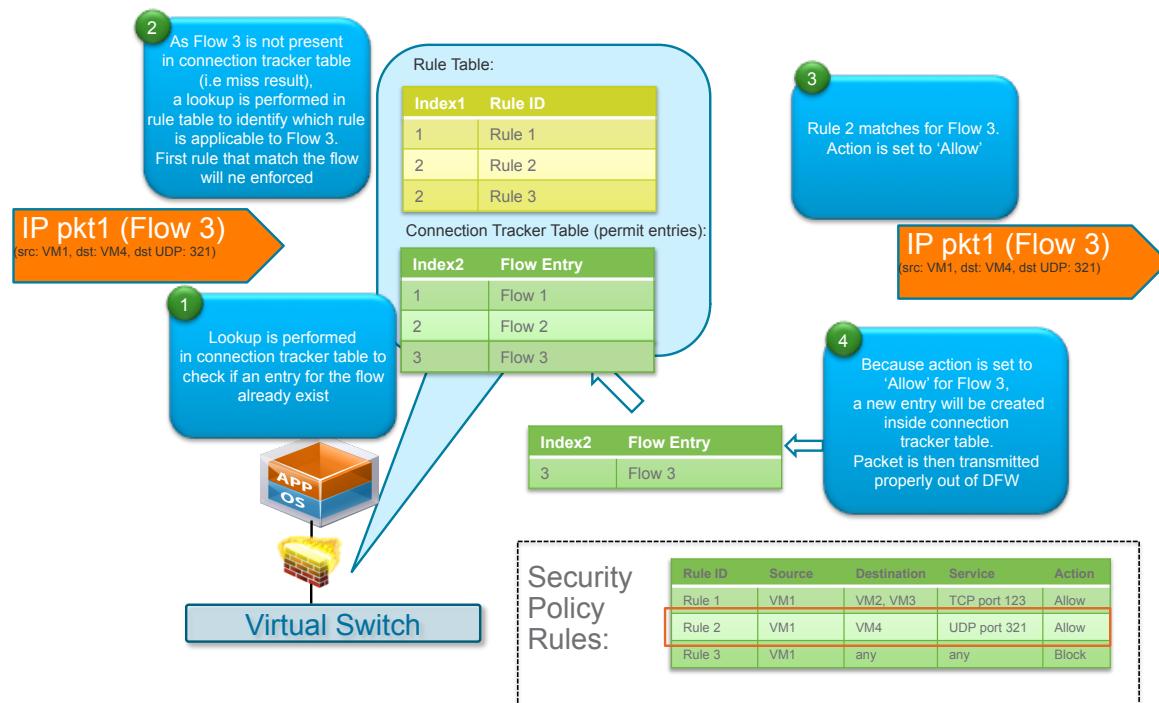


Figure 22 – DFW Policy Rule Lookup and Packet – First Packet

An IP packet identified as pkt1 that matches rule number 2. The order of operation is the following:

1. A lookup is performed in the connection tracker table to determine if an entry for the flow already exists.
2. As flow 3 is not present in the connection tracker table, a lookup is performed in the rule table to identify which rule is applicable to flow 3. The first rule that matches the flow will be enforced.
3. Rule 2 matches for flow 3. The action is set to 'Allow'.
4. Because the action is set to 'Allow' for flow 3, a new entry will be created inside the connection tracker table. The packet is then transmitted out of DFW.

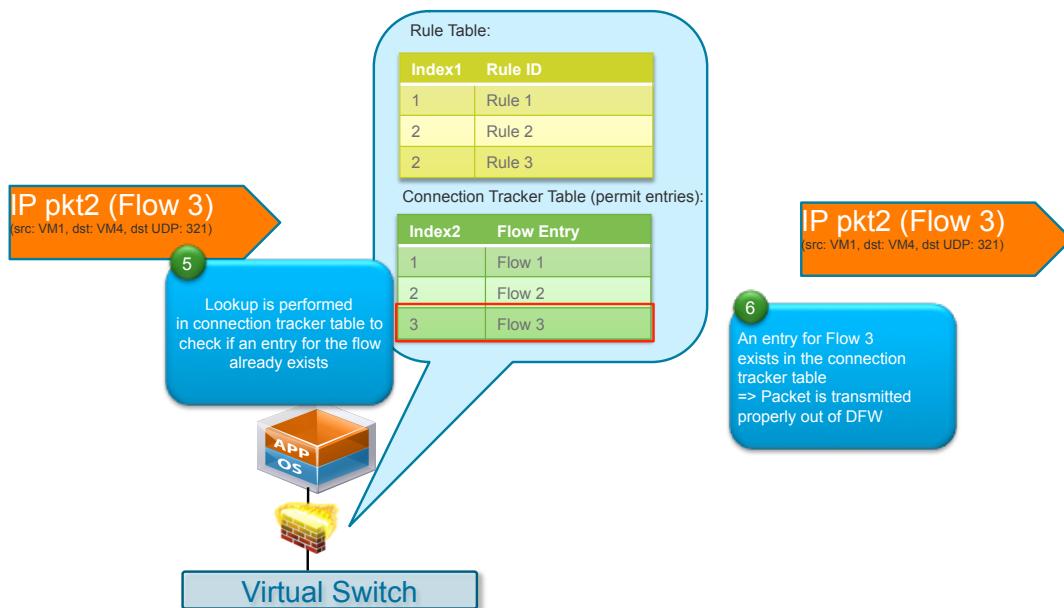


Figure 23 – DFW Policy Rule Lookup and Packet – Subsequent Packets.

Subsequent packets are processed in this order:

5. A lookup is performed in the connection tracker table to check if an entry for the flow already exists.
6. An entry for flow 3 exists in the connection tracker table. The packet is transmitted out of DFW

As DFW fully supports vMotion – either automatic vMotion with DRS or manual vMotion - the rule table and the connection tracker table always follow the VM during vMotion operation. This ensures that there is no traffic disruption during workload moves while connections initiated before vMotion remain intact after the vMotion is completed. DFW brings VM movement freedom while ensuring continuous network traffic protection. This functionality is not dependent on the availability of controllers or NSX manager.

NSX DFW offers a paradigm shift that was not previously possible. Security services are no longer dependent on the network topology as DFW security enforcement is completely decoupled from logical network topology.

In order to provide security services to a server or set of servers in legacy environments, traffic to and from these servers must be redirected to a firewall using VLAN stitching or L3 routing operations. Flows must go through this dedicated firewall in order to protect network traffic.

With NSX DFW, this legacy constraint is removed as the firewall function is brought directly to the VM. Any traffic sent or received by this VM is systematically processed by the DFW. As a result, traffic protection between VMs (i.e., workload to workload) can be enforced if VMs are located on same Logical Switch, VDS VLAN-backed port-group, or even on different Logical Switches.

This key concept is the foundation for the micro-segmentation use cases described in the “[Micro-Segmentation with NSX DFW and Implementation](#)” section.

## 4 NSX Functional Services

In the context of the NSX architecture, logical networking functionality (e.g., switching, routing, security) can be viewed as having a packet forwarding pipeline that implements multiple features, including L2/L3 forwarding, security filtering, and packet queuing.

Unlike a physical switch or router, NSX does not rely on a single device to implement the entire forwarding pipeline. Instead, the NSX controller creates packet processing rules on all relevant ESXi hosts to mimic how a single logical device would handle data. Those hypervisors can be thought of as line cards that implement a portion of the logical network function pipeline. The physical network connecting the ESXi hosts acts as the backplane that carries packets from one line card to the other.

### 4.1 Multi-Tier Application Deployment Example

The network connectivity and services provided through deployment of the NSX components are central to enabling the agile and flexible creation of applications.

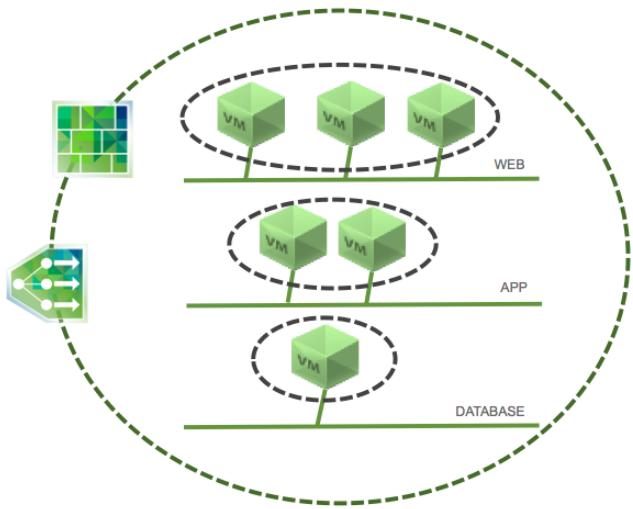


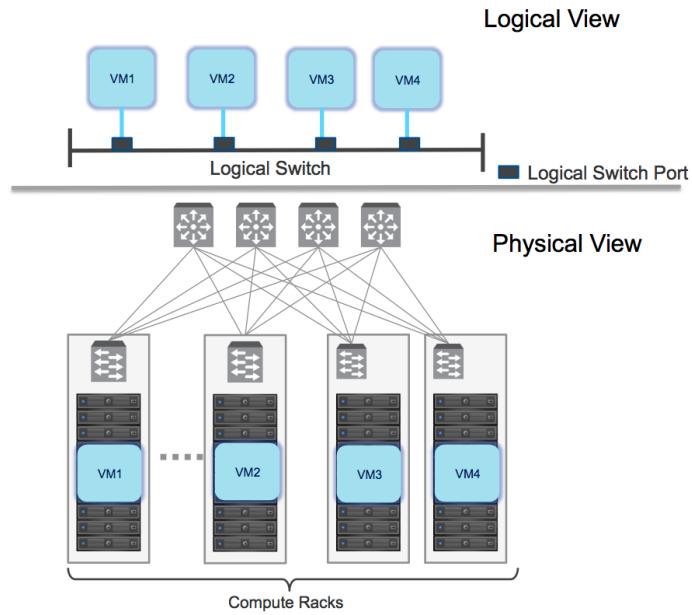
Figure 24 - Deployment of a Multi-Tier Application

A typical multi-tier application architecture is presented in Figure 24. The following sections will discuss the functionality – logical switching, routing and other services – required for the dynamic creation such an application.

### 4.2 Logical Switching

The logical switching capability in the NSX platform provides the ability to spin up isolated logical L2 networks with the same flexibility and agility that exists virtual machines. Endpoints, both virtual and physical, can connect to logical segments and establish connectivity independently from their physical location in the data center network. This enabled through the decoupling of network infrastructure

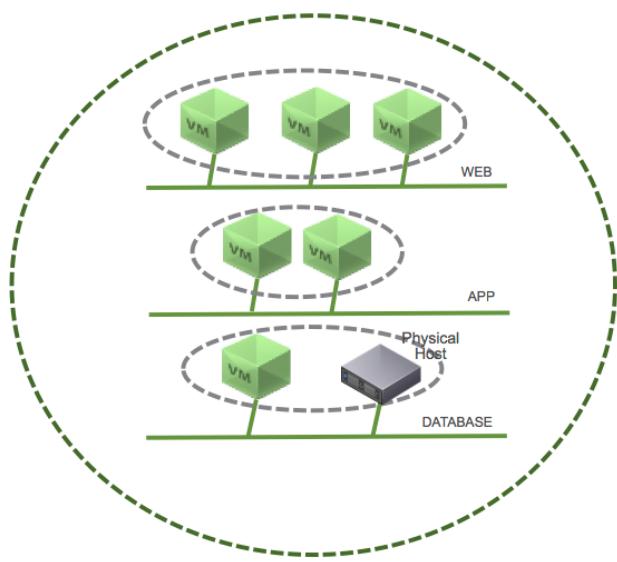
from logical network (i.e., underlay network from overlay network) provided by NSX network virtualization.



**Figure 25 - Logical Switching (Logical and Physical Network Views)**

Figure 25 presents logical and physical network views of a logical switching deployment. In this picture, use of VXLAN overlay technology allows for stretching of an L2 domain across multiple server racks through the use of a Logical Switch. This extension is independent from the specific underlay inter-rack connectivity (e.g., L2 or L3).

When utilized in conjunction with the multi-tier application previously discussed, logical switching allows creation of distinct L2 segments mapped to the different workload tiers, supporting both virtual machines and physical hosts.



**Figure 26 - Creation of Logical Switches for the App Tiers**

Logical switching functionality must enable both virtual-to-virtual and virtual-to-physical communication in each segment. The use of NSX VXLAN-to-VLAN bridging functionality is enabled when logical to physical address space cannot be separated. Use case for bridging includes migration of workloads with the inability of embedded application dependencies to change their IP addresses.

Logical Switching is defined by a segment ID (VXLAN ID) and is unique per NSX manager. Starting with the NSX 6.2 release, segment ID range planning is required in order to enable cross-VC connectivity. It is recommended to keep the segment ID unique for each NSX domain to leverage cross-VC capabilities in NSX 6.2 release; this will help avoid the requirement of renumbering of segment IDs.

#### 4.2.1 Replication Modes for Multi-Destination Traffic

When two VMs connected to different ESXi hosts need to communicate directly, unicast VXLAN-encapsulated traffic is exchanged between the VTEP IP addresses associated with their associated hypervisors. Traffic originated by a VM may also need to be sent to all the other VMs belonging to the same logical switch. Specific instances of this type of L2 traffic include broadcast, unknown unicast, and multicast. These multi-destination traffic types are collectively referred to using the acronym BUM (Broadcast, Unknown Unicast, Multicast).

In each of these scenarios, traffic originated by a given ESXi host must be replicated to multiple remote. NSX supports three different replication modes to enable multi-destination communication on VXLAN backed Logical Switches: multicast, unicast and hybrid. By default, a Logical Switch inherits its replication mode from the Transport Zone, though this behavior can be overridden at the Logical Switch level.

Understanding of the concept of the “VTEP segment” is important for discussion of replication modes.

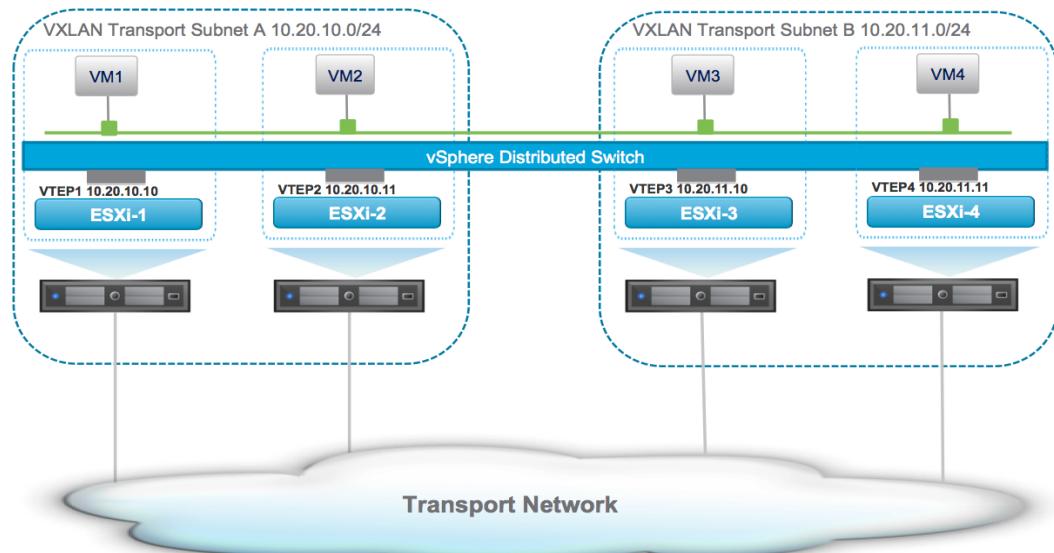


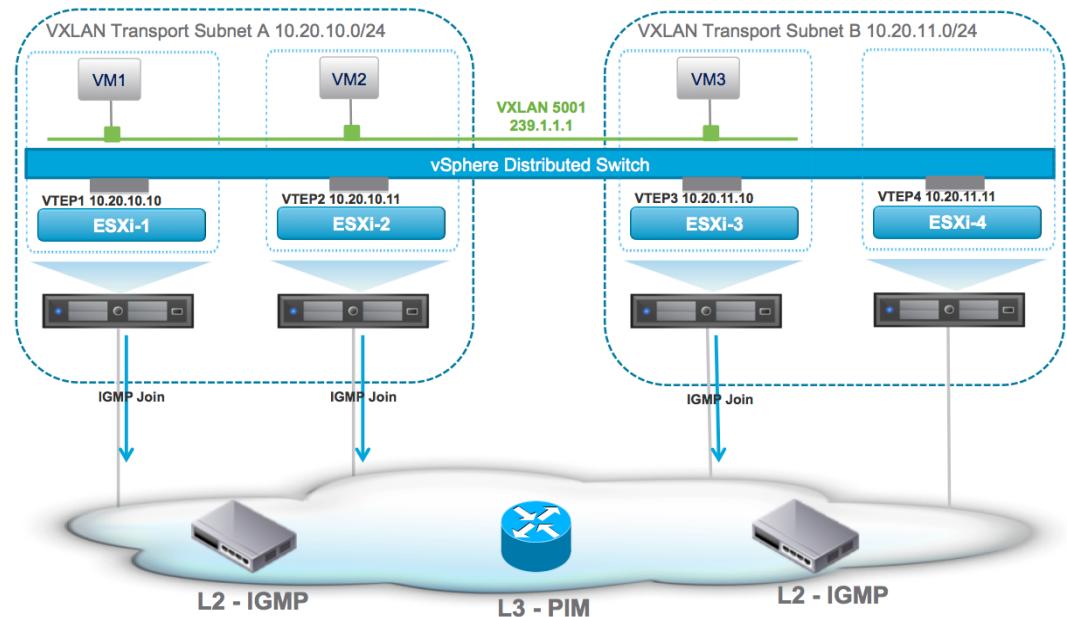
Figure 27 - VTEP Segments

Figure 27 shows four ESXi hosts belonging to two separate VTEP segments. This assumes layer 3 topology. The VTEP interfaces for ESXi-1 and ESXi2 are part of the same transport subnet A (10.20.10.0/24), whereas the VTEPs for ESXi3 and ESXi4 are defined in a separate transport subnet B (10.20.11.0/24). Each VXLAN transport subnet is a distinct VTEP segment connected to a common Logical Switch. Both VTEP segments are connected single transport zone VLAN.

#### 4.2.1.1 Multicast Mode

When Multicast replication mode is chosen for a given Logical Switch, NSX relies on the native L2/L3 multicast capability of the physical network to ensure VXLAN encapsulated multi-destination traffic is sent to all VTEPs. Multicast mode is the process for handling BUM traffic specified by the VXLAN IETF draft, and does not leverage any of the enhancements brought by NSX with the introduction of the controller clusters. This behavior does not leverage the decoupling of logical and physical networking as communication in the logical space is predicated on the multicast configuration required in the physical network infrastructure.

In this mode, a multicast IP address must be associated to each defined VXLAN L2 segment (i.e., Logical Switch). L2 multicast capability is used to replicate traffic to all VTEPs in the local segment (i.e., VTEP IP addresses that are part of the same IP subnet). Additionally, IGMP snooping should be configured on the physical switches to optimize the delivery of L2 multicast traffic. To ensure multicast traffic is also delivered to VTEPs in a different subnet from the source VTEP, the network administrator must configure PIM and enable L3 multicast routing.



**Figure 28 – IGMP Joins**

In the example in Figure 28, the VXLAN segment 5001 is associated with multicast group 239.1.1.1. When the first VM is connected to the logical switch, the ESXi hypervisor hosting the VM generates an IGMP join message to notify the physical infrastructure that it is interested in receiving multicast traffic sent to that specific group.

As a result of the IGMP joins sent by ESXi1-ESXi-3, multicast state is stored in the physical network to ensure delivery of multicast frames sent to the 239.1.1.1 destination. In this example, ESXi-4 does not send the IGMP join since it does not host any active receivers (i.e., VMs connected to the VXLAN 5001 segment).

The sequence of events required to deliver a BUM frame in multicast mode is depicted in Figure 29.

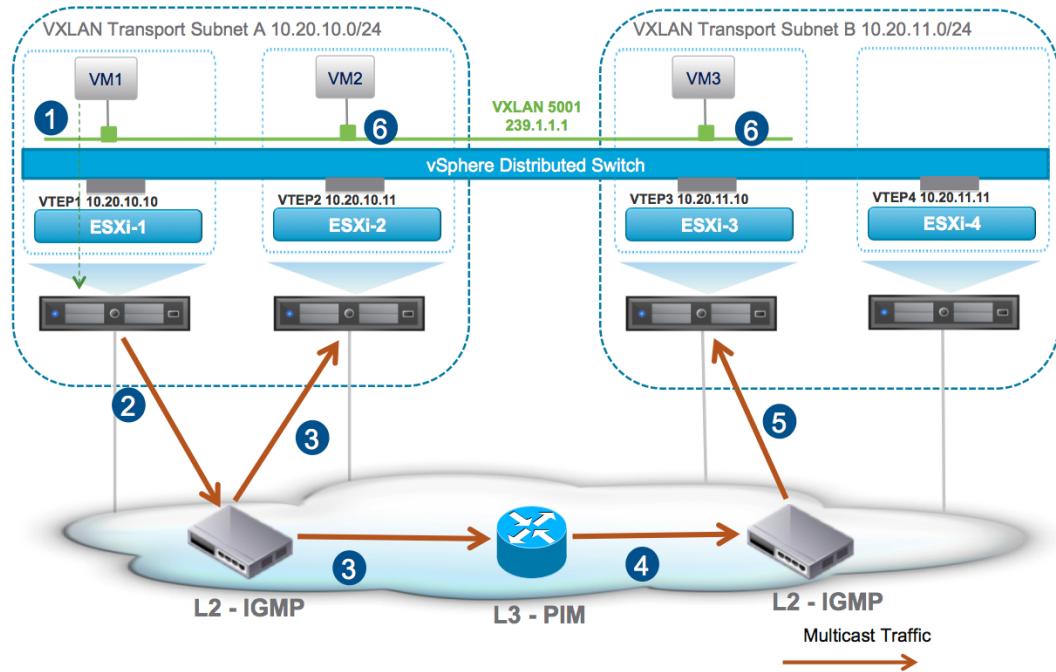


Figure 29 - Replication in Multicast Mode

1. VM1 generates a BUM frame.
2. ESXi-1 VXLAN-encapsulates the original frame. The destination IP address in the outer IP header is set to 239.1.1.1 and the multicast packet is sent into the physical network. In this case, ESXi-1 acts as a source for the multicast stream 239.1.1.1.
3. The L2 switch receiving the multicast frame performs replication. Where IGMP snooping is configured on the switch, it will be able to replicate the frame to the relevant interfaces connecting to ESXi-2 and the L3 router. If IGMP snooping is not enabled or supported, the L2 switch treats the frame as an L2 broadcast packet and replicates it to all interfaces belonging to the same VLAN of the port where the packet was received.
4. The L3 router performs L3 multicast replication and sends the packet into the transport subnet B.

5. The L2 switch behaves similarly to what discussed at step 3 and replicates the frame.
6. ESXi-2 and ESXi-3 decapsulate the received VXLAN packets, exposing the original Ethernet frames that are then delivered to VM2 and VM3.

When configuring multicast mode, consideration must be given on how to perform the mapping between VXLAN segments and multicast groups.

The first option is to perform a 1:1 mapping. This has the advantage of providing a multicast traffic delivery in a very granular fashion; a given ESXi host would receive traffic for a multicast group only if at least one local VM is connected to the corresponding multicast group. Negatively, this option may significantly increase the amount of multicast state required in physical network devices, and understanding the maximum number of groups those platforms can support is critical.

The other choice involves leveraging a single multicast group for all the defined VXLAN segments. This dramatically reduces the volume of multicast state in the transport infrastructure, but may cause unnecessary traffic to be received by the ESXi hosts. Referring back to Figure 29, ESXi-4 would receive traffic belonging to VXLAN 5001 as soon as a local VM gets connected to any logical segment, not simply VXLAN 5001).

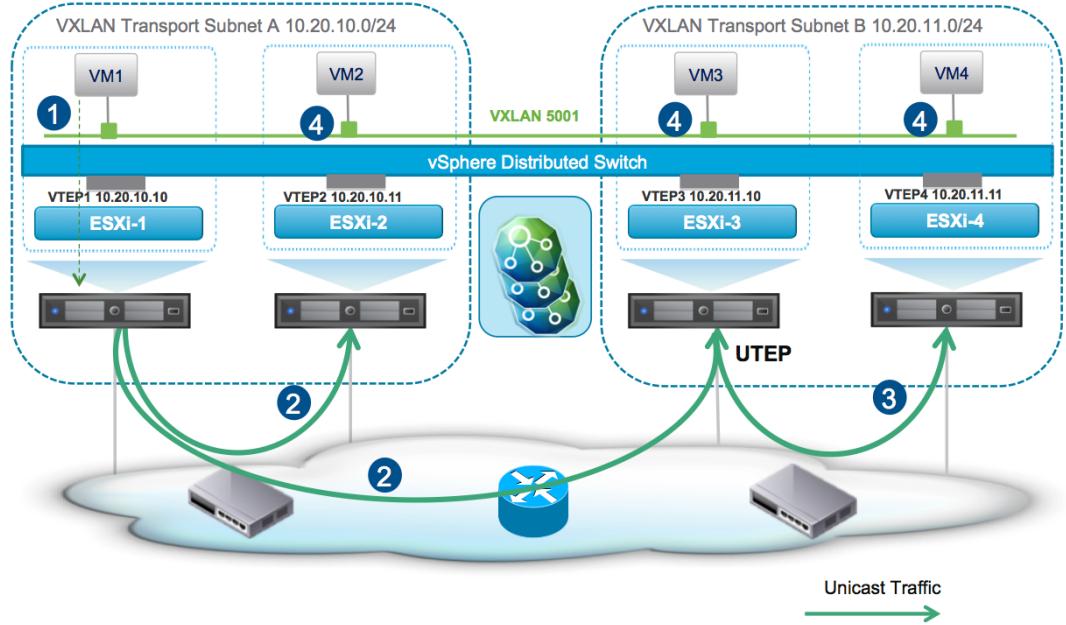
The most common strategy involves the decision to deploy an m:n mapping ratio as a trade-off scenario between these two options. With this configuration, every time a new VXLAN segment is instantiated, up to the maximum specified “m” value; it will be mapped to a multicast group part of the specified range in round robin fashion. In this implementation, VXLAN segment “1” and “n+1” will be using the same group.

Details on multicast mode described here help build the baseline of knowledge required to better understand and appreciate the enhancements made possible by NSX with the other two options (i.e., unicast and hybrid modes).

#### **4.2.1.2 Unicast Mode**

Unicast mode represents the opposite approach from multicast mode, wherein the decoupling of logical and physical networks is fully achieved. In unicast mode, the ESXi hosts in the NSX domain are divided in separate groups (i.e., VTEP segments) based on the IP subnet of VTEP interfaces. An ESXi host in each VTEP segment is selected to play the role of Unicast Tunnel End Point (UTEP). The UTEP is responsible for replicating multi-destination traffic received from ESXi hypervisors hosting VMs in different VTEP segments. It distributes this traffic to all the ESXi hosts within its own segment (i.e., hosts whose VTEPs belong to the same subnet of the UTEP’s VTEP interface).

In order to optimize the replication behavior, every UTEP will only replicate traffic to ESXi hosts on a local segment that have at least one VM actively connected to the logical network where multi-destination traffic is destined. In the same way, traffic will only be sent by the source ESXi to the remote UTEPs if there is at least one active VM connected to an ESXi host in that remote segment.



**Figure 30 – Unicast Mode**

Figure 30 illustrates the unicast mode replication mechanism. This process is defined as follows:

1. VM1 generates a BUM frame to be sent to each VM connected to Logical Switch 5001. In this instance there is no need to specify a multicast group associated to this VXLAN segment.
2. ESXi1 references its local VTEP table. This table is filled with the information received via control plane communication with the controller nodes. The check validates the need to replicate the packet to the other VTEP belonging to the local segment, ESXi2, as well as to the UTEP part of remote segments, ESXi3. The unicast copy sent to the UTEP has a specific bit set in the VXLAN header – the “REPLICATE\_LOCALLY” bit – as an indication to the UTEP that this frame is coming from a remote VTEP segment and may need to be locally replicated.
3. The UTEP receives the frame, references the local VTEP table, and replicates it to all the ESXi hosts which are part of the local VTEP segment with at least one VM connected to VXLAN 5001. In this example, that is simply ESXi-4.

In the above example, if VM2 generated a BUM frame, the unicast replication would be performed by the ESXi-2, following the same steps, though ESXi-2 may elect a different host as remote UTEP. This decision is made locally and independently by each ESXi host to ensure that replication duties, even for traffic belonging to the same logical segment, can be shared by multiple VTEPs.

Unicast mode replication mode does not require explicit configuration on the physical network to enable distribution of multi-destination VXLAN traffic. This mode can be used for small to medium size environments where BUM traffic is not high and NSX is deployed in an L2 topology where all VTEPs are in the same subnet. Unicast mode also scales well with L3 topologies where UTEP boundaries can be clearly identified (e.g., each L3 rack has its own VTEP subnet). The increasing load of BUM replication is managed by spreading the burden among all participating hosts.

#### 4.2.1.3 Hybrid Mode

Hybrid mode offers operational simplicity similar to unicast mode – IP multicast routing configuration is not required in the physical network – while leveraging the L2 multicast capability of physical switches.

This is illustrated in Figure 31, where the specific VTEP responsible for performing replication to the other local VTEPs named “MTEP”. In hybrid mode the [M]TEP uses L2 [M]ulticast to replicate BUM frames locally, while the [U]TEP leverages [U]nicast frames.

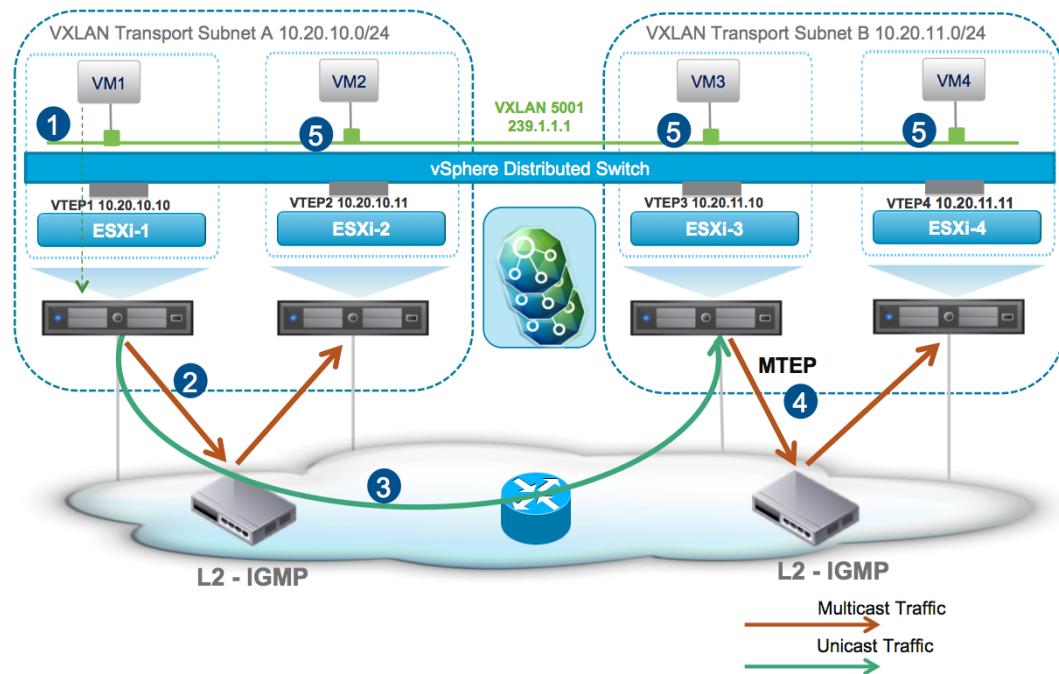


Figure 31 - Hybrid Mode Logical Switch

1. VM1 generates a BUM frame which must be replicated to all VMs that are part of VXLAN 5001. The multicast group 239.1.1.1 must be associated with the VXLAN segment, as multicast encapsulation is performed for local traffic replication.
2. ESXi1 encapsulates the frame in a multicast packet addressed to the 239.1.1.1 group. Layer 2 multicast configuration in the physical network is leveraged to ensure that the VXLAN frame is delivered to all VTEPs in the local VTEP segment. In hybrid mode the ESXi hosts send an IGMP join

when there are local VMs interested in receiving multi-destination traffic, similar to Figure 28). Since PIM is not required, it is strongly recommended to define an IGMP querier per VLAN to ensure successful L2 multicast delivery and avoid non-deterministic behavior. When IGMP snooping is enabled, but there's no IGMP querier definition, some Ethernet switches will resort to flooding the multicast traffic in the VLAN while others will drop it. Please refer to the documentation provided by your vendor of choice.

3. At the same time ESXi-1 looks at the local VTEP table and determines the need to replicate the packet to the MTEP part of remote segments, in this case ESXi3. The unicast copy is sent to the MTEP with the corresponding bit set in the VXLAN header as an indication to the MTEP that this frame is coming from a remote VTEP segment and needs to be locally re-injected in the network.
4. The MTEP creates a multicast packet and sends it to the physical network where will be replicated by the local L2 switching infrastructure.

As with unicast mode, if VM2 were to generate a BUM frame, the ESXi-2 host would perform the unicast replication to remote MTEPs. Each ESXi host locally determine which ESXi hosts belonging to remote VTEP segments are acting as MTEP. ESXi-1 may use ESXi-3 as remote MTEP, whereas ESXi-4 may be used as remote MTEP by ESXi-2.

Hybrid mode allows deployment of NSX in large L2 topologies by helping scale multicast at layer 2 with the simplicity of unicast. It also allows scaling in L3 leaf-spine topologies without requiring PIM to forward multicast (BUM frame) beyond layer 3 ToR boundaries while still allowing multicast replication in physical switch for layer 2 BUM replication. In summation, hybrid mode addresses the requirement for BUM replication in large-scale design regardless of underlay topology.

#### 4.2.2 Populating the Controller Tables

Controller tables handle information essential for L2 unicast communication. Control plane communication between ESXi hosts and the controller cluster is used to populate the VTEP, MAC, and ARP tables on controller nodes. This process is detailed in Figure 32.

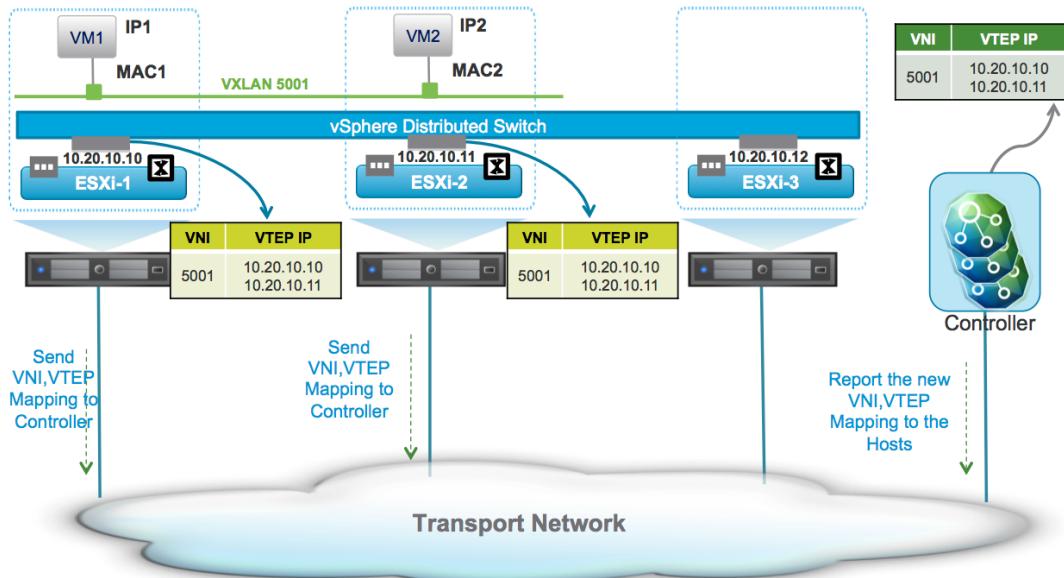


Figure 32 - VNI-VTEP Report

When the first VM connects to a VXLAN segment – VM1 on ESXi1 and VM2 on ESXi-2 in this example – the ESXi host generates a control plane message to the specific controller node in charge of that specific logical switch slice with VNI/VTEP mapping information. The controller node populates its local VTEP table with this information and sends a report message to all ESXi hypervisors hosting VMs actively connected to that same VXLAN segment. In this example, no message is sent to ESXi-3 as there is no active VM connected to the segment. The ESXi hosts can then populate their local VTEP tables, and this information can be leveraged to determine the list of VTEPs for multi-destination traffic replication.

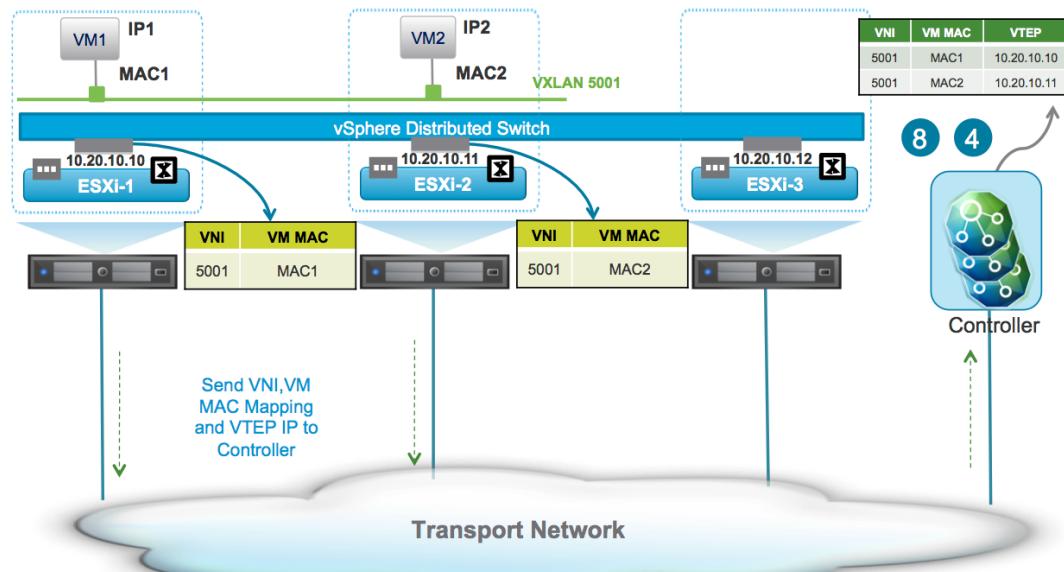
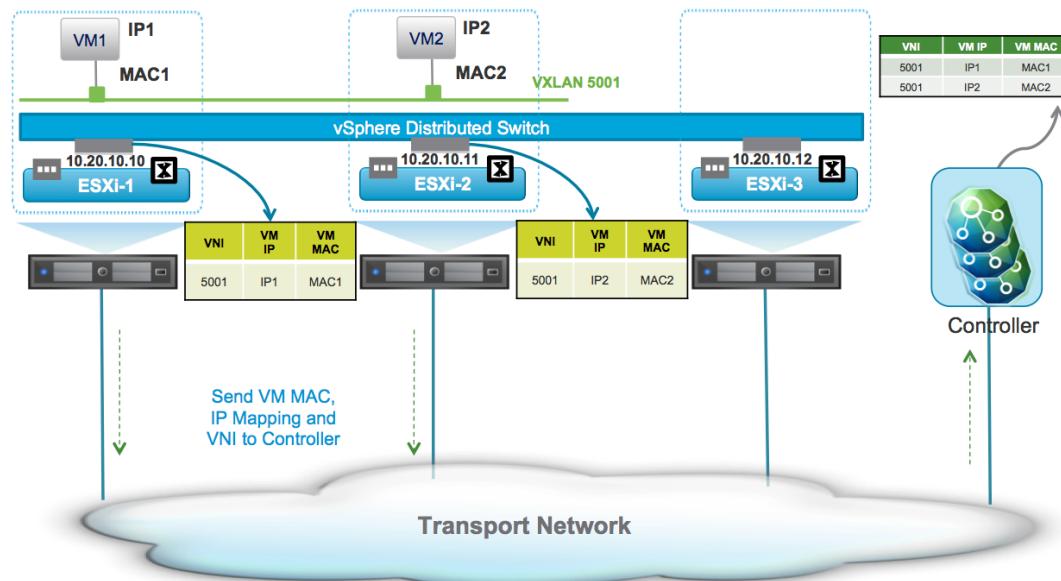


Figure 33 – VNI-MAC Address Report

ESXi hosts also report the MAC address for VMs locally connected to a specific VNI. The controller uses those reports to populate its local MAC table, but unlike the VNI-VTEP report, it does not send this information to all the ESXi hosts. Because of this, ESXi hosts only aware of the locally connected MAC addresses, as highlighted in Figure 33.

The final piece of information shared with the controller is the IP address of the VMs. This controller populates its local ARP table in order to perform the ARP suppression functionality discussed the “Unicast Traffic (Virtual to Virtual Communication)” section.



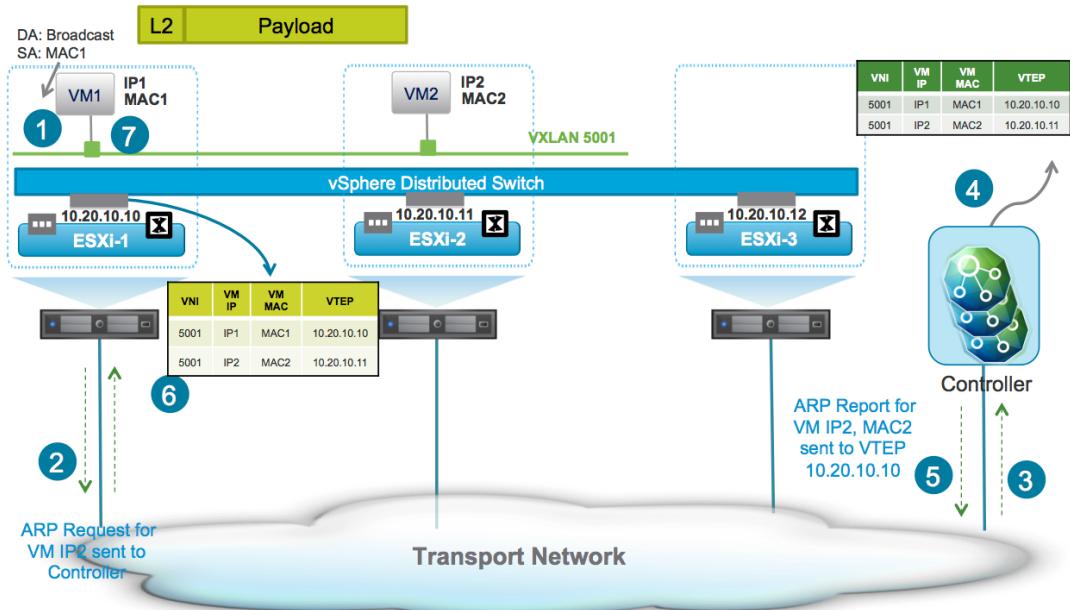
**Figure 34 - VNI-IP Address Report**

The ESXi hosts learn the IP address of locally connected VMs in two ways. For VMs obtaining an IP address using DHCP, the ESXi host will snoop the DHCP response sent by the DHCP server. For VMs that are statically addressed, ARP requests originated by the VMs will be used to learn their IP addresses. Once the IP address of the machine is learned, the ESXi hosts send the MAC/IP/VNI information to the controller to populate its local ARP table.

#### 4.2.3 Unicast Traffic (Virtual to Virtual Communication)

Using the information from its tables, the NSX controller can perform an “ARP suppression” that avoids the need to flood ARP traffic in the L2 domain (i.e., VXLAN segment) where the virtual machines are connected. ARP requests represent the vast majority of L2 broadcast traffic found in the network, so removing these provides significant benefits to the stability and scalability of the overall network infrastructure.

Figure 35 shows how ARP resolution is performed leveraging the control plane with the NSX controller when there is a need to establish unicast communication between virtual machines belonging to the same logical switch (i.e., VXLAN segment).

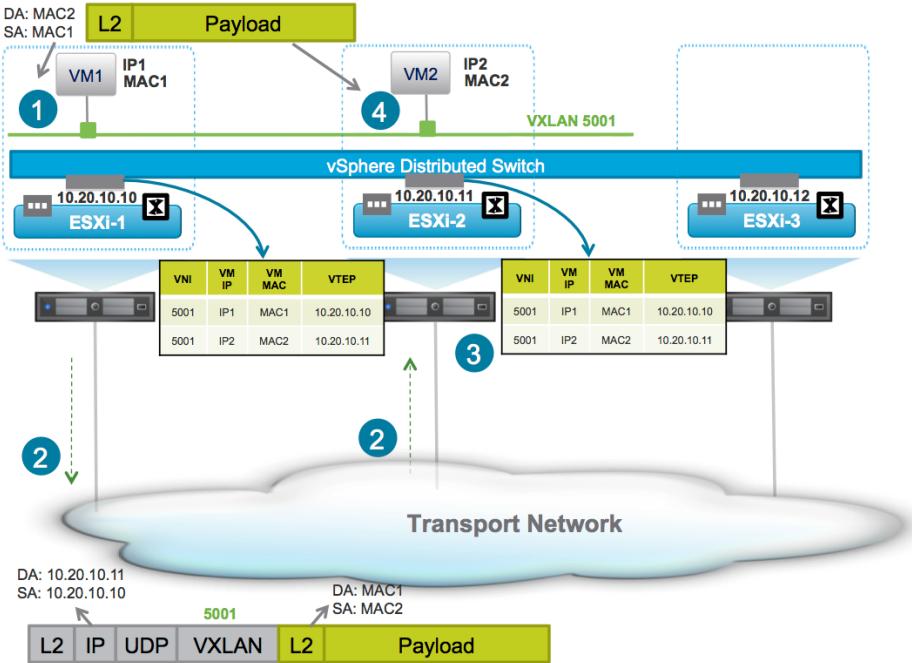


**Figure 35 – ARP Resolution via NSX Controller**

1. VM1 generates an ARP request (e.g., L2 broadcast packet) to determine the MAC/IP mapping information for VM2.
2. ESXi-1 intercepts the request and generates a control plane request to the controller asking for the MAC/IP mapping information.
3. The controller receives the control plane request.
4. The controller checks its local ARP table for the required mapping information.
5. The mapping information is sent to ESXi-1 with a control plane ARP report message.
6. ESXi-1 receives the control plane message and updates its local table with the mapping information. At this point the VTEP where VM2 is connected is known (10.20.10.11).
7. ESXi-1 generates an ARP response on behalf of VM2 – the source MAC address in the frame is MAC2 – and delivers it to VM1. This proxy process is invisible to VM1; it simply views the reply as having come directly from VM2.

If the controller does not have the mapping information, it will notify ESXi-1. This will flood the ARP frame in the VXLAN 5001 segment for discovery of VM2. How this flooding of the ARP request is performed depends on the configured logical switch replication mode, as described in the “Replication Modes for Multi-Destination Traffic” section.

Once VM1 populates its ARP cache, it will be able to send data traffic to VM2, as highlighted in Figure 36.



**Figure 36 - Intra Logical Switch Unicast Communication**

1. VM1 generates a data packet directed to VM2.
2. ESXi-1 learned about the location of VM2 from the ARP report received from the controller through control plane learning. It encapsulates the packet originated by VM1 in a VXLAN packet destined to the VTEP of ESXi2 – 10.20.10.11.
3. ESXi-2 receives the packet and leverages the information in the external IP header to learn about the location of VM1, associating VM1 MAC and IP addresses to the VTEP of ESXi-1.
4. The frame is delivered to VM2.

As both ESXi-1 and ESXi-2 have populated their local tables, traffic can flow in each direction.

#### 4.2.4 Unicast Traffic (Virtual to Physical Communication)

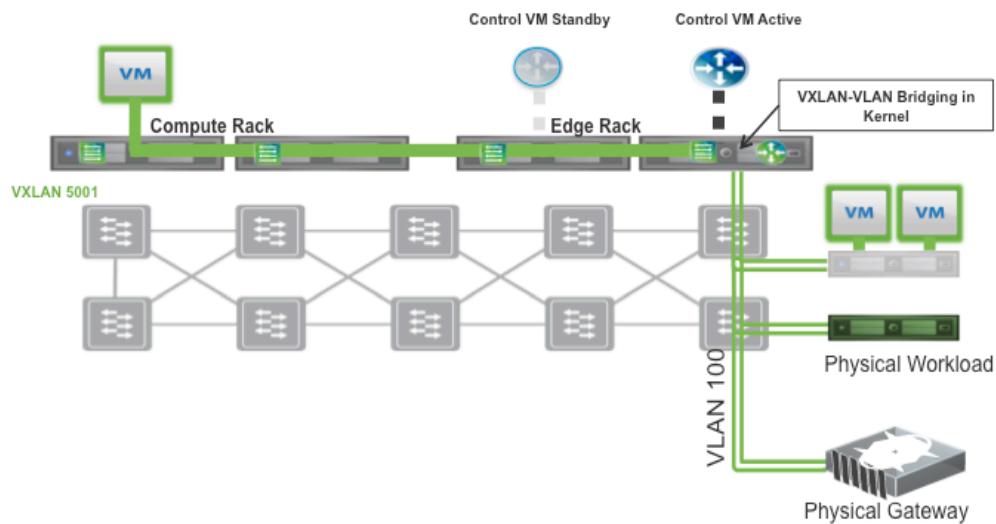
Circumstances exist where it may be required to establish L2 communication between virtual and physical workloads. Typical scenarios include:

- **Deployment of Multi-Tier Applications:** Web, application, and database tiers can be deployed as part of the same IP subnet. Web and application tiers are typically leverage virtual workloads, while the database tier commonly deploys bare-metal servers. As a consequence, it may be required to establish intra-subnet/intra-L2 domain communication between the application and the database tiers.
- **Physical to Virtual (P-to-V) Migration:** During an ongoing migration project, virtualization of applications previously running on bare metal servers is required to support the mix of virtual and physical nodes on the same IP subnet.

- **Leveraging External Physical Devices as Default Gateway:** A physical network device may be deployed to function as a default gateway for the virtual workloads connected to a logical switch. In this case an L2 gateway function is required to establish connectivity to that gateway.
- **Deployment of physical appliances:** These devices could include common physical appliances that have not yet or will not be virtualized (e.g., firewalls, load balancers, etc.).

It is possible to deploy devices performing a bridging functionality that enables communication between the virtual world (e.g., logical switches) and the physical world (e.g., non-virtualized workloads and network devices connected to traditional VLANs).

NSX offers this functionality in software through the deployment of NSX L2 bridging, allowing VMs to be connected at layer 2 to a physical network through VXLAN to VLAN ID mapping. This is supported even where the hypervisor running the VM is not physically connected to that L2 physical network.



**Figure 37 - NSX L2 Bridging**

Figure 37 shows an example of L2 bridging. A VM connected in logical space to the VXLAN segment 5001 needs to communicate with a physical device deployed in the same IP subnet but connected to a physical network infrastructure through VLAN 100. The VXLAN-VLAN bridging configuration is part of the distributed router configuration; it is running on the same ESXi host where the control VM is located. This control VM serves as an anchor for the location of the bridging instance. In case of failure of that ESXi host, the ESXi hosting the standby control VM will be activated and restart the bridging instance. The data plane for the bridging is entirely run in kernel space, so the control VM is not in the data path.

For more information on distributed routing and the role of the control VMs, please refer to the following “Logical Routing” section.

Additional deployment considerations for the NSX L2 bridging include:

- The VXLAN-VLAN mapping is always performed in a 1:1 fashion. Traffic for a given VXLAN can only be bridged to a specific VLAN, and vice versa.
- A given bridge instance for a specific VXLAN-VLAN pair is always active on a single ESXi host.
- Through configuration it is possible to create multiple bridges instances for different VXLAN-VLAN pairs and ensure they are spread across separate ESXi hosts. This improves the overall scalability of the L2 bridging function.
- The NSX layer 2 bridging data path is entirely performed in the ESXi kernel rather than in user space. The control VM is only used to determine on which ESXi host where a given bridging instance is active; not to perform the actual bridging function.
- Similar to Edge VM, the control VM runs in active-standby mode. In case of the active control VM failure, the convergence of bridged traffic is governed by hear-beat timer (15 seconds by default) between active-standby. Typically bridging functionality is temporary for migration of workload thus timer tuning is not necessary, however if required, the minimum timers value should not be set below six seconds.

Figure 38 and Figure 39 show the ARP exchange between a virtual machine and a bare-metal server. This is the first step required in providing virtual-to-physical unicast communication.

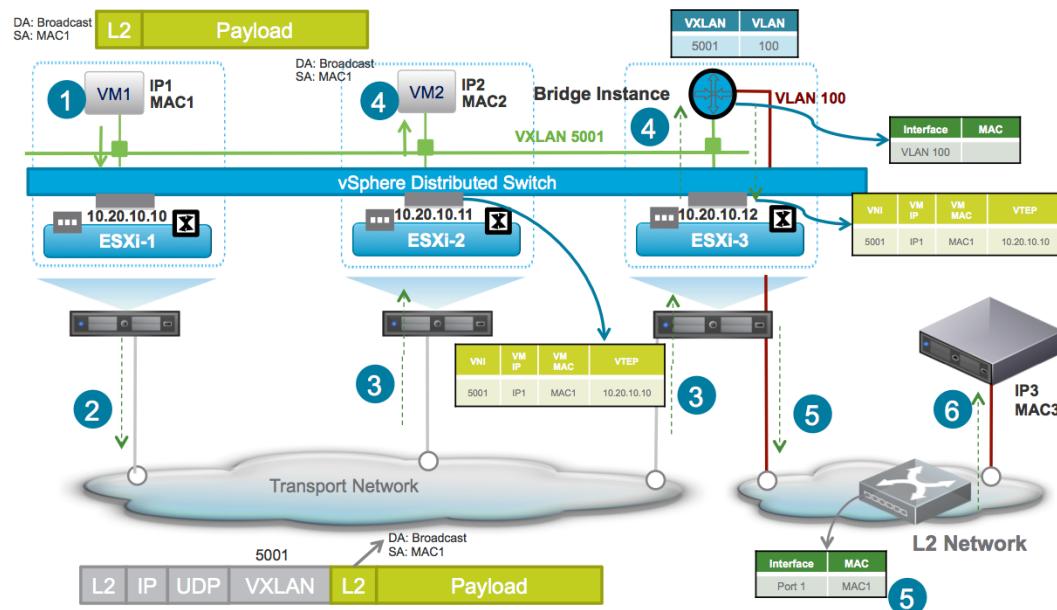


Figure 38 - ARP Request Originated from the Virtual Machine

- VM1 generates an ARP request to retrieve the MAC/IP mapping for the bare-metal server.
- The ESXi hosts intercepts the ARP request and generates a control-plane request directed to the controller to retrieve the mapping information. In

this scenario that the controller does not have this information because the bare-metal server has just been connected to the network and it has not yet generated traffic yet. ESXi-1 must then flood the ARP request in the VXLAN 5001 segment, leveraging one of the methods discussed in the “Replication Modes for Multi-Destination Traffic” section.

3. The ARP request is sent to ESXi-2 and ESXi-3 since both have workloads actively connected to VXLAN 5001; VM2 on ESXi-2 and the bridging instance on ESXi-3. The ESXi hosts learn VM1 location from the reception of this packet.
4. VM2 receives and discard the request. The bridge instance forwards the L2 broadcast packet into the physical L2 network.
5. The frame is flooded in VLAN 100 and all the physical L2 switches perform regular data-plane learning for VM1 MAC address MAC1.
6. The ARP request reaches the bare-metal server.

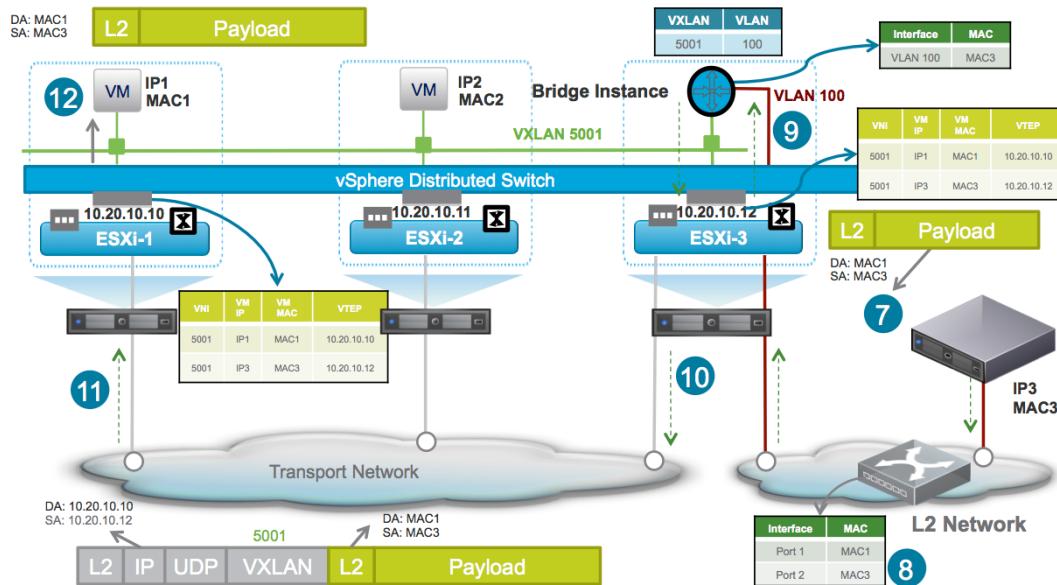


Figure 39 - ARP Response Originated from the Bare-Metal Server

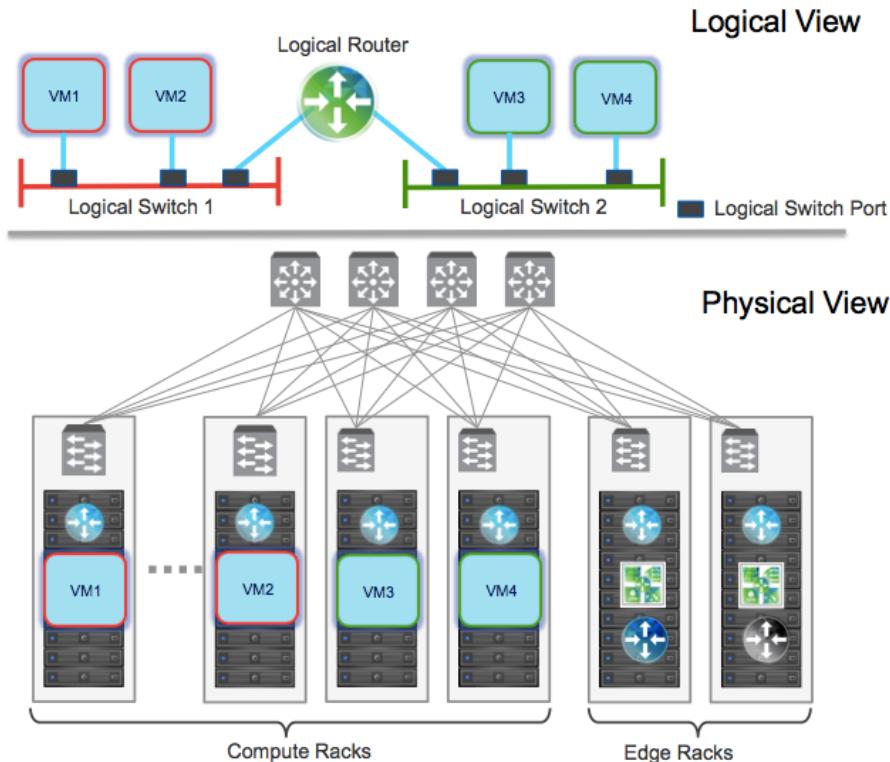
7. The bare-metal server generates the unicast ARP reply destined to VM1.
8. The ARP reply is switched in the physical network and the L2 switches perform data-plane learning for the MAC address of the bare-metal server MAC3.
9. The active NSX L2 Bridge on ESXi-3 receives the ARP reply and learns MAC3.
10. ESXi-3 knows where the MAC1 is located from the learning performed at previous step 3. It encapsulates the frame and send it to the VTEP of ESXi-1, 10.20.10.10.
11. ESXi-1 receives the frame, decapsulates it, and adds to its local table the information that MAC3 is associated with the VTEP of ESXi3. This is the host where the active bridge instance for VXLAN 5001 is running.
12. ESXi-1 generates an ARP response on behalf of the bare-metal server and delivers it to VM1. The source MAC address in this frame is MAC3.

Data plane communication between VM1 and the bare-metal host can now commence in similar fashion to what discussed for unicast virtual to virtual L2 communication.

### 4.3 Logical Routing

The logical routing capability in the NSX platform provides the ability to interconnect both virtual and physical endpoints deployed in different logical L2 networks. This is possible due to the decoupling of network infrastructure from logical networks provided by the deployment of network virtualization.

Figure 40 shows both the logical and corresponding physical view of a routed topology interconnecting two logical switches.



**Figure 40 - Logical Routing (Logical and Physical Network Views)**

The deployment of logical routing can serve two purposes; interconnecting endpoints – logical or physical – belonging to separate logical L2 domains or interconnecting endpoints belonging to logical L2 domains with devices deployed in the external L3 physical infrastructure. The first type of communication, often confined inside a data center, is referred to as east-west communication. The second type is north-south communication, which provides connectivity into the data center from the external physical world (e.g., WAN, Internet, or intranet).

In the multi-tier application deployment example, logical routing is both the functionality required to interconnect between the different application tiers (e.g., east-west communication) as well as mechanism that provides access to the web tier from the external L3 domain (north-south communication). These relationships are shown in Figure 41.

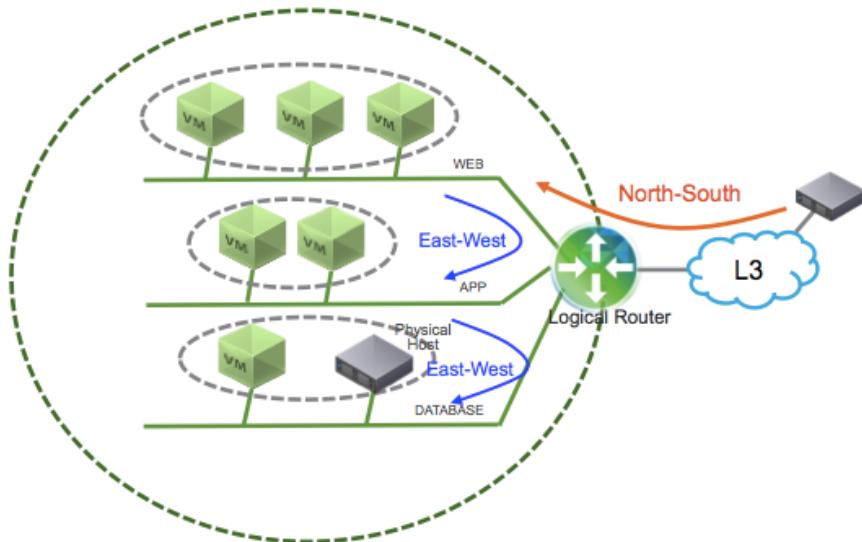


Figure 41 - Logical Routing for a Multi-Tier Application

#### 4.3.1 Logical Routing Components

These two types of logical routing usually achieved leveraging two different functionalities – centralized routing and distributed routing.

Centralized routing represents the on-ramp/off-ramp functionality allowing communication between the logical network space and the external L3 physical infrastructure.

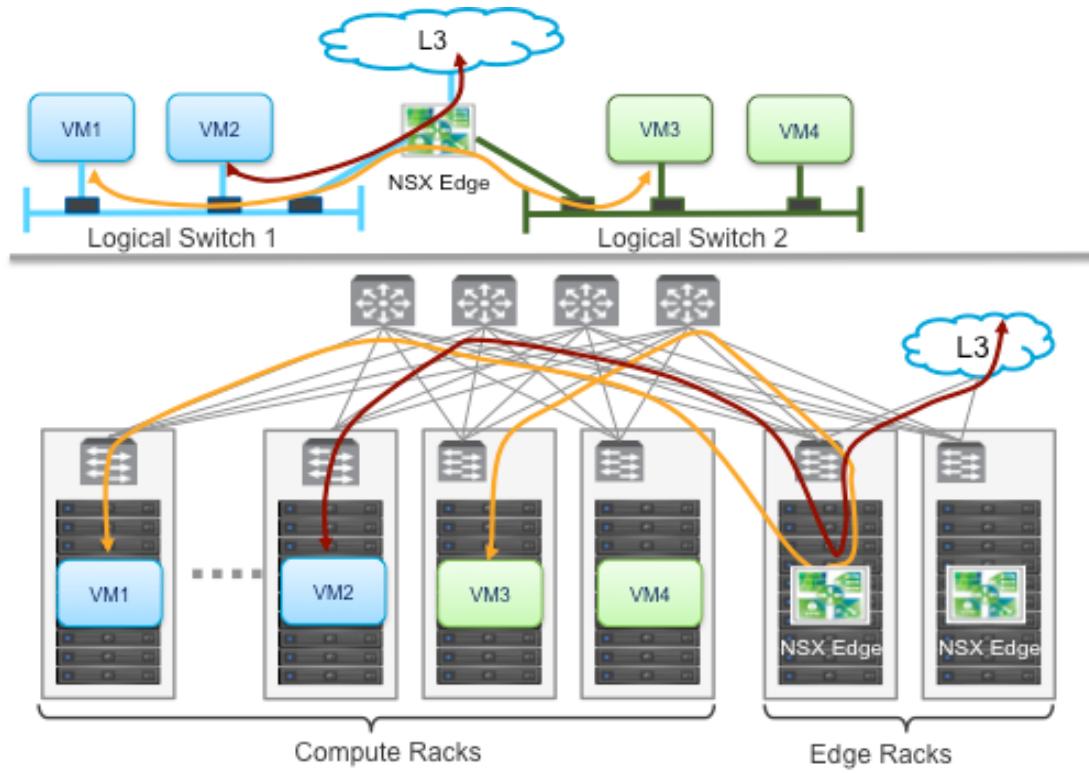
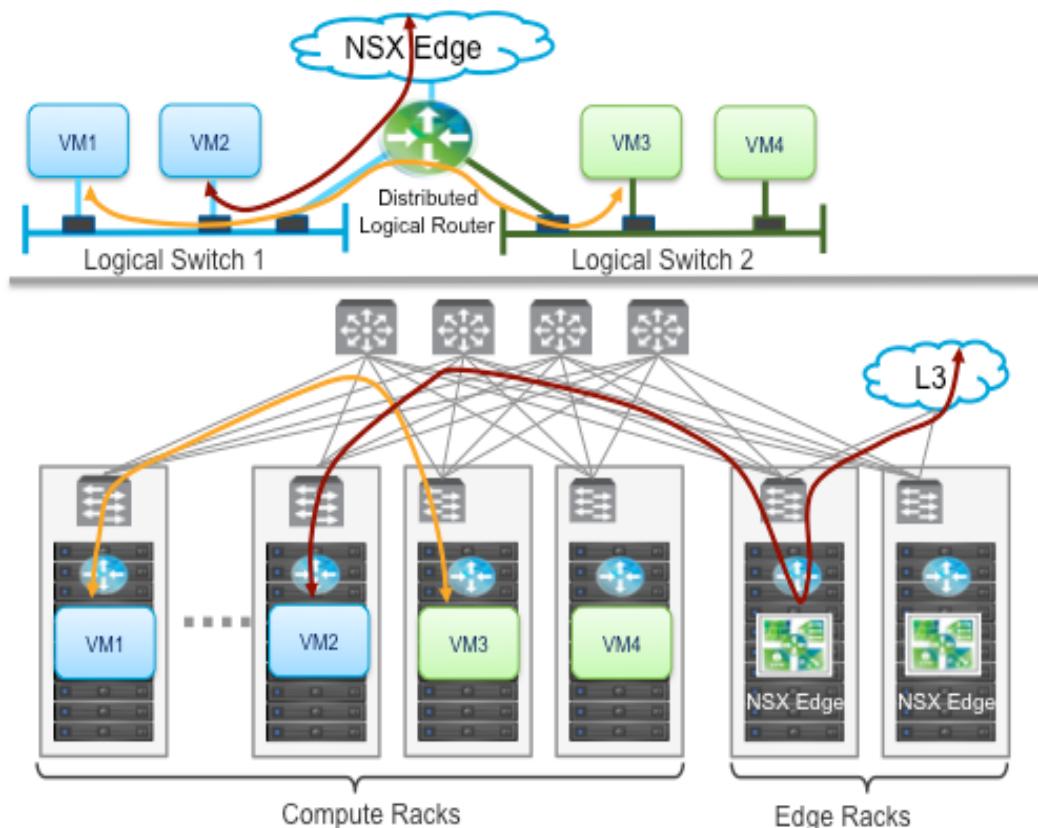


Figure 42 - Centralized Routing

Figure 42 highlights how the NSX Edge services gateway provides traditional centralized routing support in the NSX platform. In addition to routing services, NSX Edge also supports other network services including DHCP, NAT, firewall, load balancing, and VPN.

While centralized routing deployment can be utilized both for east-west and north-south routed communications, east-west routed flows are not optimized in a centralized routing deployment since traffic must be hair-pinned from the compute racks toward the edge rack where the active NSX Edge is deployed. This applies even when two virtual machines belonging to separate Logical Switches are deployed inside the same hypervisor. Figure 42 shows a packet route in which the yellow flow runs between two VMs in distinct subnets and goes through NSX Edge VM. The red flow is from the VM to an external network which always goes through an Edge VM.

As highlighted in Figure 43, the deployment of distributed routing prevents this hair-pinning of VM-to-VM routed communication by providing hypervisor-level routing functionality. The NSX controller manages (with help of control VM) the routing updates to each hypervisor. This ensures a direct communication path even when the endpoints belong to separate Logical Switches (e.g., IP subnets).



**Figure 43 - Distributed Routing**

Figure 44 shows the distributed logical routing components on left and centralized Edge Services Gateway on right.

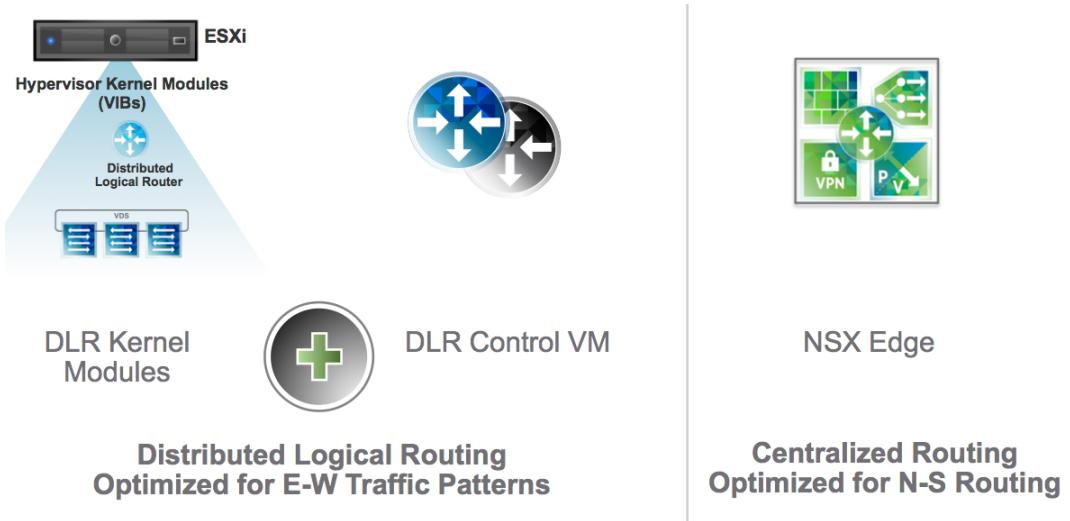
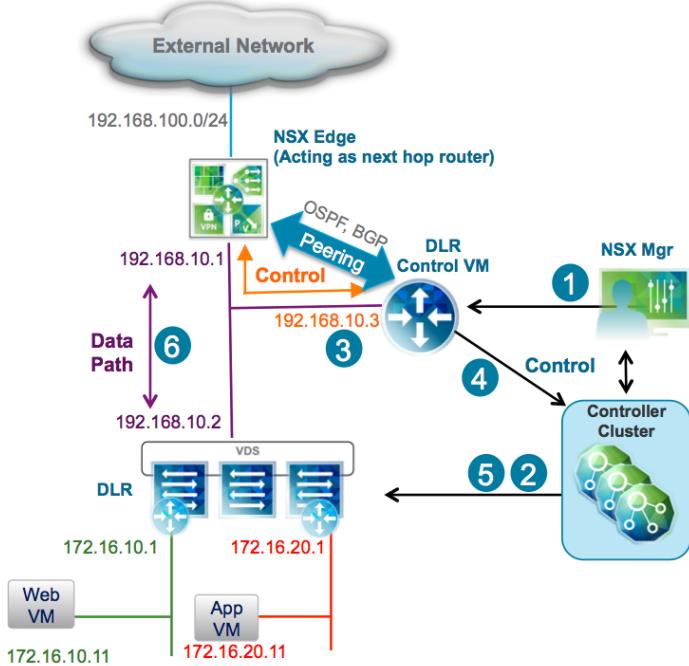


Figure 44 - Distributed Routing

Distributed routing is provided by a logical element called Distributed Logical Router (DLR). The DLR is essentially a router with directly connected interfaces to all hosts where VM connectivity is required. The supervisory function (i.e., control plane) to control the forwarding is imported from a control VM. The DLR consists of two primary components:

- **Control Plane:** The control plane is provided by the DLR Control VM and NSX controller. The control plane supports dynamic routing protocols (e.g., BGP, OSPF), exchanges routing updates with the next layer 3 hop device (e.g., NSX Edge), and communicates with the NSX manager and the controller cluster. High availability for the DLR control VM is supported through an active/standby configuration.
- **Data Plane:** DLR kernel modules (e.g., VIBs) are installed on the ESXi hosts part of the NSX domain. The kernel modules are similar to the line cards in a modular chassis supporting layer 3 routing. These kernel modules have a routing information base (RIB) that is pushed through the controller cluster. The traditional data plane functionality of route and ARP lookups is performed by the kernel modules. The kernel modules are equipped with logical interfaces (LIFs) connecting to the different logical switches. Each LIF has an IP address representing the default IP gateway for its logical L2 segment as well as a vMAC address. The IP address is unique per LIF and remains same where the logical switch exists. The vMAC associated with each LIF remains the consistent in each hypervisor as well and thus during the vMotion, the default gateway and MAC remains the same.

Figure 45 shows the interaction between logical routing components to enable distributed routing.



**Figure 45 - Logical Routing Components**

1. A DLR instance is created from the NSX Manager UI or via API calls. Routing is enabled, leveraging the protocol of choice (e.g., OSPF or BGP).
2. The controller leverages the control plane with the ESXi hosts to push the new DLR configuration, including LIFs and their associated IP and vMAC addresses.
3. Where a routing protocol is also enabled on the next hop layer device, OSPF/BGP peering is established with the DLR control VM. Routing information is exchanged between devices. In this example, the DLR peers with an NSX Edge.
4. The DLR Control VM can be configured to redistribute into OSPF the IP prefixes for all the connected logical networks; 172.16.10.0/24 and 172.16.20.0/24 in this example). It then pushes those routes advertisements to next hop router. The next-hop for those prefixes is not the IP address assigned to the control VM (192.168.10.3), rather the IP address identifying the data plane component of the DLR (192.168.10.2). The former is called the DLR protocol address, whereas the latter is the forwarding address.
5. The NSX Edge pushes the prefixes to reach IP networks in the external network to the control VM. A single default route is sent by the NSX Edge, since it represents the single point of exit toward the physical network infrastructure.
6. The DLR control VM pushes the IP routes learned from the NSX Edge to the controller cluster.
7. The controller cluster distributes routes learned from the DLR control VM across the hypervisors. Each controller node in the cluster takes responsibility for distributing the information for a particular logical router

instance. In a deployment where multiple logical router instances are deployed, the load is distributed across the controller nodes. A separate logical router instance is usually associated to each deployed tenant.

8. The DLR routing kernel modules on the hosts handle the data path traffic for communication to the external network via the NSX Edge.

The required steps to establish routed communication between two virtual machines connected to separate logical segments are shown in Figure 46.

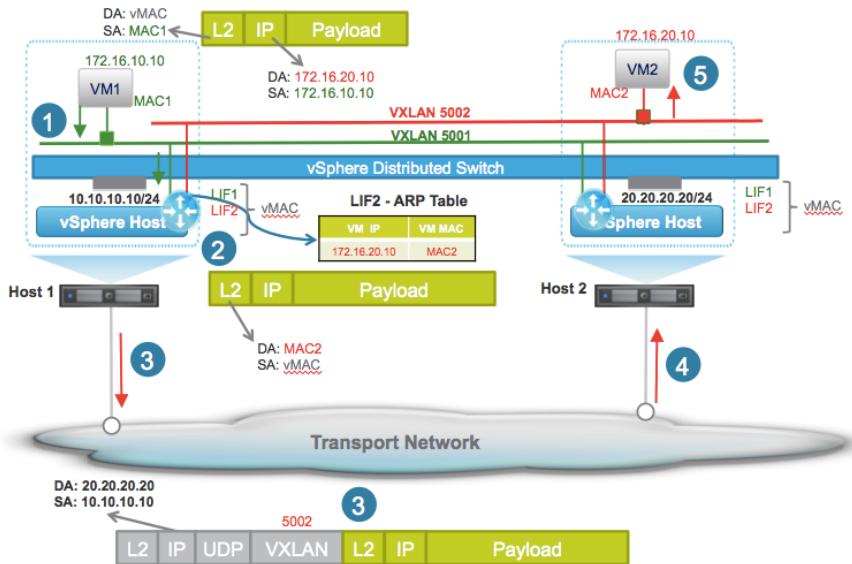


Figure 46 - Routed Communication between Virtual Machines

1. VM1 sends a packet to VM2 connected to a different VXLAN segment. The packet is sent to the VM1 default gateway interfaces located on the local DLR (172.16.10.1).
2. A routing lookup is performed on the local DLR, which determines that the destination subnet is directly connected to DLR LIF2. A lookup is performed in the LIF2 ARP table to determine the MAC address associated with the VM2 IP address. If the ARP information is not available, the DLR will generate an ARP request on VXLAN 5002 to determine the required mapping information.
3. An L2 lookup is performed in the local MAC table to determine how to reach VM2. The original packet is VXLAN encapsulated and sent to the VTEP of ESXi2 (20.20.20.20).
4. ESXi2 decapsulates the packet and performs an L2 lookup in the local MAC table associated to VXLAN 5002 segment.
5. The packet is delivered to the destination VM2.

When VM2 replies to VM1, the routing between the two logical segments would be performed on ESXi-2 where VM2 is connected. This represents the normal behavior of NSX DLR, which is to always perform local routing on the DLR instance running in the kernel of the ESXi hosting the workload that initiates the communication.

Figure 47 shows the sequence of steps required for establishing ingress communication from external networks to logical segments connected to the DLR.

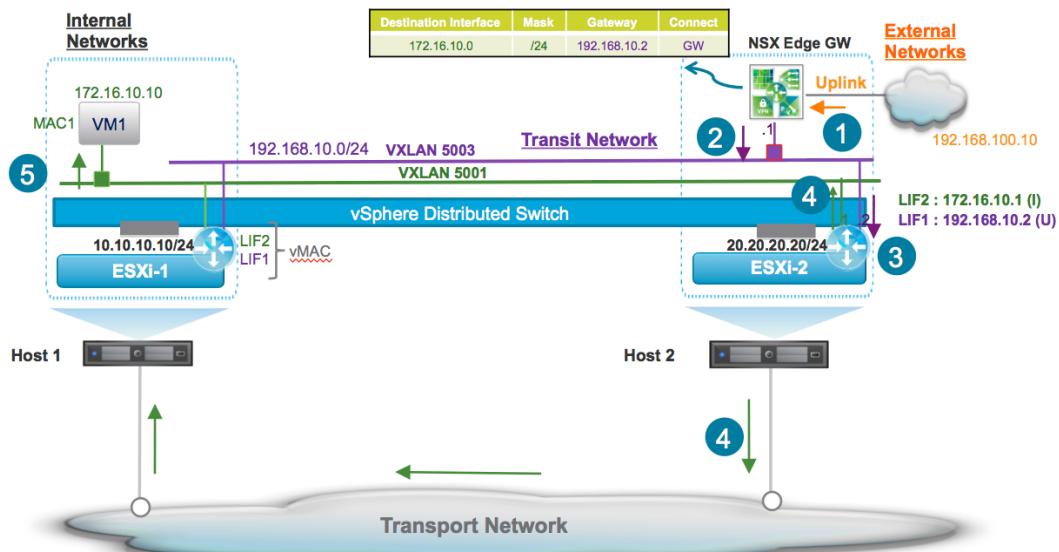
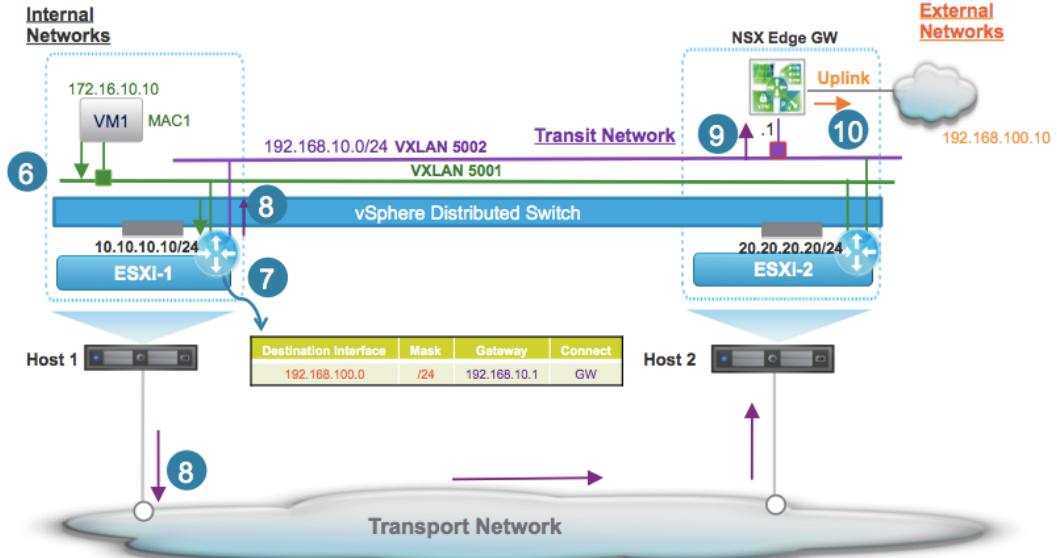


Figure 47 - DLR: Ingress Traffic from External Networks

1. A device on the external network (192.168.100.10) wants to communicate with VM1 on VXLAN 5001 segment (172.16.10.10).
2. The packet is delivered from the physical network to the ESXi server hosting the NSX Edge. The NSX Edge receives the packet and performs a routing lookup, finding a route to the 172.16.10.0/24 subnet via the DLR IP address on the transit network. The IP prefix was learned via a routing exchange with the DLR control VM. The next-hop (192.168.10.2) represents the IP address of the DLR on the data.
3. The DLR is also installed in the kernel of ESXi-2, where the active NSX Edge is deployed. Both routing pipelines, at the NSX Edge level and at the DLR level, are thus handled locally on ESXi-2.
4. The destination IP subnet (172.16.10.0/24) is directly connected to the DLR, so the packet is routed from the transit network into the VXLAN 5001 segment. After an L2 lookup, the packet is encapsulated in a VXLAN header and sent to the VTEP address (10.10.10.10) where VM1 resides.
5. ESXi-1 decapsulates the packet and delivers it to the destination.



**Figure 48 - DLR: Egress Traffic to External Networks**

6. VM1 replies to the external destination (192.168.100.10). The packet is sent to VM1 default gateway interface located on the local DLR (172.16.10.1).
7. A routing lookup is performed at the local DLR, which determines that the next hop to the destination is the NSX Edge interface on the transit network (192.168.10.1). This information is received on the DLR control VM from the NSX Edge and pushed to the kernel DLR module by the controller.
8. An L2 lookup is performed to determine how to reach the NSX Edge interface on the transit network. The packet is VXLAN encapsulated and sent to the VTEP of ESXi2 (20.20.20.20).
9. ESXi-2 decapsulates the packet and sends it to the destination (NSX Edge).
10. The NSX Edge performs a routing lookup, then sends the packet to the next L3 hop on the physical network. The packet will then be delivered by the physical infrastructure to the final destination (IP 192.168.100.10).

### 4.3.2 Routing Capabilities in NSX

This section examines routing protocol choices and basic capabilities offered in NSX. Routing in an NSX environment is the combination of east-west routing done by the NSX DLR, north-south routing from the NSX ESG. The optimized NSX topology leverages the capabilities of distributed logical routing as well as ESG. Concepts central to these capabilities include:

- Localizing routing and forwarding in hypervisor reduces the oversubscription on uplinks from hosts to access switch. This provides the potential for greater

host density for the same amount of bandwidth by eliminating the hair pinning of traffic.

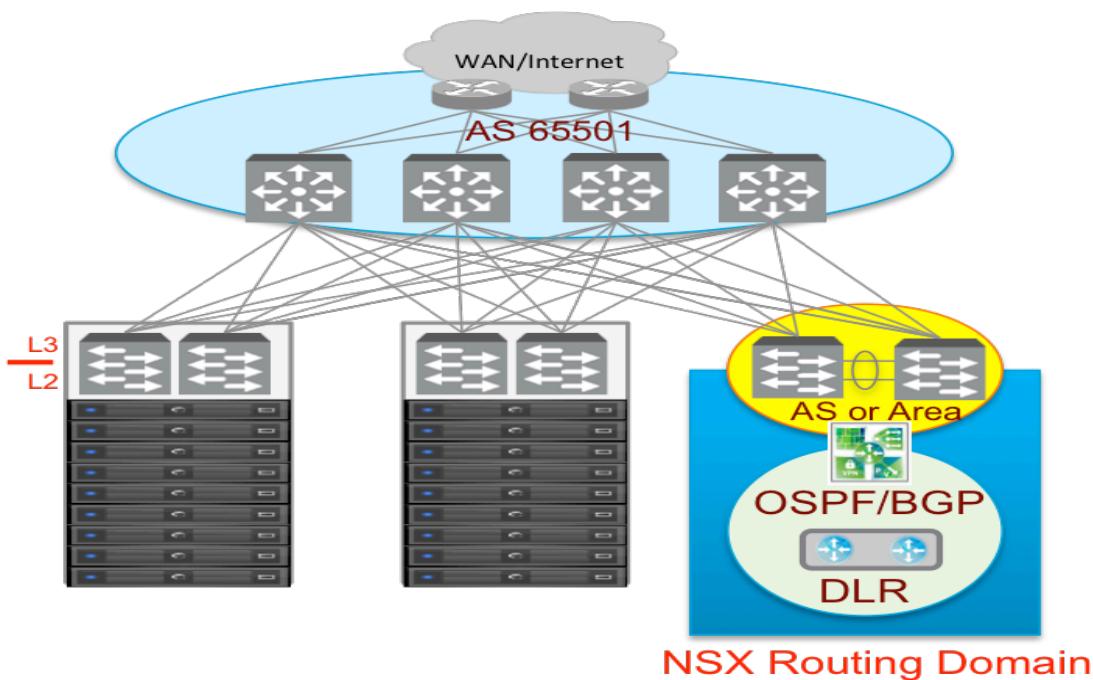
- Localization of forwarding inside the hypervisor allows higher speed transfer of data, potentially reducing the time to transfer data and improving latency. Localized forwarding in hypervisor also removes uplink bandwidth constraint.
- Building optimized topology in which forwarding is solely in hypervisor (i.e., software defined) between VXLAN segments, independent of underlying physical topology. This is fundamental to automation, flexibility, and on demand provisioning of edge services required for self-service IT and multi-tenancy.

Both the NSX DLR and the NSX ESG support OSPF and BGP. The primary difference between these routing protocols is the level of control on routes propagation and attribute manipulations they allow, with BGP being the more flexible of the two. The concept of an Autonomous System (AS), defined as a group of routers under the same administrative domain, comes also into play. Routing inside an AS is usually done via an IGP (e.g., OSPF) while routing between Autonomous Systems is done via an EGP (e.g., BGP).

In typical deployments the routing protocol choice is already decided. The corporate routing team typically handles the backbone and core WAN/Internet domains with its administrative control extending to core/aggregation datacenter routers. The data center team usually handles the routing protocol decision below the core/aggregation boundary. In most data centers OSPF is the prevalent choice to maintain a consistency with core routing protocol; however, in modern, large scale data centers, BGP is the de facto protocol of choice – specifically those with spine-leaf topology. The NSX routing protocol selection and its interaction with existing network routing services should be evaluated with following critical understanding:

- The NSX routing domain connects to existing network as an edge network. The NSX domain does not act as transport network, thus simplifying the connectivity and features required in a typical routing protocol to support a stub topology.
- NSX does not dictate the routing protocol choice in itself, however the design requirements and connectivity options may restrict the choice to a protocol.

Figure 49 illustrates the connectivity of NSX domain to an existing data center fabric. The physical edge connectivity is dependent on the specific type of data center topology. For spine-leaf, it connects to border-leaf. For a L3 routed data center to ToR. Finally, for classical three tiers topology, it can connect either to an aggregation or access switch. The key concept in each case is that NSX acts as stub (i.e., leaf) network.



**Figure 49 - NSX Domain Connectivity with Datacenter Routed Fabric**

The routing protocol best practices that are incorporated in this design guide are as follows:

- Use summarization to reduce the routing table size, decrease churn, and improve convergence. The routing protocol configuration at the border leaf, layer 3 ToR, or aggregation should only advertise reachability of NSX domain via summarized networks/subnets managed under NSX routing domain.
- Use a default route to reach any destination outside the NSX routing domain. NSX is stub network topology and does not need to know all the routes that exist external to its control. The physical router should advertise only a default route derived either from core routers or known Internet exit points.
- The non-default neighbor hello/hold timers can be used to achieve faster convergence in ECMP based topologies.
- Use routed (or SVI) links between pair of physical routers (both top-of-rack and aggregation), to heal the routing protocol connectivity in case of loss of all uplinks from a router/switch (see Figure 116). This is highly recommended in OSPF but not limited to only that protocol.

NSX offers flexibility in configuring a variety of options and combinations of routing protocols for ESG and DLR. The choice of a protocol, its features and control of route propagation is highly dependent on corporate preference and policy. This design guide focuses on simplicity; follow best practices that are well understood to provide consistent connectivity for NSX segments. With this understanding, a use of single protocol for both DLR to Edge VM and Edge VM to physical connectivity is proposed - either OSPF or BGP. These two designs are optimal and suitable most organizations. Mixing BGP for Edge to physical and OSPF for DLR to Edge is possible, however it will add complexity and

complicate operational troubleshooting. If the core/aggregation is running non-standard protocol, then it is recommended to run eBGP from Edge to physical. If the OSPF in core/aggregation is running non-standard configuration, then again eBGP is the right choice from Edge to physical. The BGP section further provides the justification of why eBGP solves many combination of connectivity and it's the recommended protocol of choices.

#### 4.3.3 OSPF and NSX Connectivity Options:

For OSPF we assume that the reader is familiar with the basic of how routing protocols works (SPF), the data structure it relies on (LSA Database) and with other concepts like adjacency and areas.

Simplicity, consistency, and reduced operational burden are the primary motives for selecting an end-to-end OSPF topology for NSX. If the existing organizational administrative policy requires OSPF for the general network, it is recommended to keep configuration choice for the DLR to Edge the same as the Edge to the physical network.

Table 3 details OSPF feature capabilities of Edge and DLR as well the default timers:

Protocol Feature	Edge HA	Edge ECMP	DLR Control VM
<b>Area Support</b>	Regular & NSSA	Regular & NSSA	Regular & NSSA
<b>ABR Support</b>	Yes	Yes	NA
<b>Redistribution</b>	N2 (if NSSA) else E2	N2 (if NSSA) else E2	N2 (if NSSA) else E2
<b>Hello</b>	10	10	10
<b>Dead</b>	40	40	40
<b>Priority</b>	Yes	Yes	Yes
<b>Cost</b>	Yes	Yes	Yes
<b>Graceful Restart</b>	Yes	NA	Yes

Table 3 – OSPF Features and Supported Functions

The default OSPF values may not be desired in all types of Edge deployed. For active-standby Edge the default timer may not be adequate for certain types of

failures. These timers should be matched with the physical routers to form an adjacency; this is critical for maintaining a graceful recovery during primary Edge failure. Details of this recovery and other failure statuses are described in the [“NSX Edge Deployment Considerations”](#) section. The ECMP and DLR control VM timers are also configurable, supporting shorter timers for faster recovery.

Regarding route redistribution, the redistribution into OSPF will generate either OSPF external E2 (type 2) or N2 routes for NSSA.

### **Area Design Considerations:**

OSPF connectivity and area design depends on individual corporate routing policy. Typical choices in area design as include:

- Assuming layer 3 routed topology, whether area 0 can be extended to the top of rack where the NSX Edge is connected.
- Multi-tenant design in which area 0 must be extended to the ToR for routed topology or for layer 2 topology, the Edge connectivity must be extended to aggregation routers to allow for a separate area per tenant.
- If the NSX Edge is part of the existing area for data center and the Edge is peered with non-ABR router – either a regular or NSSA area – the area properties will be inherited for routes summarization and default routes. If the NSX becomes part of existing area, then the NSX Edge must match the area type and its limitations. This may place restrictions on filtering capabilities for reducing LSAs or routing table size; thus it is preferable to allocate a dedicated area for NSX routing domains.

It is recommended is to run an ABR on a physical router, either at the top of rack or aggregation to simplify NSX connectivity. This provides consistent operational control established for other areas attached to the same physical ABR.

In addition to supporting regular OSPF area, both NSX DLR and NSX ESG support NSSA. This is a stub area where redistribution of external routes (e.g., connected, static) is permitted. Redistributed routes will generate type 7 LSAs which will be converted by ABRs into type 5 LSAs, then flooded to other areas permitting LSA 5 propagation. An NSSA area can work either as a stub or a totally stubby area. For a standard stub area, all routers belonging to the area must be configured as such and type 3 LSAs will be accepted in the area. For an NSSA totally stubby, an additional configuration is needed only on the ABR routers to permit simply a summary type 3 LSA (e.g., default route). Additionally, the configuration to declare an area being NSSA must be done on every router of the area, while the configuration to declare a totally stubby NSSA is done only on the ABR routers. This later design choice fits well with the recommendation of using physical routers as ABR. The table below summarizes the area type, associate behavior of LSA and default configuration for each area.

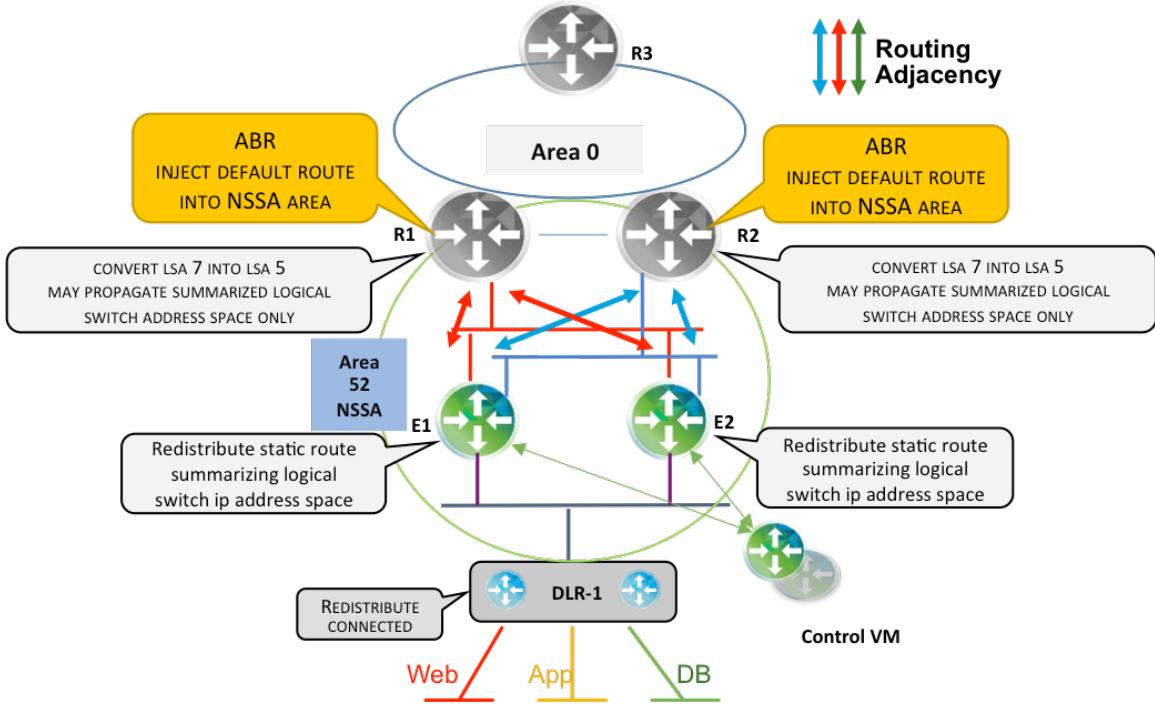
Area	LSA Type allowed inside the area	Generate AS External Route LSA	NSSA ABR Behavior	Default Route Behavior to NSSA area
<b>NSSA</b>	All Type 3, no type 4 and 5	Type 7	Translate type 7 to 5 and propagates to other area as type 5	Need to generate
<b>Totally NSSA</b>	Filters 3 & 4 (no inter-area) and 5. One type 3 summary LSA	Type 7	Translate type 7 to 5 and propagates to other area as type 5	Automatic as type 3 summary LSA

Table 4 – LSA Types and Behavior with an Area Type

If routing between tenants is desired in a multi-tenant design where more than one NSX routing domain is attached to same the OSPF ABR routers, then a default route is needed on the ESGs. This is because OSPF areas are configured as NSSA – either totally stubby or normal – and the LSA related to the remote NSX tenants' logical networks cannot be imported. (as they become LSA type 5 which be imported - see above table, second column) Thus the default route generation at the ABR is mandatory in an all NSSA (e.g., multi-tenant all NSSA design).

Key design concepts of this recommended topology include:

- If the OSPF is selected as edge connectivity option, keep the same protocol for DLR to ESG connectivity.
- The ABR is always a physical router and the area for NSX routing domain is NSSA. This is applicable to any data center physical topology
- Summarization techniques available at both ABR and ESG stages are used to adopt the best practices. If the summarization of NSX network is not possible due to non-contiguous address space, the design still applies for sending default routes to NSX routing domain.



**Figure 50 – OSPF End-to-end Connectivity**

Figure 50 depicts ECMP-based connectivity. The gray routers R1 and R2 can represent either L3-capable ToR/EoR switches that serve as ABRs and are connected to the datacenter backbone OSPF area 0, or L3-capable switches positioned at the aggregation layer of a traditional pod-based datacenter network architecture that serve as the L2/L3 demarcation point. ESG1, ESG2 and DLR-1 are in the same NSSA area 52.

The VLANs configuration for Edge deployment is discussed in detail in the [Edge VM Connectivity and Availability Design](#) section. Typically, one VLAN is sufficient in HA mode while two are recommended in an ECMP based design.

This design and recommendation is applicable to both active-standby and ECMP deployment. See the best practices for each in the [Edge Design and Deployment Considerations](#).

### Route Advertisement for North-to-South Traffic:

This section describes how to advertise logical networks to north bound so external users or application traffic (north-to-south) can access the required resources in NSX domain.

- DLR-1 redistributes its connected interfaces (i.e., logical networks) into OSPF as type 7 LSAs. This provides immediate availability inside this area as well as for other non-stub areas.
- As part of best practices, ESG1 and ESG2 each have a static route pointing to DLR that summarizes the entire logical switch IP address space. This is then redistributed into the OSPF area as a LSA type 7. This route will be installed by both R1 and R2 and will be used to ensure forwarding of north-

south traffic. This static summary route is an optimization though not technically required.

- With the static route or routes representing the logical network address space, the requirement of redistributing the logical network connected to DLR can be eliminated. This reduced the configuration burden on ESGs. In addition, it allows a reduction in configuration optimization at physical routers as well as increases overall availability. This design recommendation should be adapted only if the contiguous address space is available; otherwise the redistribution of connected interface is necessary.
- This is essential for maintaining the reachability of NSX logical networks during active control VM failure with aggressive timers in an ECMP-based design. See the [Edge Design and Deployment Considerations](#) section for additional details. It is not required for an ESG active-standby, stateful services design where aggressive timers are not used.
- These routes will reach R1 and R2 as type 7 LSAs where they will be translated into type 5 LSAs. By default, one of the two available ABRs will actively inject the translated LSA into the OSPF domain. Load sharing on R1 and R2 of incoming traffic sourced from other areas will depend on other areas' routing availability the translated LSA's forwarding address. This can happen regardless of the number of active ABR translators.
- If further optimization with area summarization is desired, it is recommended that only the summary route injected by ESG1 and ESG2 is translated and propagated into other areas. The following optimization is necessary only if redistribution of connected networks is enabled at the ESGs. Those specific networks will be available in the ABR routing table, so suppressing announcement towards other area becomes necessary for the optimization. Furthermore, the options are not required if static route summary is enabled at ESGs while redistribution of connected is not enabled at DLR. Such a configuration would already eliminate the specific DLR networks from appearing in the ABR routing table. The optimization process is as follows:
  - Configure the ABRs (R1 and R2) to inject only the summary route covering entire range of IP networks redistributed by the DLR (e.g., Cisco's *summary-address <subnet> <netmask>* command), thus suppressing all the specifics. As long as the summary route correctly summarizes the DLR logical networks, this configuration will not need further configuration changes with new logical network deployments.
  - Depending on the specific implementation it is possible that the active translator ABR will automatically install a static route for the summarized address space pointing towards a null 0 interface. This static route would be preferred over the summary routes coming from ESG1/ESG2 for the same address space. In the case of a DLR control VM failure, it may be necessary to disable the automatic generation of this static route to null 0 (e.g., Cisco's *no discard-route external* command) on the ABR in order to

- preserve packet forwarding towards DLR logical networks. This is a one-time configuration.
3. Configure the ABRs (R1 and R2) to not propagate the routes for the more specific subnets redistributed by the DLR (e.g., Cisco's *summary-address <subnet> <netmask> not-advertise*). As no external network summarization is active on R1 and R2, the static route towards a null 0 interface will not be created. With this configuration – which is required if “redistribute connected” is enable at the DLR – and a proper summarization at ESGs, it is possible to inject into the OSPF domain only the type 5 LSA referring to the summary route redistributed by ESG1 and ESG2. This type 5 LSA will retain the forwarding address of the original type 7 LSA. These will further help in providing two forwarding paths towards R1 and R2. This is a preferred method of configuration to avoid intricacies related to managing vendor specific implementations on ABRs.

#### **Routes Advertisement for South-to-North Traffic:**

- Providing a default route to the NSX domain is necessary so that NSX can reach the ABR without sending entire routing table to Edge and DLR.
- In this design R1 and R2 are the ABRs. They inject the default route into the NSSA area; neither ESG1/ESG2 nor DLR-1 generate it. A specific configuration will be needed to correctly inject the default route onto both R1 and R2 depending on the area's state as an NSSA (e.g., Cisco's *area <area-id> default-information originate*) or totally stubby NSSA. The default route will be injected by the virtue of declaring the NSSA as being totally stubby at the ABR (e.g., Cisco's *area <n> nssa no-summary*). This default route will also be used to route traffic towards any other tenant NSSA areas as reachability of those networks is propagated in the OSPF domain via type 5 LSAs which are not allowed inside an NSSA area – either standard or totally stubby.
- It is important to confirm that R1 and R2 do not push a default route towards ESGs if they get disconnected from the OSPF backbone area 0. Depending on the implementation and the support of active backbone detection capabilities, it may be sufficient for a router to have an interface belonging to area 0 and in an up state to keep generating the default route even if no OSPF adjacency is established across that interface (e.g., loopback interface).
- If R1/R2 keep generating the default route without an active connection towards area 0, it is important to ensure a direct path between R1 and R2 exists and can be leveraged whichever is disconnected from area 0. This direct path should not go through ESG1 or ESG2 because the disconnected ABR is still generating a default route towards both the ESGs. This will result in the black holing of traffic forward to the router that is disconnected from backbone.

#### **4.3.4 BGP and NSX Connectivity Options:**

BGP is becoming a preferred protocol in modern data center spine-leaf topology. Common BGP design choices for large-scale data centers are documented at

<https://datatracker.ietf.org/doc/draft-lapukhov-bgp-routing-large-dc/>. Relevant best practices described in the RFC is adapted with specific connectivity of NSX treated as an edge/stub routing domain. The concept of BGP as protocol to interconnect autonomous system applies well when considering NSX as separate autonomous routing zone with its own internal routing supporting east-west forwarding. BGP is highly recommended for connectivity of ESG and DLR in the following situations:

- It is advantageous to avoid the complexity of OSPF area design choices, default route dependencies with multi-tenant, interconnect link requirements, and reachability of area 0.
- NSX is considered separate autonomous area.
- Top-of-rack is part of an existing L3 BGP Fabric
- Better route control is required.
- Multi-tenant design is required.

NSX supports both eBGP and iBGP with their associated properties. Table 5 explains the default values and capabilities of each protocol.

BGP Protocol Feature	Edge HA	Edge ECMP	DLR Control VM
<b>EBGP</b>	Yes	Yes	Yes
<b>iBGP</b>	Yes	Yes	Yes
<b>Redistribution</b>	Yes	Yes	Yes
<b>Keepalive</b>	60	60	60
<b>Hold</b>	180	180	180
<b>Multi-Path</b>	Yes	Yes	Yes
<b>Graceful Restart</b>	Yes	NA	Yes

**Table 5 – BGP Features and Supported Functions**

EBGP is typically used when exchanging routes with physical networks and iBGP between DLR and ESG within the NSX domain. The timer configuration and consideration as well failure and recovery option are described in the “[NSX Edge Deployment Considerations](#)” section.

Some simple considerations should be kept in mind when planning the ASN numbering schema:

- NSX currently supports only 2-byte autonomous system numbering (ASN). It is recommended to use private BGP AS numbers for NSX tenants. RFC 6996

describes a contiguous block of 1023 autonomous system numbers (64512-65534 inclusive) that have been reserved for private use. This should provide sufficient ASN for most implementations.

- As per RFC 4271, ASN 65535 is reserved. It is not a private AS, and any attempt to configure it will be rejected by UI/API.
- If required, it is possible to reuse the same BGP ASN if the physical router devices support BGP configuration capabilities such as “*neighbor <ip address> allowas-in <occurrences>*” and “*neighbor <ip address> as-override*”. Such a configuration is beyond the scope of this document. As private ASN must not be exported in the public domain, specific configuration must be made to physical eBGP peering towards the public Internet (e.g., Cisco’s “*neighbor <ip address> remove-private-as <replace-as>*” or “*set neighbor <ip address> remove-private*”).

For the multi-tenant design with BGP, a summary route for the NSX tenant’s logical networks is advertised from the ESGs to the physical routers through an EBGP session. A proper outgoing BGP route filtering policy permits this advertisement to be sent only towards those physical routers; an IGP (e.g., OSPF) cannot provide explicit LSA filtering within an area. This summary route is then propagated to the other NSX tenants and on to the rest of the network.

The recommended topology covers key design attribute as follows:

- If the BGP is selected as Edge connectivity option, keep the same protocol for DLR to ESG connectivity.
- Use eBGP between ESG and physical and iBGP between DLR and ESG
- The iBGP full mesh requirement can be relaxed as the connectivity is straight north to south, and ESG is not acting as transit device for its own AS.
- Summarization techniques available at each stage should be used to adopt the previously detailed best practices. If the summarization of the NSX network is not possible due to non-contiguous address space, the design still applies for sending default routes to the NSX routing domain.
- For a default route that is passed by physical routers towards the DLR, the BGP next-hop reachability towards the physical router interface IP is required. This is achieved via redistributing connected interfaces (i.e., subnet of links connected to physical router and uplinks of ESGs) of the ESG(s) into the DLR.
- Edge(s) announces summary routes for the NSX DLR segments over eBGP to physical routers.
- DLR control VM announces connected subnets via iBGP.

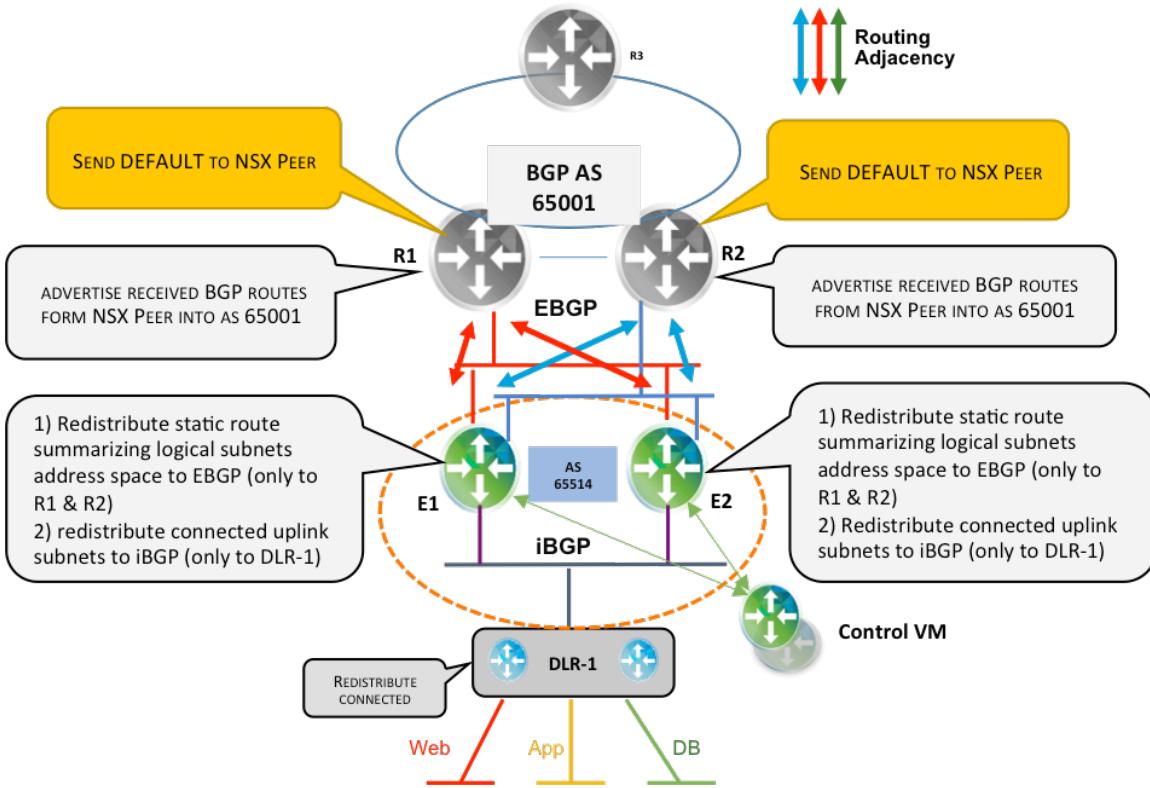


Figure 51 – BGP End-to-end Connectivity

Focusing on the NSX tenant in ASN 65514, some common design specifications can be highlighted. ESG1, ESG2 and DLR-1 all share the same iBGP ASN. ESG1 and ESG2 each have eBGP sessions with both R1 and R2 over their respective direct network connections, and also an iBGP session with DLR-1. ECMP is enabled on both ESG1/ESG2 and DLR-1 to enforce load balancing across all available paths.

This design and recommendation is applicable to both active-standby and ECMP deployment. See the best practices for each in [NSX Edge Deployment Considerations](#).

### Routes Advertisement for North-to-South Traffic:

DLR-1 injects into iBGP the connected logical subnets address space. ESG1 and ESG2 both have a static route summarizing the whole logical address subnet space pointing to DLR-1. This route is injected into BGP and using BGP neighbor-filtering capabilities and is advertised only to R1 and R2 via EBGP. Figure 52 shows outbound BGP filtering where 11.1.0.0/16 is the summary route.

BGP Filters :

Direction	Action	Network	IP Prefix GE	IP Prefix LE
Out	Permit	11.1.0.0/16		
Out	Deny	172.16.193.0/28		
Out	Deny	172.16.193.16/28		

3 items

Figure 52 – Outbound Route Filtering

The static routes representing entire NSX logical network space provide the following advantages:

- Reducing the configuration burden of advertising new logical networks introduced via automation.
- Avoiding routing churn caused by each new advertisement.
- Maintaining reachability of NSX logical networks during control VM failure with aggressive timers in ECMP based design. It is not required for an ESG active-standby, stateful services design where aggressive timers are not used.

If summarization is not possible, the logical networks space must be redistributed via explicit configuration.

### Routes Advertisement for South-to-North Traffic:

To reach any destination outside its own AS, the DLR follows the default route. This is derived either from backbone/core routers or advertised via “*network 0.0.0.0*” command by respective BGP peers when 0/0 is present in the routing table. The key requirement is to follow the route advertisement based on reachability of 0.0.0.0/0 in the physical router’s routing table.

In eBGP/iBGP route exchange, when a route is advertised into iBGP, the next hop is carried unchanged into the iBGP domain. This may create dependencies on external routing domain stability or connectivity. To avoid external route reachability issues, the BGP next-hop-self feature or redistribution of a connected interface from which the next hop is learned is required. The BGP next-hop-self is not supported in current implementation, thus it is necessary to redistribute the ESG uplink interface (e.g., two VLANs that connect to physical routers) into the iBGP session towards the DLR. Proper filtering should be enabled on the ESG to make sure the uplinks’ addresses are not advertised back to physical routers as this can cause loops/failures.

Usually an iBGP domain calls for a full mesh of iBGP sessions between all the participant routers. This is due to iBGP loop avoidance protection that forces an iBGP router to not advertise routes learned by another iBGP router. Given that NSX is a stub iBGP topology, this best practice can be relaxed. However, a condition can exist where one of the ESGs is isolated or loses its EBGP session while a second ESG within the iBGP domain is keeping EBGP session with physical routers. This situation would cause the injection of the default 0/0 into

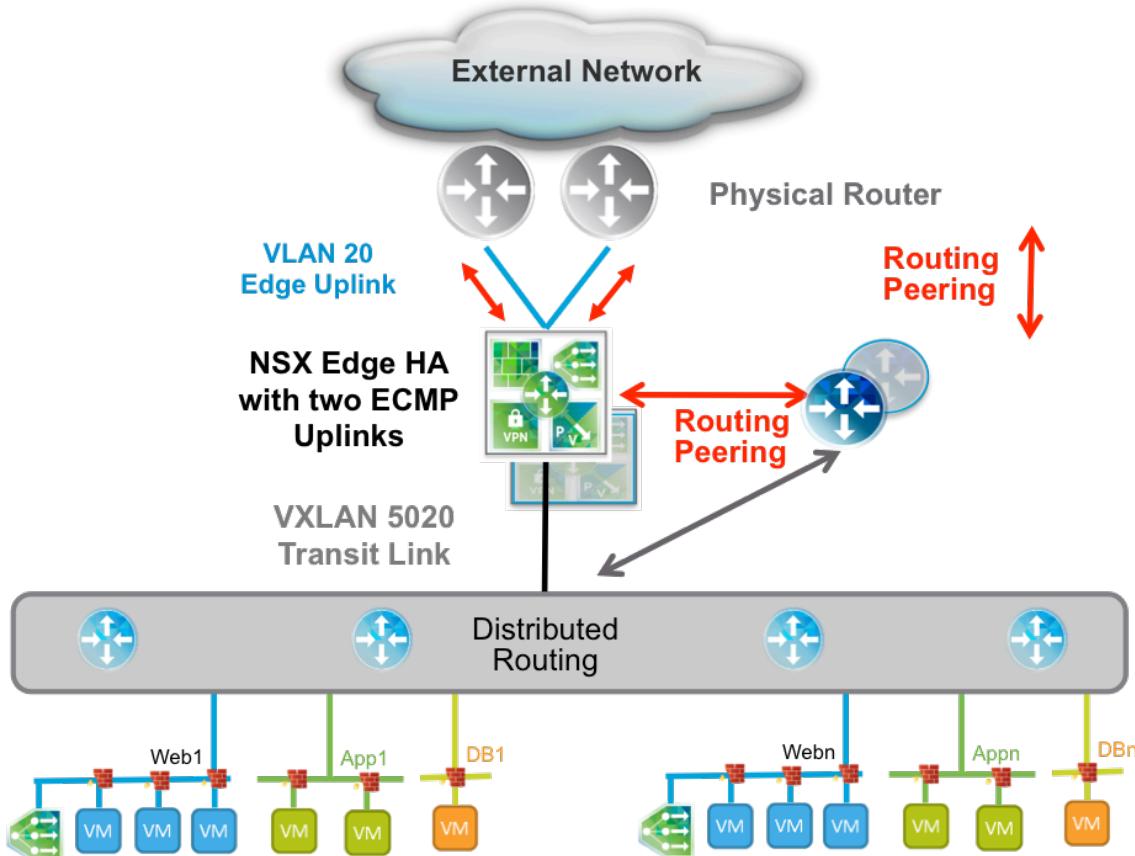
the iBGP domain and its ultimately visibility on the DLR. The DLR will not be aware of this change and will simply forward to both ESGs because the default route is carrying physical router's IP via uplink VLANs. One of the ESGs cannot forward the packet, so to avoid this failure state either run a full iBGP mesh or disable the redistribution of uplinks on the ESG requiring needing isolation along with eBGP session shut to physical routers.

#### Metric and Admin Distance:

As of release 6.2, NSX supports configuration admin distances. This functionality compares priorities for specific routing protocols in order to select the best route for a given prefix. This configuration is useful when static routes are used as backup for dynamic routes. One such use case is during the active control VM failure in ECMP Edge model described in details in [NSX Edge Deployment Considerations](#) section.

#### 4.3.5 Enterprise Routing Topology

The introduction of the NSX Edge services gateway between the DLR and the physical router, shown in Figure 53.



**Figure 53 - NSX Edge between DLR and Physical Router**

In this deployment model, the DLR control VM peers with an NSX Edge on a VXLAN transit link, while the NSX Edge uses its uplink interface to connect and

peer with the physical network infrastructure. The main upsides of this approach are:

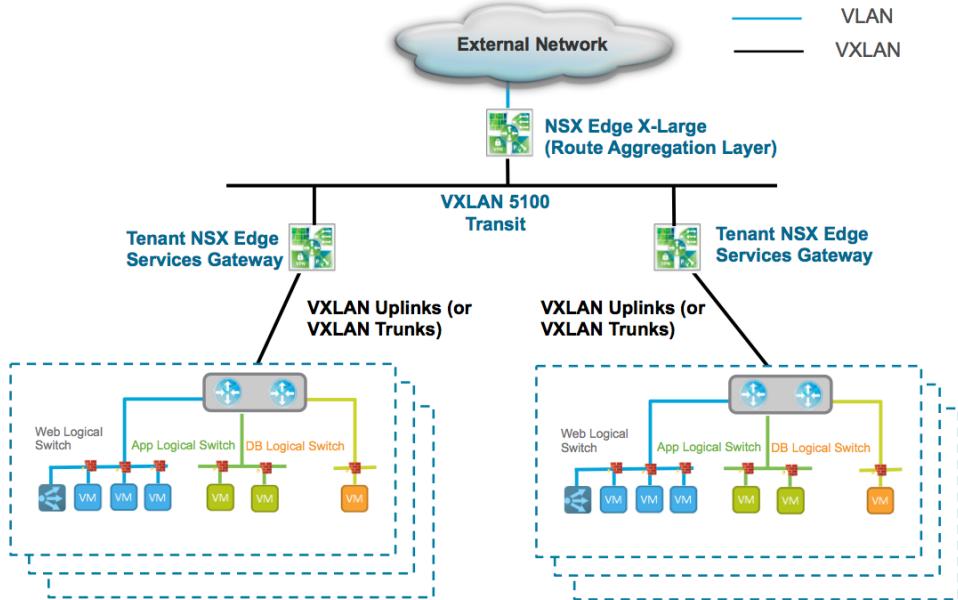
- The use of VXLAN between the DLR and the NSX Edge eliminates dependency on the physical network for connectivity between the compute ESXi hosts – where DLR routing happens for outbound traffic – and the NSX Edge. Traffic will instead be VXLAN encapsulated in the data plane.
- The establishment of route peering between the logical space and the physical network can be done in the initial configuration and provisioning phase. Deployment of additional DLRs in the logical space will not require further modification to the physical network, nor would change the number of routing adjacencies established with the physical routers. This exemplifies the decoupling between logical and physical networks promised by NSX.

It is clear that all north-south traffic is required to transit through the NSX Edge; this may represent a constraint in terms of bandwidth. The use of active/active ECMP capabilities in NSX Edge – introduced in release 6.1 – allows for scaling out of the available bandwidth for the on-ramp/off-ramp function provided by the NSX-Edge. For more information about the deployment of active/active ECMP NSX Edge, please refer to the [ECMP Edge](#) section.

In a service provider environment with multiple tenants, each tenant can have different requirements for volume of number of isolated logical networks and network services (e.g., load balancer, firewall, VPN). In such deployments, the NSX Edge services gateway provides network services capabilities along with dynamic routing protocol support.

As shown in Figure 54, two tenants are connected to the external network through the NSX Edge. Each tenant has its own logical router instance that provides routing within the tenant. The dynamic routing protocol configuration between the tenant logical router and the NSX Edge provides external network connectivity to the tenant VMs.

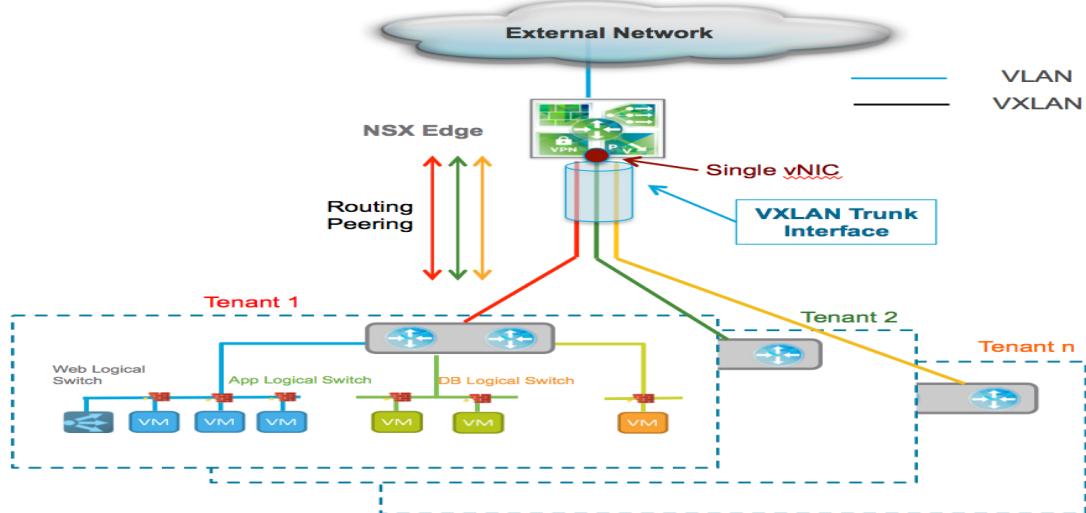
In this topology, the distributed router in the hypervisor handles east-west forwarding while the NSX Edge Services Gateway handles north-south traffic.



**Figure 54 - NSX Edge Services Gateway as Next Hop (Also Providing Network Services)**

This deployment model limits the maximum numbers tenants that can be connected to the same NSX Edge to nine. This limitation derives from the fact that the NSX Edge, as any other VM, is equipped with 10 virtual interfaces (vNICs). At least one of them is used as uplink interface, though two in recommended best practice; this would further limit maximum tenants to eight. Additionally, it is imperative not to have overlapping IP addresses between the tenants that connect to the same NSX Edge, unless the overlapping prefixes are and used only in the context of each isolated tenant (i.e., not advertised between the DLR and the NSX Edge).

NSX software release 6.1 allows for scaling up the number of tenants supported by a single NSX Edge services gateway with the introduction of the ESG trunk interface.



**Figure 55 - Use of Trunk Interfaces on ESG**

The definition of an ESG vNIC interface as trunk allows the connection of multiple DLR instances to that single vNIC, leveraging the sub-interface construct traditionally available on physical networking devices. This is depicted in Figure 55.

In this specific deployment model, only VXLAN sub-interfaces are needed, as the ESG trunk interface is used to aggregate connections to multiple southbound DLR instances.

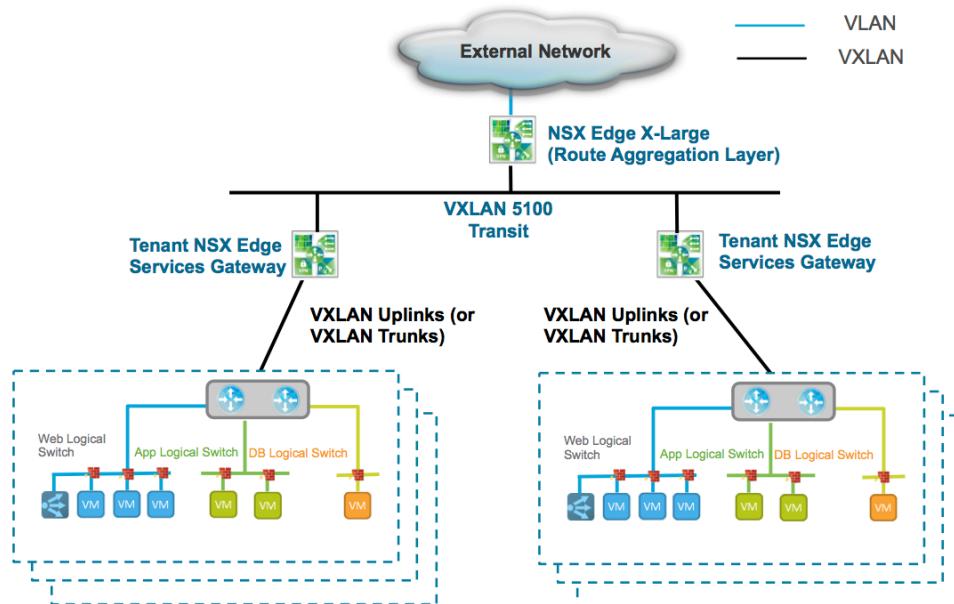
As of release 6.1.2, 200 sub-interfaces (i.e. tenants) are supported on a single ESG instance across all the available vNIC interfaces deployed as trunks. It is recommended to always check the release note for scalability values supported with a specific NSX software release.

Routing peering can be established on each trunk sub-interface. As shown in Figure 55, this allows the different DLR control VM instances to peer with a single ESG vNIC. Only BGP peering, in addition to static routing, is supported in the initial 6.1 release. OSPF peering is introduced in the 6.1.3 maintenance release.

## Scalable Topology

The service provider topology can be scaled out as shown in Figure 56. The diagram shows nine tenants served by the NSX Edge on the left and the other nine by the NSX Edge on the right. Service provider can easily provision additional NSX Edge to serve greater numbers of tenants.

The use of trunk interfaces would increase further scalability characteristics in terms of supported tenants for this deployment option.

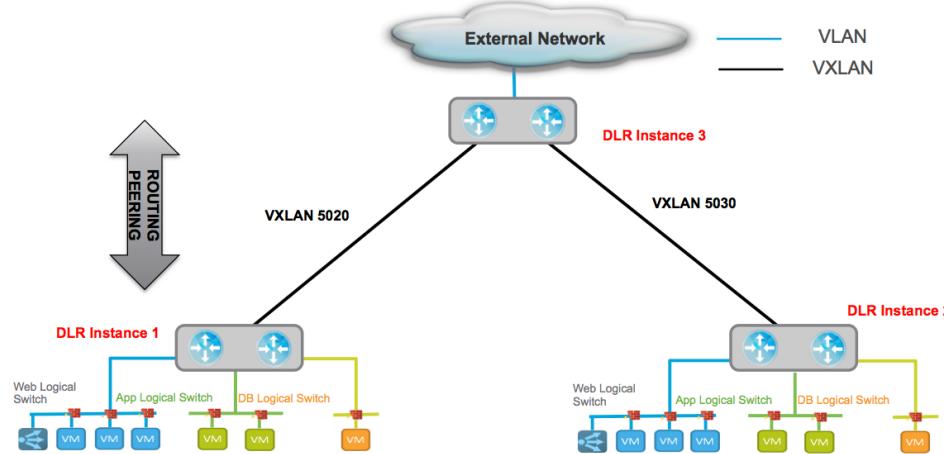


**Figure 56 - Scalable Topology**

In addition to increased scalability, this model allows for tenants with overlapping IP addresses to be deployed in separate groups connected to different NSX Edges. NAT must be performed on the first tier of NSX Edges to ensure that tenants using overlapping IP addresses can be distinguished once their traffic flows toward the aggregation NSX Edge.

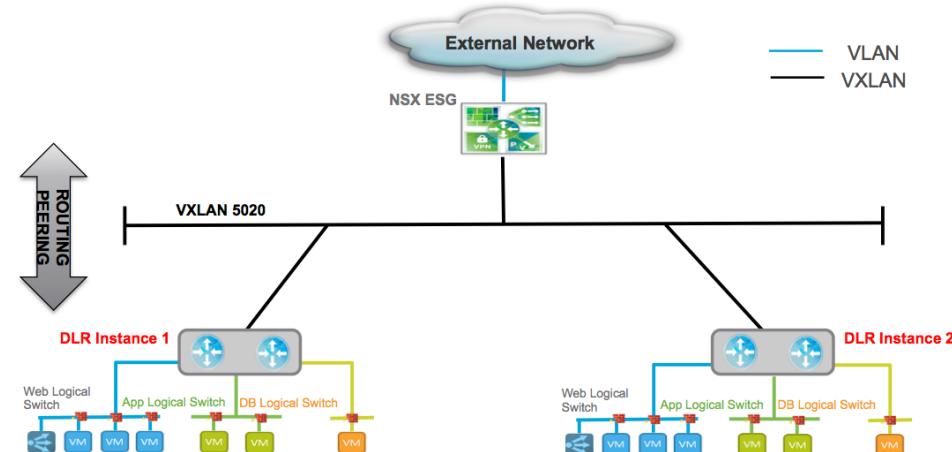
#### 4.3.5.1 Unsupported Topologies

While the deployment of logical routing components (e.g., NSX Edge, DLR) enables the building flexible multi-tiered routing topologies, it is generally forbidden to connect more than one DLR instance to the same VXLAN segment. Thus it is not possible to build a multi-tier logical routing topology using only DLR instances as shown in Figure 57; instead, DLR Instance 3 should be replaced with an NSX Edge.



**Figure 57 - Unsupported DLR Multi-Tier Topology**

Figure 57 diagrams another unsupported topology, two DLR instances peering with a northbound ESG on a common VXLAN segment. Each DLR instance should instead peer with the ESG on a separate VXLAN uplink, enabling separate DLR-Edge instances.



**Figure 58 – Multiple DLR to Single Edge Unsupported Topology**

## 4.4 Logical Firewalling and Security Services

The NSX platform supports two critical functionalities when it comes to securing a multi-tier workload. The first, its native supports for logical firewalling capability providing stateful protection of multi-tier workload. The second, NSX is a security services platform that enables multi-vendor security services and service insertion for the application workload protection. Logical firewalling is protecting a multi-tier workload is represented in Figure 59 below.

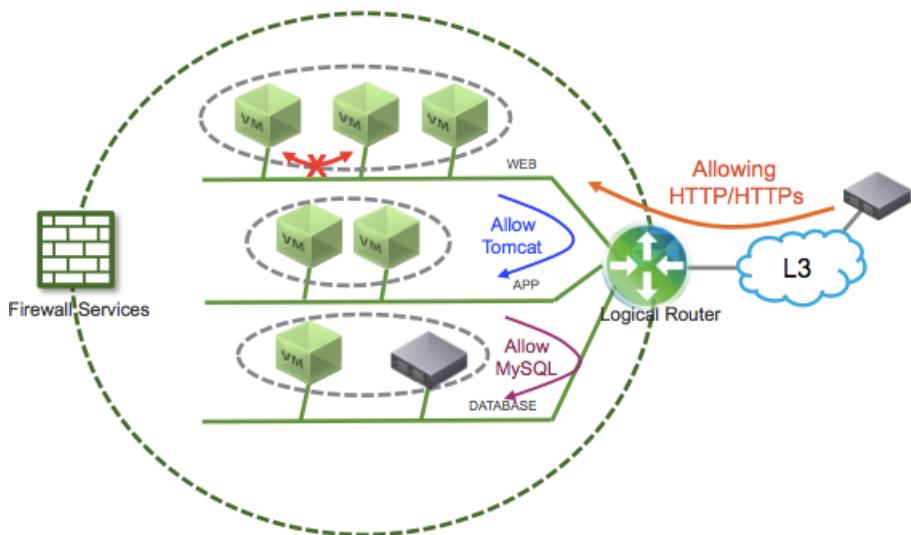


Figure 59 - Controlling Communication in a Multi-Tier Application

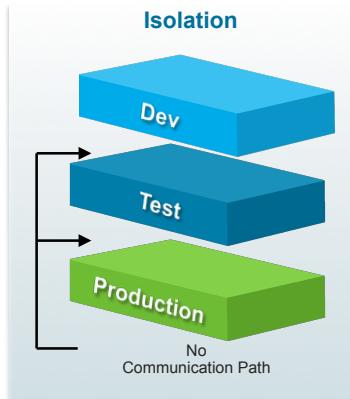
The VMware NSX platform includes two firewall components: a centralized firewall service offered by the NSX ESG; and a Distributed Firewall (DFW) enabled in the kernel as a VIB package on all the ESXi hosts part of a given NSX domain. The DFW provides firewalling with near line rate performance, virtualization, identity awareness, activity monitoring, and other network security features native to network virtualization.

### 4.4.1 Network Isolation

Isolation is the foundation of most network security, whether for compliance, containment, or isolation of development/test/production environments.

Traditionally ACLs, firewall rules, and routing policies have been used to establish and enforce isolation and multi-tenancy. With network virtualization, support for those properties is inherently provided.

Leveraging VXLAN technology, virtual networks are isolated from any other virtual networks as well as from the underlying physical infrastructure by default, delivering the security principle of least privilege. Virtual networks are created in isolation and remain isolated unless explicitly connected together. No physical subnets, VLANs, ACLs, or firewall rules are required to enable this isolation.



**Figure 60 – Network Isolation**

Any isolated virtual network can be made up of workloads distributed anywhere in the data center. Workloads in the same virtual network can reside on the same or separate hypervisors. Additionally, workloads in several multiple isolated virtual networks can reside on the same hypervisor. Isolation between virtual networks allows for overlapping IP addresses, making it possible to have isolated development, test, and production virtual networks, each with different application versions, but with the same IP addresses, all operating at the same time and on the same underlying physical infrastructure.

Virtual networks are also isolated from the underlying physical network. Because traffic between hypervisors is encapsulated, physical network devices operate in a completely different address space than the workloads connected to the virtual networks. A virtual network could support IPv6 application workloads on top of an IPv4 physical network. This isolation protects the underlying physical infrastructure from any possible attack initiated by workloads in any virtual network, independent of any VLANs, ACLs, or firewall rules that would traditionally be required to create this isolation.

#### 4.4.2 Network Segmentation

Related to isolation, but applied within a multi-tier virtual network, is segmentation. Traditionally, network segmentation is a function of a physical firewall or router, designed to allow or deny traffic between network segments or tiers. When segmenting traffic between web, application, and database tiers, traditional processes are time consuming and highly prone to human error, resulting in a large percentage of security breaches. Implementation requires deep and specific expertise in device configuration syntax, network addressing, and application ports and protocols.

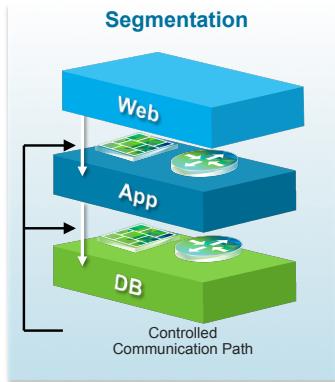


Figure 61 – Network Segmentation

Network segmentation, like isolation, is a core capability of VMware NSX network virtualization. A virtual network can support a multi-tier network environment. This allows for either multiple L2 segments with L3 segmentation or a single-tier network environment where workloads are all connected to a single L2 segment using distributed firewall rules. Both scenarios achieve the same goal of micro-segmenting the virtual network to offer workload-to-workload traffic protection, also referred to as east-west protection.

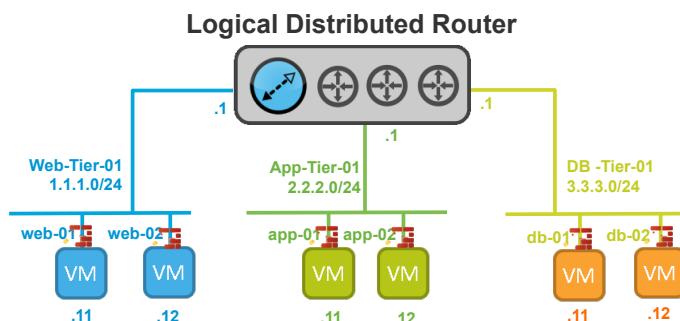


Figure 62 – Grouping Application Tiers

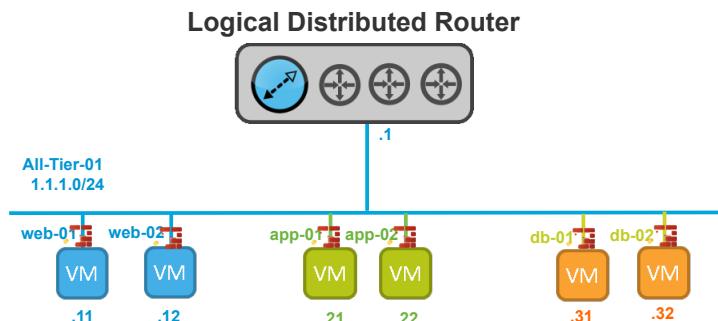


Figure 63 – Consolidating Application into Single Segment

Physical firewalls and access control lists traditionally deliver a proven segmentation function, trusted by network security teams and compliance

auditors. Confidence in this approach for cloud data centers has been shaken as more and more attacks, breaches, and downtime are attributed to human error in antiquated, manual network security provisioning and change management processes.

With network virtualization, network services (e.g., L2, L3, firewall) that are provisioned with a workload are programmatically created and distributed to the hypervisor vSwitch. Network services, including L3 segmentation and firewalling, are enforced at the virtual interface of the VM.

Communication within a virtual network never leaves the virtual environment, removing the requirement for network segmentation to be configured and maintained in the physical network or firewall.

#### 4.4.3 Taking Advantage of Abstraction

Traditionally, network security required the security team to have a deep understanding of network addressing, application ports, protocols, network hardware, workload location, and topology. Network virtualization abstracts application workload communication from the physical network hardware and topology, allowing network security to break free from physical constraints and apply network security based on user, application, and business context.

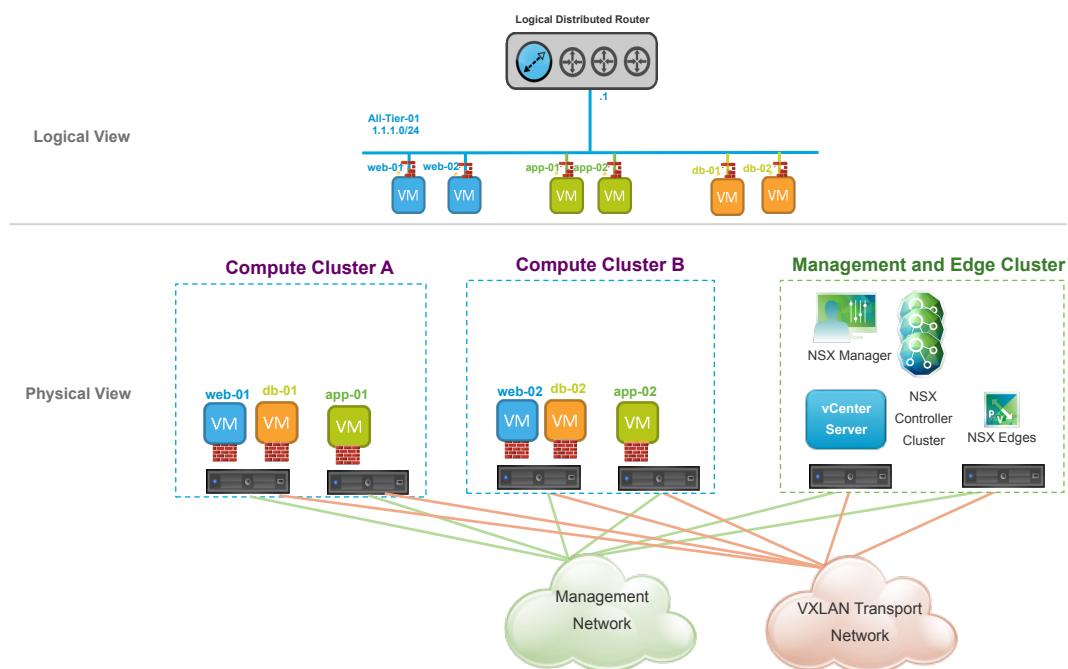
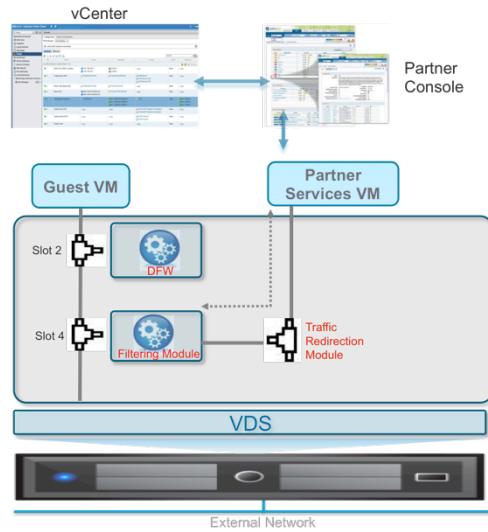


Figure 64 – Security Policies No More Tight to Physical Network Topology

#### 4.4.4 Advanced Security Service Insertion, Chaining and Steering

The NSX network virtualization platform provides L2-L4 stateful firewalling features to deliver segmentation within virtual networks. In some environments there is a requirement for more advanced network security capabilities. In these instances, customers can leverage VMware NSX to distribute, enable, and enforce advanced network security services in a virtualized network environment.

NSX distributes network services into the vNIC context to form a logical pipeline of services applied to virtual network traffic. Third party network services can be inserted into this logical pipeline, allowing physical or virtual services to be equivalently consumed.

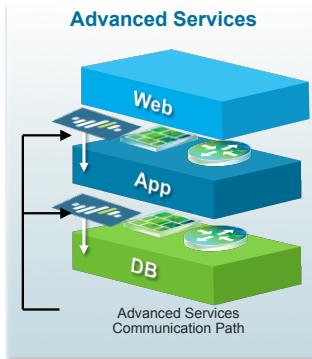


**Figure 65 – Service Insertion, Chaining, and Steering**

Between the guest VM and logical network (e.g., Logical Switch or DVS port-group VLAN-backed), there is a service space implemented into the vNIC context. Slot-ID materializes service connectivity to the VM. As depicted Figure 65, slot 2 is allocated to DFW, slot 4 to the specific third party advanced security services. Additional slots are available to plug in additional third-party services.

Traffic exiting the guest VM always follows the path with increasing slot-ID number, so a packet would first be redirected to slot 2 and then slot 4. Traffic reaching the guest VM follows the path in the reverse slot-ID order; first slot 4 and then slot 2.

Every security team uses a unique combination of network security products to address specific environmental needs. Where network security teams are often challenged to coordinate network security services from multiple vendors, VMware's entire ecosystem of security solution providers already leverages the NSX platform. Another powerful benefit of the centralize NSX approach is its ability to build policies that leverage service insertion, chaining, and steering to drive service execution in the logical services pipeline. This functionality is based on the result of other services, making it possible to coordinate otherwise completely unrelated network security services from multiple vendors.



**Figure 66 – Network Segmentation with Advanced Services Provided by Third Party Vendor.**

Integration with an advanced security services partner will leverage the VMware NSX platform to distribute the vendor's capability, making the advanced features locally available on each hypervisor. Network security policies, defined for applications workloads provisioned on or moved to that hypervisor, are inserted into the virtual network's logical pipeline. At runtime, the service insertion leverages the locally available advanced security service's feature set to deliver and enforce application, user, and context-based control policies at the workload's virtual interface.

#### 4.4.5 Consistent Visibility and Security Across Physical and Virtual

VMware NSX allows automated provisioning and context sharing across virtual and physical security platforms. Combined with traffic steering and policy enforcement at the vNIC, partner services that are traditionally deployed in a physical network environment are now easily provisioned and enforced in a virtual network environment. VMware NSX delivers customers a consistent model of visibility and security across applications residing on both physical or virtual workloads. Key motivations to implement VMware NSX include:

- **Enhance Existing Tools and Processes:** Dramatically increase provisioning speed, operational efficiency, and service quality while maintaining separation of duties between server, network, and security teams.
- **Control closer to the Application, without Downside:** This level of network security would traditionally have forced network and security teams to choose between performance and features. Leveraging the ability to distribute and enforce the advanced feature set at the applications virtual interface delivers the best of both.
- **Reduce Human Error in the Equation:** The infrastructure maintains policy, allowing workloads to be placed and moved anywhere in the data center without any manual intervention. Pre-approved application security policies can be applied programmatically, enabling self-service deployment of even complex network security services.

#### 4.4.6 Introduction to Service Composer

NSX introduces a mechanism for deploying security services independent of the underlying topology. Traditional services like firewall or advanced services like agentless AV, L7 firewall, IPS, and traffic monitoring can be deployed independent of the underlying physical or logical networking topologies. This enables a significant shift in planning and deploying services in the datacenter. Services no longer need to be tied down to networking topology. NSX provides a framework called Service Composer to enable deployment of security services for the datacenter.

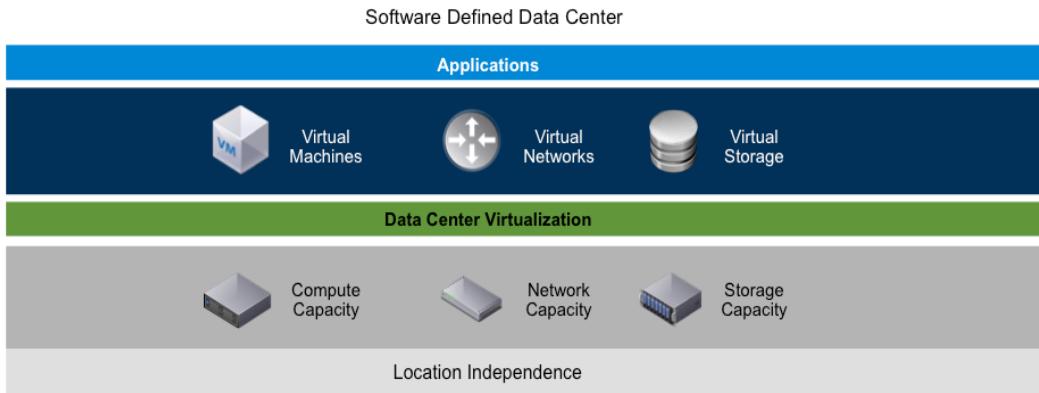
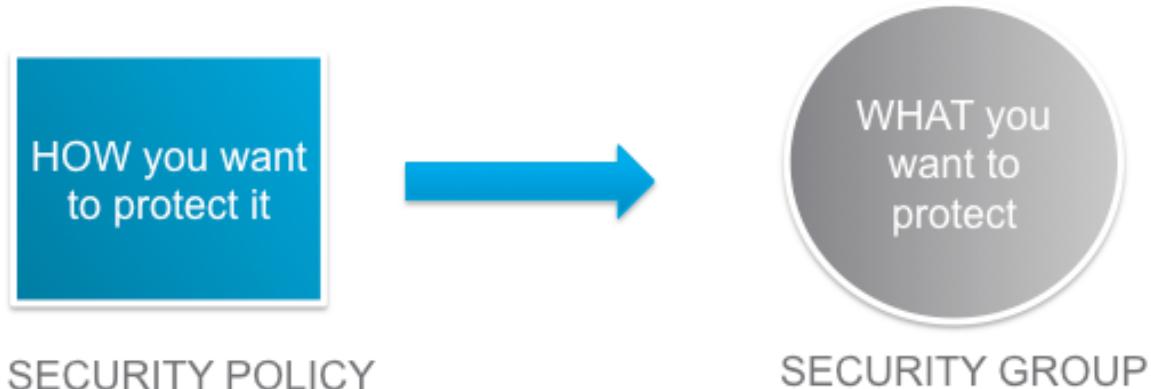


Figure 67 – Software Defined Data Center

Service Composer contains three broad parts:

- **Intelligent Grouping via Security Groups:** NSX achieves decoupling of workloads from the underlying topology via creation of these security groups.
- **3<sup>rd</sup> Party Service Registration and Deployment:** Enables 3<sup>rd</sup> party vendor registration with NSX and deployment of vendor security technologies throughout the datacenter
- **Security Policies:** A security policy allows flexibility in applying specific security rules to specified workloads. Policies include governance of not just the built-in NSX security services, but also redirection to third party services like Palo Alto Networks, CheckPoint and Fortinet once the 3<sup>rd</sup> party service is registered with NSX.

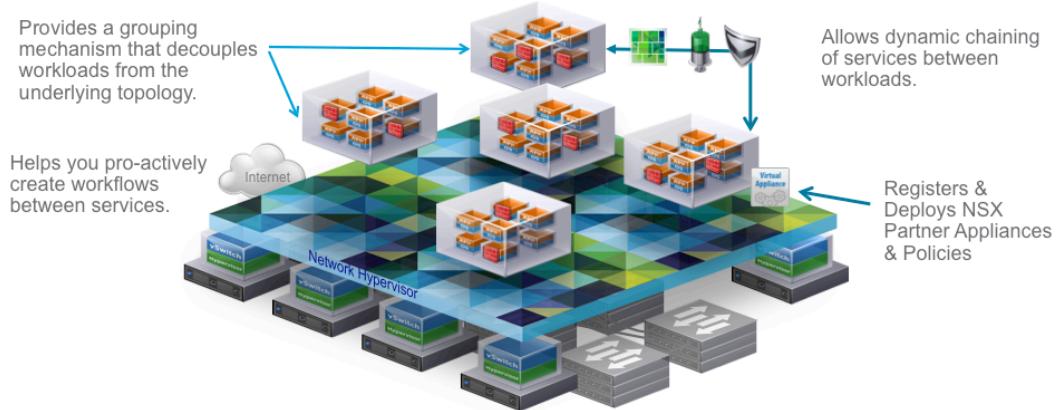
The simplicity of the Service Composer model is represented in the Figure 75.



**Figure 68 – Decoupling of Rules and Policy**

There are various advantages in decoupling the service and rule creation from the underlying topologies:

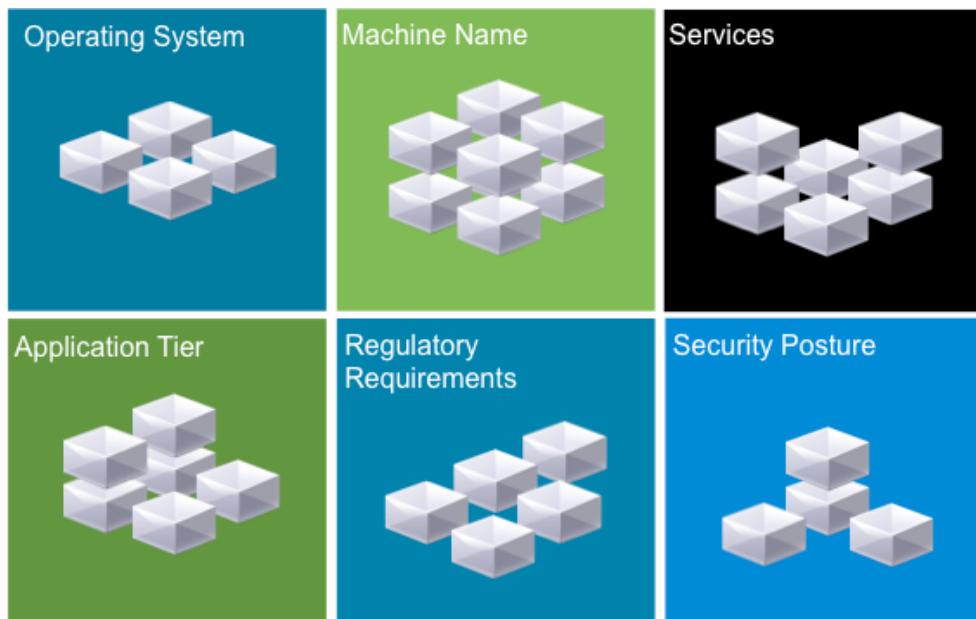
- **Distribution of Services:** The services layer of NSX allows distribution and embedding of services across the datacenter, enabling workload mobility across the datacenter without bottlenecks or hair pinning of traffic. In this model, granular traffic inspection is done for all workloads wherever they reside in the datacenter.
- **Policies are Workload-Centric:** Policies can now be truly workload centric, rather than translating from workloads to virtual machines to basic networking topology and IP address constructs. Policies can be now configured to define group of database workloads that will be allowed specific operations without explicitly calling out networking centric language (e.g., IP subnets, MACs, ports).
- **Truly Agile and Adaptive Security Controls:** Workloads are freed from design constraints based on the underlying physical networking topologies. Logical networking topologies can be created at scale, on demand, and provisioned with security controls that are independent of these topologies. When workloads migrate, security controls and policies migrate with them. New workloads do not require the recreation of security policies; these policies are automatically applied to the workloads.
- **Service Chaining is Policy-based and Vendor Independent:** With NSX, service chaining is based on a policy across various security controls. This has evolved from manually hardwiring various security controls from multiple vendors in the underlying network. With NSX, policies can be created, modified, and deleted based on the individual requirements.



**Figure 69 – Advantages of Service Composer**

#### 4.4.6.1 Introduction to Intelligent Grouping

Intelligent grouping in NSX can be created in multiple ways, with a variety of customized grouping criteria possible as shown in the Figure 70.



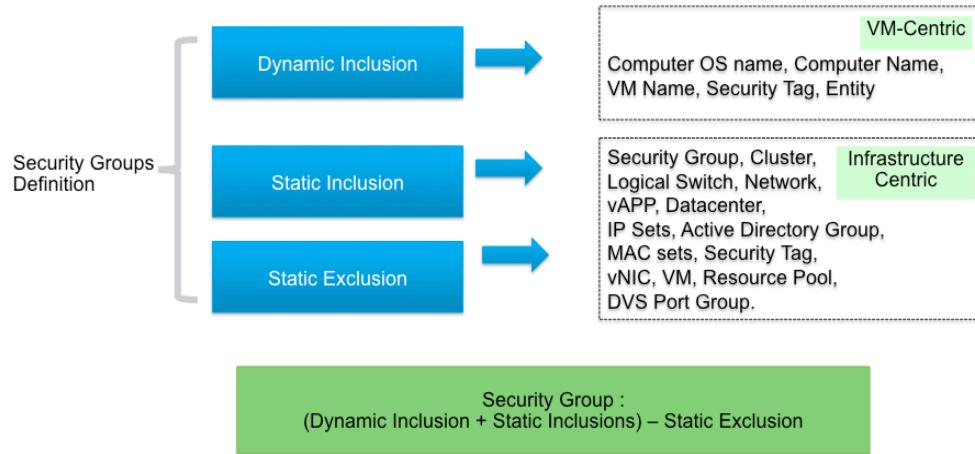
**Figure 70 – Types of intelligent Grouping**

NSX provides grouping mechanism criteria can include any of the following:

- vCenter Objects: VMs, Distributed Switches, Clusters, etc.
- VM Properties: vNICs, VM names, VM operating Systems, etc.
- NSX Objects: Logical Switches, Security Tags, Logical Routers, etc.

Grouping mechanisms can be either static or dynamic in nature, and a group can be any combination of objects. Grouping criteria can include any combination of vCenter objects, NSX Objects, VM Properties, or Identity Manager objects (e.g.,

AD Groups). A security group in NSX is based on all static and dynamic criteria along with static exclusion criteria defined by a user. Figure 71 details the security group construct and valid group objects.



**Figure 71 – Scope of Security of Group Attributes**

Static grouping mechanisms are based on a collection of virtual machines that conform to set criteria. vCenter objects define datacenter components and NSX objects define core networking components which are used in static grouping criteria. Combining different objects for the grouping criteria results in the creation of the “AND” expression in the NSX system.

Dynamic grouping mechanisms are more flexible and are characterized by allowing expressions that evaluate to defining the virtual machines for a group. The core difference is the ability to define “AND/OR” as well as “ANY/ALL” criteria for the grouping.

Evaluation of VMs that are part of a group is also different. In a static grouping mechanism, the criteria instruct the NSX manager which objects to look for in terms of change. In dynamic grouping, NSX manager evaluates each and every change in the datacenter environment to determine if this affects any group. Thus, dynamic grouping is more impactful to NSX manager than static grouping. To avoid the global evaluation of dynamic objects, the grouping criteria should be done by zone or application and mapped into sections. In this way, when updates or evaluations have to be performed, only the limited set of objects belonging to particular section are updated and propagated to specific hosts and vNIC. This ensures rule changes only require publishing of the section versus the entire rule set. This method has a critical impact on performance of NSX manager and the propagation time of policy to hosts. The sample sectionalized DFW rules set is shown in Figure 72.

No.	Name	Source	Destination	Service	Action	Applied To
▼ Organization-A Section (Rule 1)						
✓ 1	Project-X-01	ProjectX-IPSet-01	SG-Web-ProjectX	SSH	Allow	SG-Web-ProjectX
► Organization-B Section (Rule 2)						
► Organization-C Section (Rule 3)						
► Organization-D Section (Rule 4)						
► Default Section Layer3 (Rule 5 - 7)						

Figure 72 – Creating Sections for Efficient Rule Processing and Propagation

## Intelligent Grouping – An Example of Efficiency

In the following example, virtual machines are being added to a logical switch. In a traditional firewall scenario, IP addresses or subnets must be added or updated to provide adequate security to the workload. This implies that the addition of firewall rules is a prerequisite of provisioning a workload. This is depicted in the Figure 73.

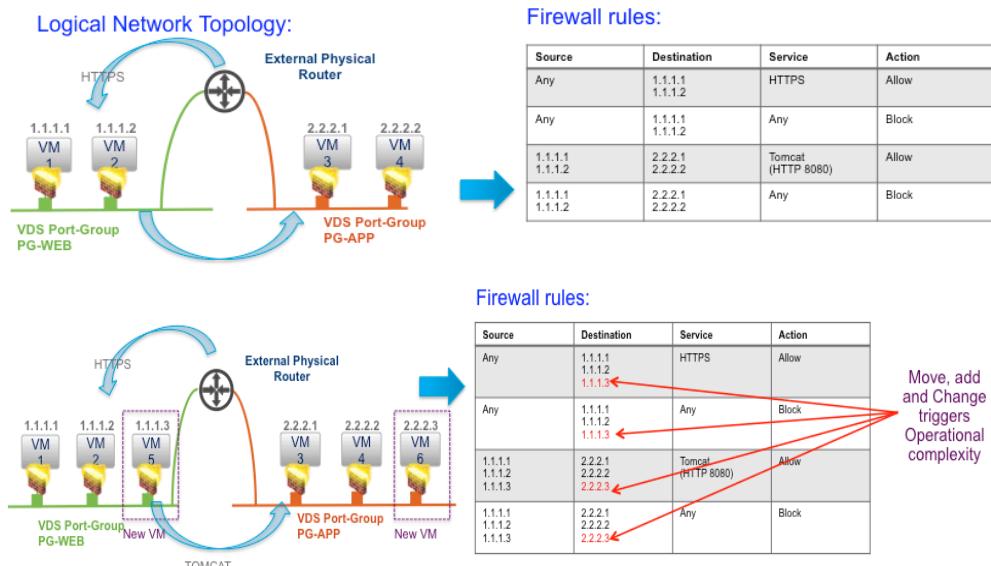


Figure 73 – Overhead of Rule Management

With NSX, a security or network administrator can define that any workload on this virtual machine will be assigned similar security controls. With intelligent grouping based on the NSX Logical Switch, a rule can be defined that does not change with every virtual machine provisioned. The security controls adapt to an expanding datacenter as shown in Figure 74.

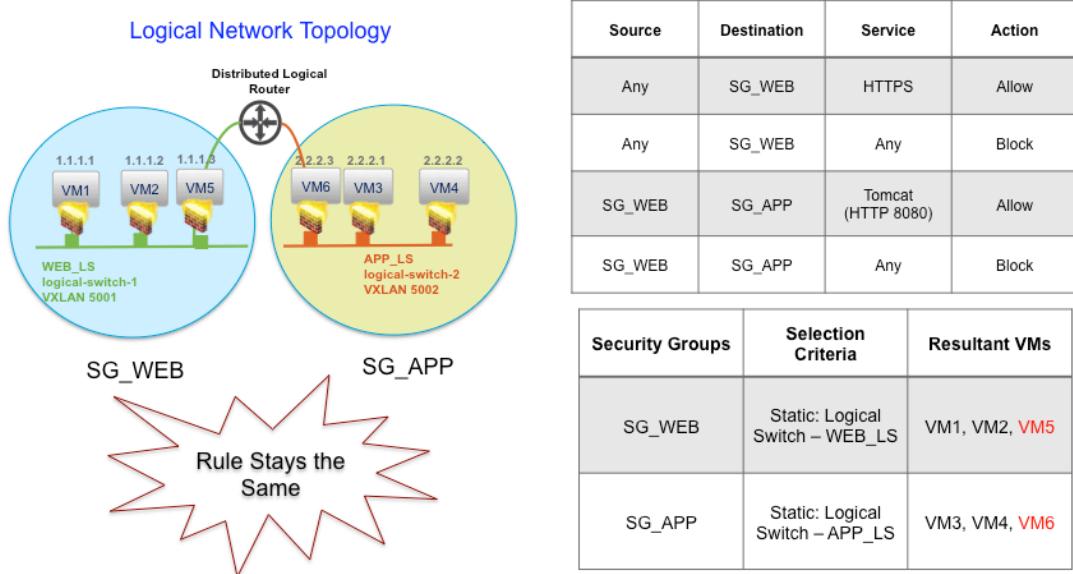


Figure 74 – Advantage of Object Based Rule Management

## Security Tags

NSX provides security tags that can be applied to any virtual machine. This allows for classification of virtual machines in any desirable form.

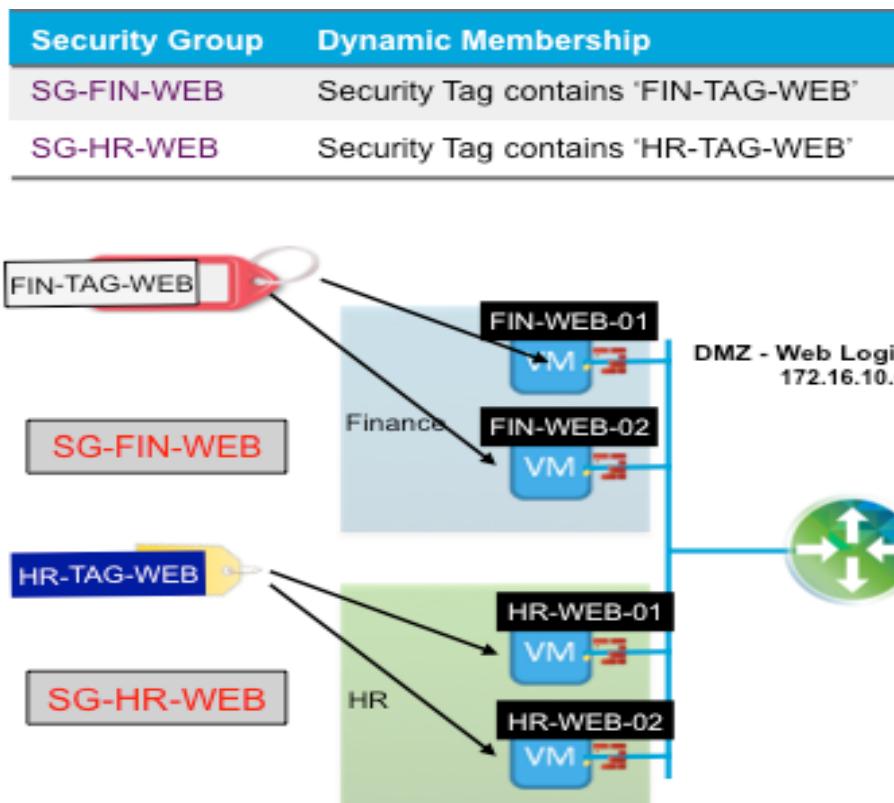


Figure 75 – Tag Used as Base Isolation Method

Some of the most common forms of classification for using security tags are:

- Security State (i.e., vulnerability identified)
- Classification by department
- Data-type classification (e.g., PCI Data).
- Type of environment (e.g., production, devops)
- VM geo-location

Security tags are intended for providing more contextual information about the workload, allowing for a better overall security. In addition to users creating and defining their own tags, third party security vendors can use the same tag for advance workload actions. Examples of vendors set tags include trigger on malware found, suspicious workload activity, CVE score compliance. This functionality allows for context sharing across different vendors.

#### 4.4.6.2 Introduction to Security Policy

NSX provides security policies as a way to group rules for security controls that will be applied to an intelligent group created in the datacenter.



Figure 76 – What goes into Security Policy and Security Group

A security policy is comprised of two broad parts:

- **Services:** The security controls that will be provisioned for a policy. Examples of services are firewall, antivirus, vulnerability management, and IPS.
- **Profiles:** Security vendors publish policies that are shared with the NSX platform. NSX defines these as service profiles for a particular service.

Security Policies can be written in two ways:

- **Traditional Method:** Rules and policies written in the traditional firewall table format. This is available both for the built-in distributed firewall as well as third party advanced services for network introspection such as IPS and traffic monitoring.
- **NSX Policy Method:** The service composer method provides a construct called security policy that enables creation rules and controls in a new manner. These policies can be created as a template and applied to as many security groups as needed. Security policies allow usage of security

groups as method of creating rules, decoupling the rule management from the context.

NSX enables creation of sections in a firewall rule table. Sections allow better management and grouping of firewall rules. A single security policy is essentially a section in a firewall rule table. This maintains synchronization between rules in a firewall rule table and those written via the security policy, ensuring consistent implementation.

As security policies are written for a specific applications or workloads, these rules are organized into specific sections in a firewall rule table. Additionally, multiple security policies can be applied to a single application. In that scenario, the order of the sections is important to determine that precedence of rule application.

#### 4.4.6.3 Anatomy of a Security Policy

NSX security policies consist of rules, weights, and inheritance specifications; their anatomy is further detailed in Figure 77.

Rules	Weights	Inheritance
<ul style="list-style-type: none"><li>Rules for various services available in the platform.</li><li>Rules have precedence.</li></ul>	<ul style="list-style-type: none"><li>Determines the policy that needs to be applied first.</li><li>Rank and Weight of the policy are the same.</li></ul>	<ul style="list-style-type: none"><li>If it contains rules from a parent policy.</li><li>Parent and Child policy weights are automatically adjusted.</li></ul>

Figure 77 –Anatomy Security Policy

Each security policy contains the following:

- **Rules:** Each policy contains a set of rules that define the security control behavior. A single policy may contain one or more security controls as shown in Figure 77. There are three types of security controls allowed in security policy rules:
  - **Firewall:** NSX built in distributed firewall.
  - **Guest Introspection Control:** VM-based controls including anti-virus, anti-malware, file integrity monitoring, and data security controls.
  - **Network Introspection Controls:** L7 firewall, IDS/IPS.

Vendors integrate with NSX using the guest introspection framework and/or the network introspection framework to provide the requisite controls for those technologies. Any policy created on the vendor platform is published on the NSX platform. These policies are termed service profiles in NSX.

For each rule that contains a security control from a vendor, a service profile must be associated with the rule.

Rules within a given security control can have precedence. Rules will be executed in a top-down order.

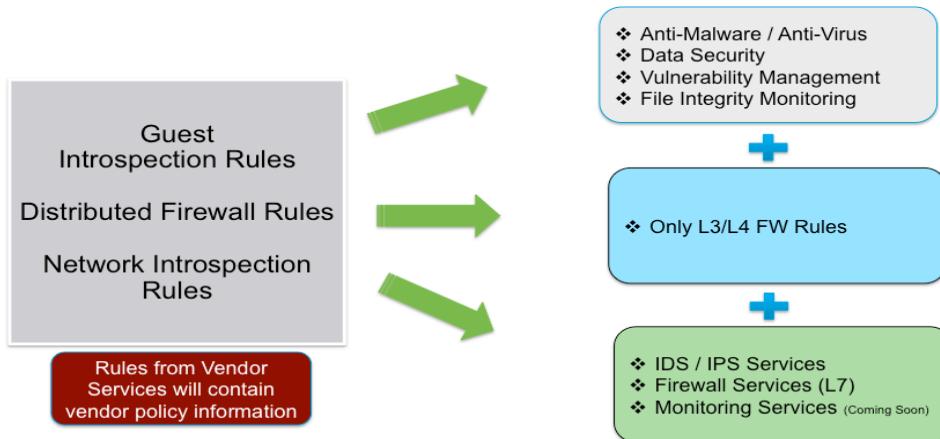


Figure 78 – Types of Rules and Properties

- **Weight:** A given security group can have multiple security policies applied to it. The weight of a policy provides NSX with the information about the order of rules to be applied. Weight of a policy determines the rank of the policy vis-à-vis other policies in the NSX eco-system. Higher weight rules are executed first.
- **Inheritance:** A policy can be inherited from multiple security policies. NSX provides a mechanism to create base policies and child policies. Applying a child policy to a security group automatically applies the base/parent policies to the security group. Security policies can be inherited from base policies, then augmented with additional rules. A child security policy will contain all the rules in the order created in the base policy; additional child rules will be added after those. Applying a child policy to a security group will automatically apply the parent policy.

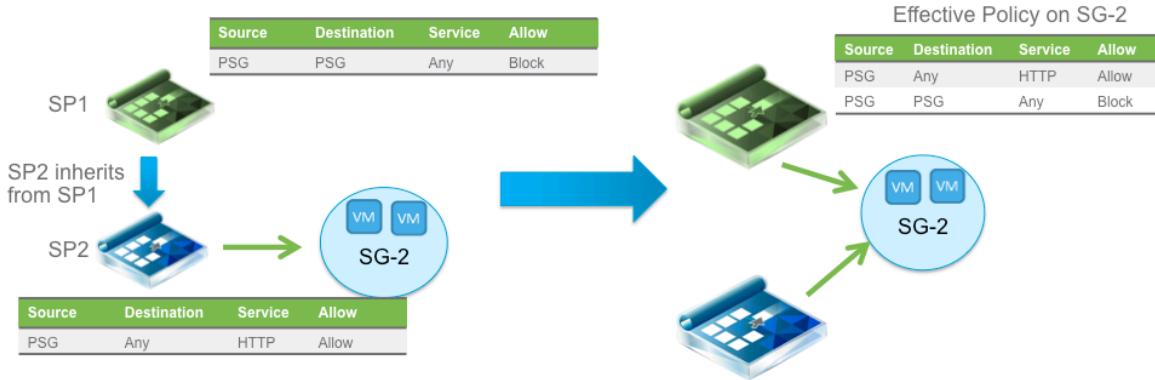


Figure 79 – Rule Inheritance

#### 4.4.7 Micro-Segmentation with NSX DFW and Implementation

DFW can be leveraged to implement micro-segmentation for a 3-tier application (e.g., web/app/database) where multiple organizations (e.g., finance, HR) share the same logical network topology.

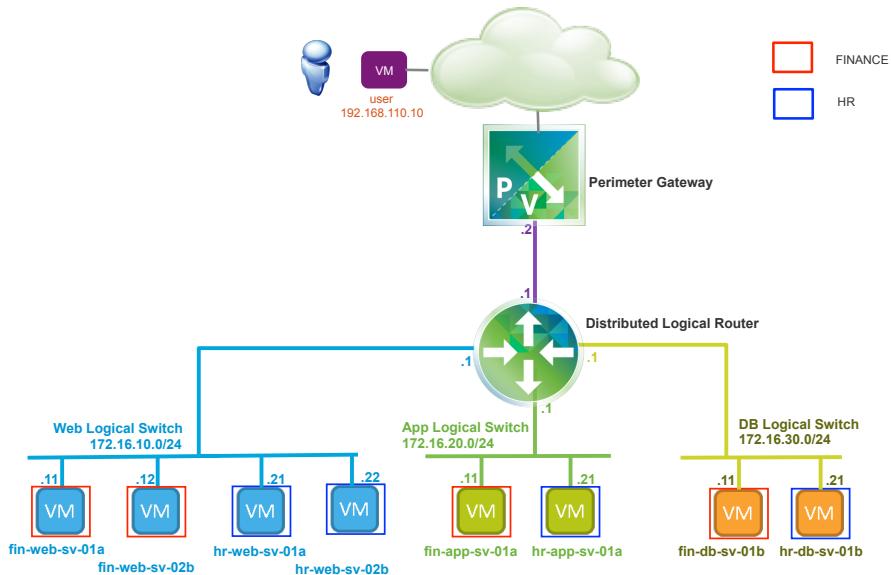


Figure 80 – Micro-Segmentation Use Case

This network design contains 3 logical switches (web LS, app LS, and db LS) interconnected through a Distributed Logical Router. IP addresses defined on DLR are the default gateway for VM connected to any of the Logical Switches.

DLR is connected to an Edge services gateway that provides connectivity to the physical world. An end user connected to the external network can reach the web Logical Switch subnet where a dynamic routing protocol enabled on the DLR and the ESG to announce web LS subnet.

In this example, the finance and HR organization each have 2 web servers, 1 app server and 1 database server.

In this micro-segmentation design, servers of same role are connected to same Logical Switch, irrespective of the organization they belong to. HR and Finance are used as examples; it is possible to find many more organizations to host on this logical network infrastructure.

Using DFW, it is possible to achieve the same level of security if Finance and HR workloads are connected to different logical networks. The power of DFW resides in the fact that the network topology is no longer a barrier to security enforcement; the same level of traffic access control can be achieved for any type of topology.

To group VMs of same role and of same organization together, leverage the Service Composer functionality.

One property within Service Composer is the Security Group (SG) construct. SG allows for dynamically or statically including objects in a container. This container will then be used as the source or destination field of our DFW security policy rule.

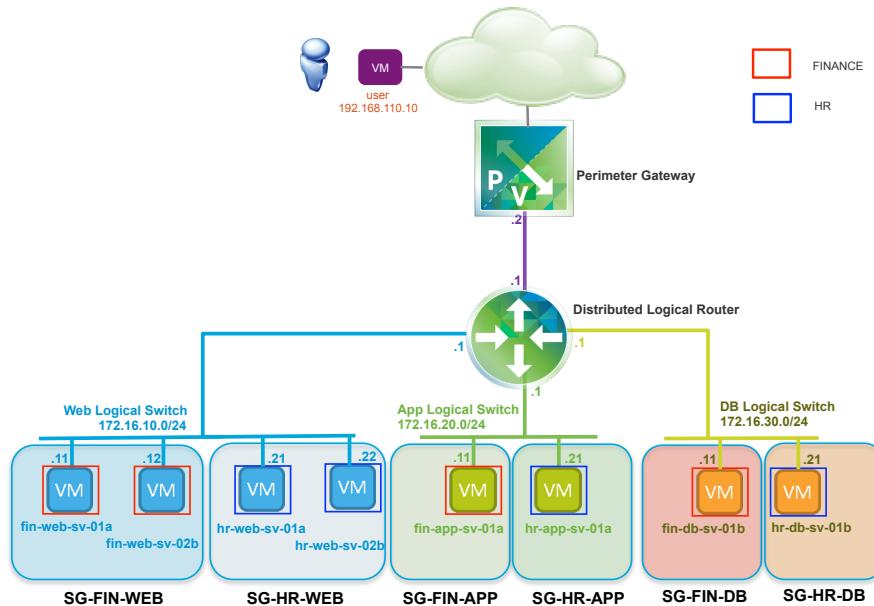


Figure 81 – Grouping VM into Service Composer/Security Groups

As shown in the Figure 81, web servers from the finance organization are grouped into a security group called SG-FIN-WEB. To build this SG, use dynamic inclusion based on VM name or security tag. A VM name containing 'fin-web' syntax will be automatically included in SG-FIN-WEB or a VM assigned with the security tag 'FIN-WEB-TAG' will be automatically added to SG-FIN-WEB. Where static inclusion is preferred, fin-web-sv-01a and fin-web-sv-02b can be selected and added into SG-FIN-WEB manually.

This design contains a total of 6 Security Groups: 3 for finance and 3 for HR with 1 SG per tier.

Grouping VMs into the right container is the foundation to implementing micro-segmentation. Once done, it is then easy to implement traffic policy as shown in Figure 82.

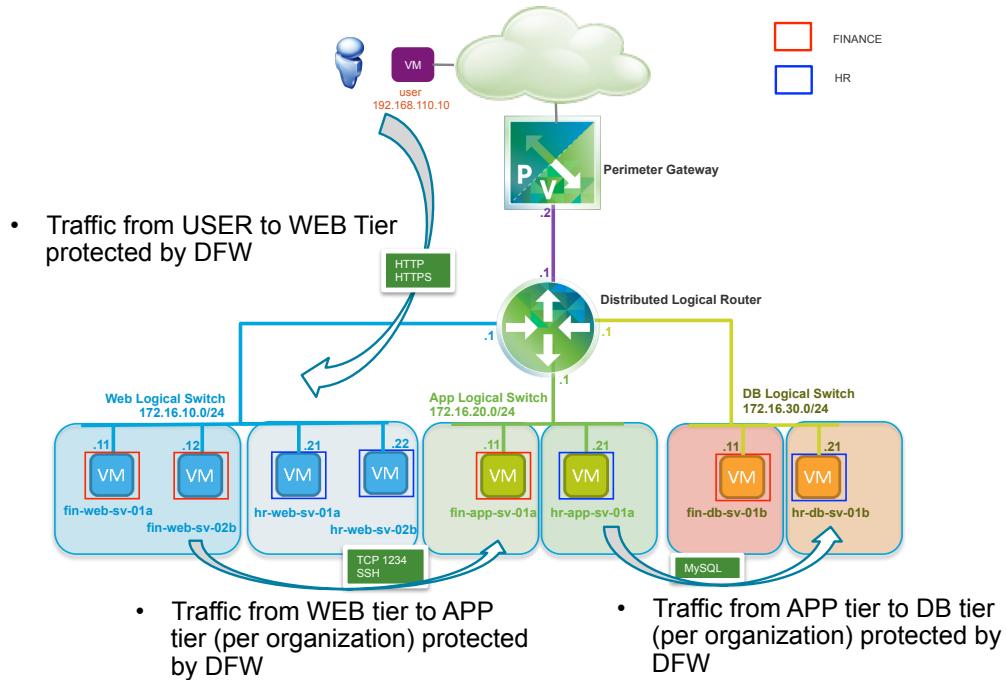


Figure 82 – Inter-Tier Network Traffic Security Policy

Inter-Tier network traffic security policy is enforced as following:

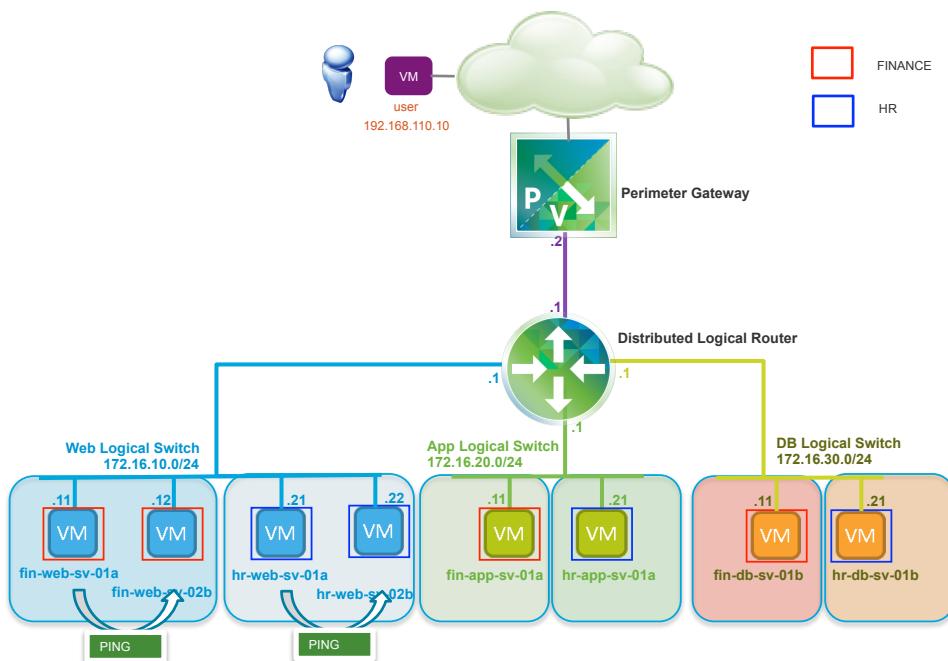
1. From INTERNET to web servers, allow HTTP and HTTPS. Other traffic is discarded.
2. From web tier to app tier, allow TCP 1234 and SSH. Other traffic is discarded.
3. From app tier to DB tier, allow MySQL. Other traffic is discarded.

---

Note that finance cannot communicate with HR for inter-tier network traffic; communication between the 2 organizations across tiers is completely prohibited.

---

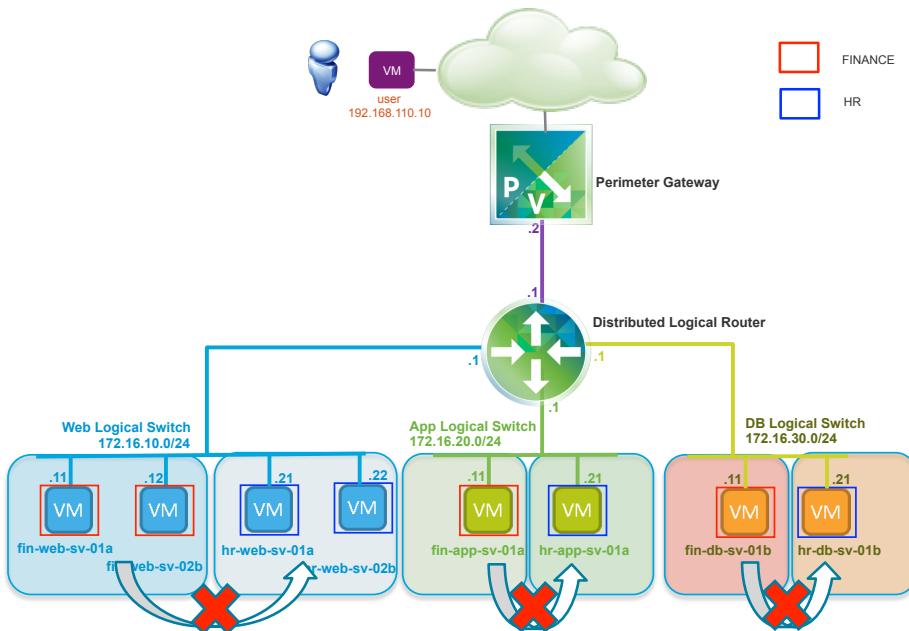
Intra-Tier network traffic security policy is enforced where servers of the same organization can ping each other when belonging to the same tier. As an example, fin-web-sv-01a can ping fin-web-sv-02b. However, fin-web-sv-01a cannot ping hr-web-sv-01a or hr-web-sv-02b.



- Intra-Tier traffic permitted for PING within each organization

**Figure 83 – Intra-Tier Network Traffic Security Policy (Same Organization)**

Communication between workloads from different organizations within the same tier should simply be forbidden, as shown in Figure 84.



- Intra-Tier traffic blocked across organization

**Figure 85 – Intra-Tier Network Traffic Security Policy (Across Organizations)**

The first step in implementing this micro-segmentation use case is to modify the DFW default policy rule from “Action = Allow” to “Action = Block”. This creates a positive security model where only explicitly allowed traffic flows are defined in the security rules table; all other communication is denied.

Name	Source	Destination	Service	Action
Default rule	ANY	ANY	Any	Block

Figure 86 - DFW Default Policy Rule

In deployments where the vCenter Server is hosted on an ESXi hypervisor that has been prepared for VXLAN, the DFW default policy rule would apply also to the vCenter’s vNIC. In order to avoid losing connectivity to vCenter, it is recommended to put vCenter into a DFW exclusion list to avoid enforcing DFW policies for that specific virtual machine. Alternatively, it is possible to create a dedicated DFW rule to specify the IP subnets allowed to communicate with vCenter. The same considerations hold true for other management products (e.g., vRA, vCOps), as they will each receive one instance of DFW per vNIC. Only NSX manager and NSX controllers are automatically excluded from DFW policy enforcement.

The security policy rules governing inter-tier traffic can thus be defined as shown in Figure 86.

Name	Source	Destination	Service	Action
<b>FIN</b> – INTERNET to WEB Tier	ANY	SG-FIN-WEB	HTTP HTTPS	Allow
<b>HR</b> – INTERNET to WEB Tier	ANY	SG-HR-WEB	HTTP HTTPS	Allow
<b>FIN</b> – WEB Tier to APP Tier	SG-FIN-WEB	SG-FIN-APP	TCP-1234 SSH	Allow
<b>HR</b> – WEB Tier to APP Tier	SG-HR-WEB	SG-HR-APP	TCP-1234 SSH	Allow
<b>FIN</b> – APP Tier to DB Tier	SG-FIN-APP	SG-FIN-DB	MySQL	Allow
<b>HR</b> – APP Tier to DB Tier	SG-HR-APP	SG-HR-DB	MySQL	Allow

Figure 87 - DFW Inter-Tiers Policy Rules

Finally, for intra-tier communication, this example implements the security policy rules shown in Figure 87.

Name	Source	Destination	Service	Action
<b>FIN</b> – WEB Tier to WEB Tier	SG-FIN-WEB	SG-FIN-WEB	ICMP echo ICMP echo reply	Allow
<b>HR</b> – WEB Tier to WEB Tier	SG-HR-WEB	SG-HR-WEB	ICMP echo ICMP echo reply	Allow
<b>FIN</b> – APP Tier to APP Tier	SG-FIN-APP	SG-FIN-APP	ICMP echo ICMP echo reply	Allow
<b>HR</b> – APP Tier to APP Tier	SG-HR-APP	SG-HR-APP	ICMP echo ICMP echo reply	Allow
<b>FIN</b> – DB Tier to DB Tier	SG-FIN-DB	SG-FIN-DB	ICMP echo ICMP echo reply	Allow
<b>HR</b> – DB Tier to DB Tier	SG-HR-DB	SG-HR-DB	ICMP echo ICMP echo reply	Allow

Figure 88 - DFW Intra-Tier Policy Rules

## 4.5 Logical Load Balancing

Load balancing is another network service available within NSX that can be natively enabled on the NSX Edge device. The two main drivers for deploying a load balancer are scaling out an application through distribution of workload across multiple servers and improving its high-availability characteristics.

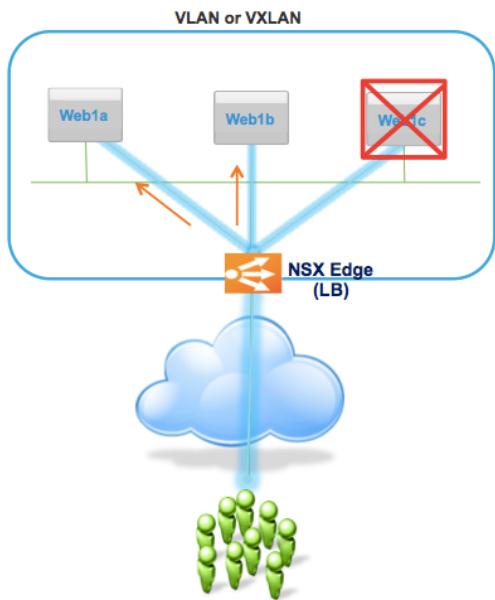


Figure 89 – NSX Load Balancing

The NSX load balancing service is specially designed for IT automating IT and dev-ops style deployment and thus to be fully programmable via API while utilizing the same central point of management and monitoring as other NSX network services.

The feature and functionalities are comprehensive enough to satisfy the needs of the majority of the application deployments. This functionality includes:

- The Edge VM active-standby mode provides high availability for load-balancer
- Support for any TCP application, including LDAP, FTP, HTTP, and HTTPS.
- Support for UDP application as of NSX release 6.1.
- Multiple load balancing distribution algorithms: round-robin, least connections, source IP hash, and URI.
- Multiple health checks: TCP, HTTP, and HTTPS including content inspection.
- Persistence: Source IP, MSRDP, cookie, and SSL session-id.
- Connection throttling of maximum connections and connections per second.
- L7 manipulation, including URL block, URL rewrite, and content rewrite.

- Optimization through support of SSL offload.
- The NSX platform can also integrate load-balancing services offered by 3rd party vendors.
- The NSX Edge offers support for two deployment models: one-arm or proxy mode, and inline or transparent mode.

### Proxy Mode (One-Arm)

Proxy mode configuration is highlighted in Figure 90 and consists of deploying an NSX Edge directly connected to the logical network where load-balancing services are required.

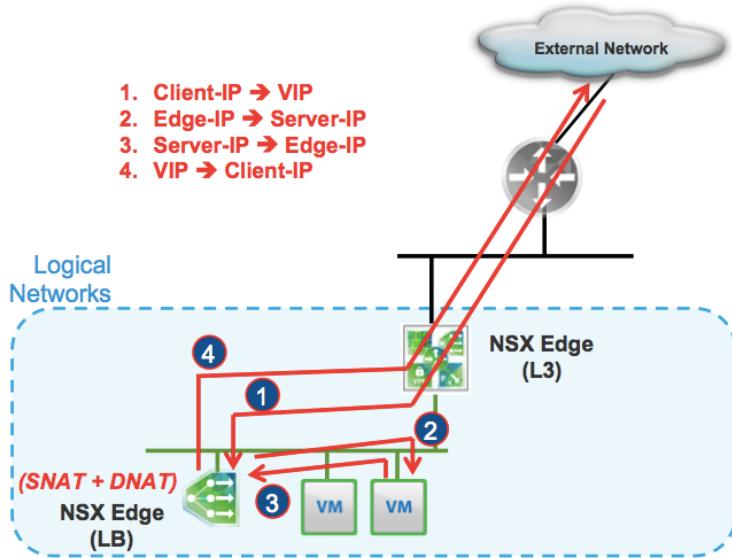


Figure 90 - One-Arm Mode Load Balancing Services

In this configuration mode:

1. The external client sends traffic to the Virtual IP address (VIP) exposed by the load balancer.
2. The load balancer performs two address translations on the original packets received from the client: destination NAT (D-NAT) to replace the VIP with the IP address of one of the servers deployed in the server farm, and source NAT (S-NAT) to replace the client IP address with the IP address identifying the load-balancer itself. S-NAT is required to force through the load balancer the return traffic from the server farm to the client.
3. The server in the server farm replies by sending the traffic to the load balancer per S-NAT functionality.
4. The load balancer again performs a source and destination NAT service to send traffic to the external client, leveraging its VIP as source IP address.

This model is simpler to deploy and provides greater flexibility than traditional load balancers. It allows deployment of load balancer services (e.g., NSX Edge

appliances) directly on the logical segments without requiring any modification on the centralized NSX Edge providing routing communication to the physical network. On the downside, this option requires provisioning more NSX Edge instances and mandates the deployment of source NAT that does not allow the servers in the datacenter to have visibility into the original client IP address.

The load balancer can insert the original IP address of the client into the HTTP header before performing S-NAT – a function named "Insert X-Forwarded-For HTTP header". This provides the servers visibility into the client IP address, and is limited to HTTP traffic.

### Transparent Mode (Inline)

Inline mode deploys the NSX Edge inline to the traffic destined to the server farm. Figure 91 shows the topology for this configuration.

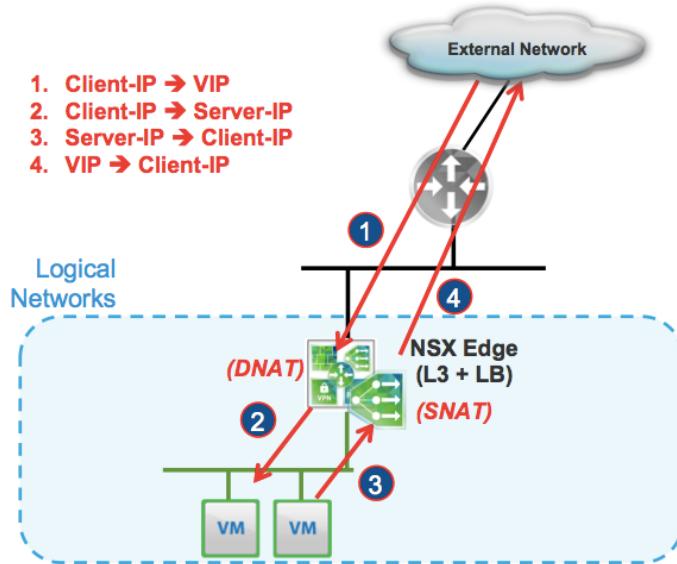


Figure 91 – Inline Mode Load Balancing Services

Transparent mode traffic flow is processed as follows:

1. The external client sends traffic to the virtual IP address (VIP) exposed by the load balancer.
2. The load balancer – a centralized NSX Edge – performs only destination NAT (D-NAT) to replace the VIP with the IP address of one of the servers deployed in the server farm.
3. The server in the server farm replies to the original client IP address. The traffic is received again by the load balancer since it is deployed inline, usually as the default gateway for the server farm.
4. The load balancer performs source NAT to send traffic to the external client, leveraging its VIP as source IP address.

This deployment model is also quite simple, and additionally provides the servers full visibility into the original client IP address. It is less flexible from a design perspective as it usually forces the load balancer to serve as default gateway for

the logical segments where the server farms are deployed. This implies that only centralized, rather than distributed, routing must be adopted for those segments. Additionally, in this case load balancing is another logical service added to the NSX Edge which is already providing routing services between the logical and the physical networks. Thus it is recommended to increase the form factor of the NSX Edge to X-Large before enabling load-balancing services.

Best case scalability and throughput numbers for load balancing on a single NSX Edge are:

- Throughput: 9 Gbps
- Concurrent connections: 1 million
- New connections per second: 131k

Figure 92 details deployment examples of tenants with different applications and load balancing requirements. Each of these applications is hosted in the same cloud with the network services offered by NSX.

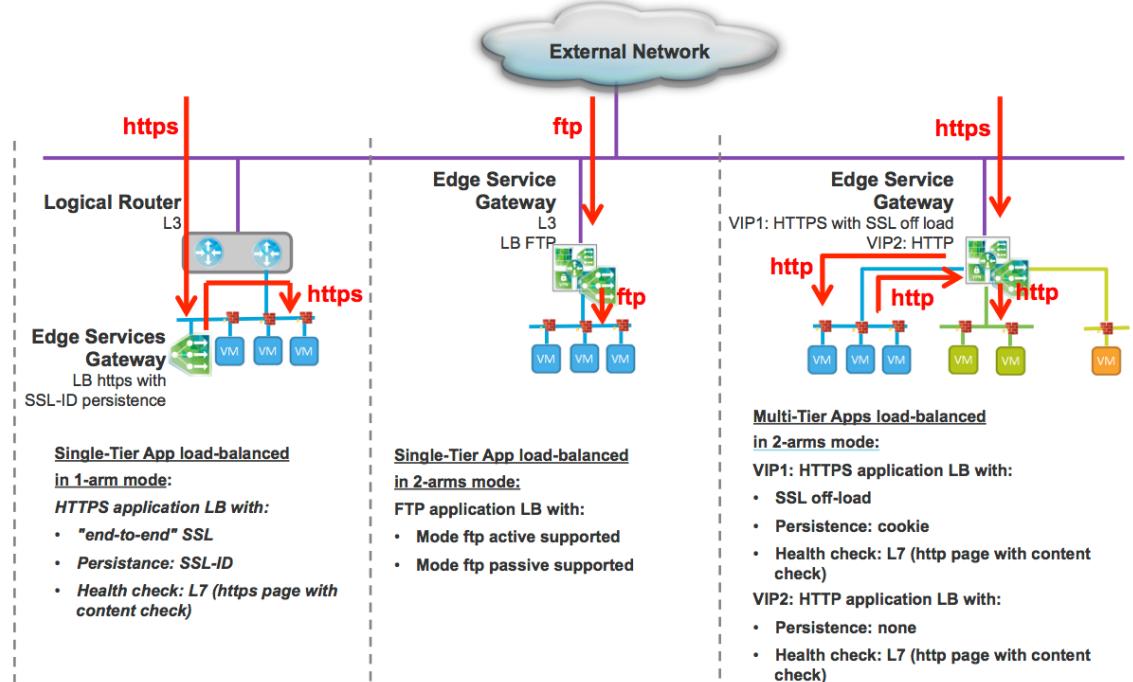


Figure 92 - Deployment Examples of NSX Load Balancing

The design criteria that are can be utilized in deploying load-balancer:

- In a traditional enterprise topology, it is desirable to deploy load-balancer near application segments. This is a proxy-mode configuration allowing scaling per application segments or workload as well allowing centralized Edge VM to be in ECMP mode for the bandwidth scaling.
- The load balancing service can be fully distributed across tenants. This brings multiple benefits:

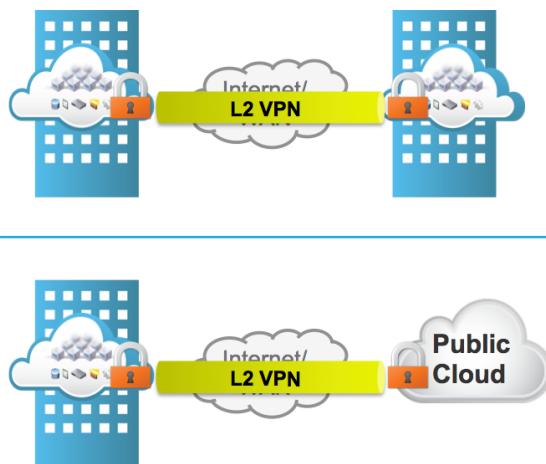
- Each tenant can have an independent load balancer, either in-line or proxy mode
- Each tenant's configuration change does not impact other tenants.
- Each tenant load balancing service can scale up to the defined limits
- The same tenant can mix its load balancing service with other network services such as routing, firewalling, and VPN

## 4.6 Virtual Private Network (VPN) Services

The final two NSX logical network functionalities described in this paper are Virtual Private Network (VPN) services. Those are categorized as L2 VPN and L3 VPN.

### 4.6.1 L2 VPN

The deployment of an L2 VPN service allows extension of L2 connectivity between two data center locations.

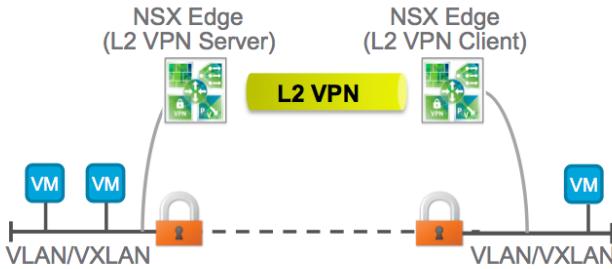


**Figure 93 - Use Cases for NSX L2 VPN**

There are several use cases that can benefit from this functionality, both for enterprise and SP deployments:

- Enterprise workload migration/datacenter consolidation
- Service provider tenant on-boarding
- Cloud bursting/hybrid cloud)
- Stretched application tiers/hybrid cloud

Figure 94 highlights the creation of a L2 VPN tunnel between a pair of NSX Edge devices deployed in separate datacenter sites.



**Figure 94 – NSX Edges for L2 VPN Services**

Some considerations for this specific deployment include:

- The L2 VPN connection is an SSL tunnel connecting separate networks in each location. The connected networks offer connectivity to the same address space (e.g., IP subnet). This is the characteristic that makes this a L2 VPN service.
- The local networks can be of any nature, VLAN or VXLAN. Additionally, the L2 VPN service can interconnect networks of different nature – VLAN at one site, VXLAN at the other site.
- L2 VPN a point-to-point service that can be established between two locations. The NSX Edge deployed in one datacenter site takes the role of the L2 VPN server while the NSX Edge at the second site is the L2 VPN client initiating the connection to the server.
- The NSX L2 VPN is usually deployed across a network infrastructure interconnecting the sites, provided by a service provider or owned by the enterprise. No specific requirements placed on the network in terms of latency, bandwidth, or MTU. The NSX L2 VPN service is designed to work across any quality of network connection.

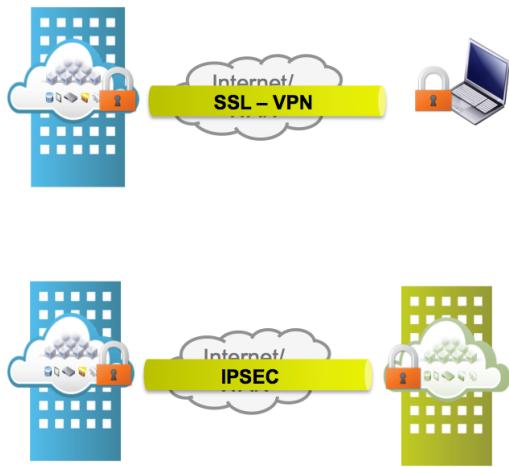
The NSX 6.1 software further enhances the L2 VPN solution. Significant improvements include:

- With 6.0 releases, it is required to deploy two independent NSX domains between sites connected via L2 VPN. This implies the deployment of separate vCenter, NSX manager and controller clusters at each location. This may not be desirable in case of hybrid cloud and disaster recovery solution where all is required a simple L2 VPN connectivity. From NSX 6.1 software release onward, it is allowed for a remote NSX Edge deployment (functioning as L2 VPN client) without the requirement of full blown NSX at the remote site
- NSX 6.1 introduces the trunk interface on the NSX Edge, expanding on the uplink and internal interface capabilities. Leveraging trunks, it is possible to extend L2 connectivity between multiple networks (e.g., VLAN or VXLAN backed port-groups) deployed at each site. In 6.0 this was limited to one VLAN/VXLAN per NSX Edge.

- Full HA support for NSX Edge deployed as L2 VPN server or client is introduced from 6.1. A pair of NSX Edges working in active/standby can be deployed in each site.

#### 4.6.2 L3 VPN

Finally, L3 VPN services are used to provide secure L3 connectivity into the datacenter network from remote locations.



**Figure 95 - NSX L3 VPN Services**

As highlighted in Figure 95, the L3 VPN services can be used by remote clients leveraging SSL tunnels to securely connect to private networks behind a NSX Edge gateway acting as L3 VPN Server in the datacenter. This service is usually referred to as SSL VPN-Plus.

Alternatively, the NSX Edge can be deployed to use standard IPSec protocol settings to interoperate with all major physical VPN vendors' equipment and establish site-to-site secure L3 connections. It is possible to connect multiple remote IP subnets to the internal network behind the NSX Edge. Differently from the previously described L2 VPN service, connectivity in this case is routed since the remote and local subnets are part of different address spaces. In the current implementation, unicast-only communication is supported across the IPSec connection; thus dynamic routing protocols cannot be used and a static routing configuration is required.

## 5 NSX Design Considerations

Logical networks implemented by NSX using VXLAN overlay technology can be deployed over common data center network topologies. This section of the design guide addresses requirements for the physical network and examines design considerations for deploying NSX network virtualization solution.

### 5.1 Topology Independent Design with NSX

The NSX enables a multicast free, controller based VXLAN overlay network. Its extreme flexibility enables it to work with:

- Any type of physical topology – PoD, routed, or spine-leaf
- Any vendor physical switch
- Any install base with any topology – L2, L3, or a converged topology mixture
- Any fabric where proprietary controls or interaction does not require specific considerations

The only two strict requirements for NSX from the physical network are IP connectivity and jumbo MTU support.

In this section, the terms access layer switch, top-of-rack (ToR) switch, and leaf switch are used interchangeably. Topologically, a pair of leaf switches is typically located inside a rack and provides network access to the servers connected to that rack. The terms aggregation and spine layer – which effectively provide connectivity between racks – refer to the location in the network that aggregates all the access switches.

A key benefit of NSX's network virtualization functionality is virtual-to-physical network abstraction and decoupling. For successful NSX deployment, it is important that the physical fabric provides robust IP transport with the following attributes:

- High-bandwidth & fault-tolerance
- Jumbo MTU
- Quality of Service (QoS)

#### High-bandwidth & Fault-tolerance

Compute workload expansion and contraction is responsible for most of the moves, adds, and changes (MAC) in datacenter physical network configurations. NSX streamlines this operational overhead, turning connectivity of the compute cluster/hosts to a ToR into a repeatable, one-time configuration.

Compute workloads generate most of the east-west traffic. East-west traffic bandwidth capacity is bound by the cross sectional bandwidth of the fabric, which is determined by the number and size of uplinks from each ToR switch and topology details. Inter-rack traffic capacity is further governed by the degree of oversubscription built into the architecture. In a typical data center physical topology, the oversubscription ratio ranges from 1:4 to 1:12, depending upon the uplink bandwidth and number of spines in a spine-leaf architecture. Similarly for

north-south traffic, oversubscription is further determined by border-leaf or aggregation uplinks to the data center core. The north-south traffic is handled by Edge VMs in NSX. The design consideration for Edge connectivity is discussed in “[Edge Design and Deployment Considerations](#)” section.

Network fault-tolerance is governed by the topology. This is applicable to either classic or leaf-spine topology as well as uplinks from the ToR switch and rack connectivity. The wider the fabric (i.e., number of spines and uplinks from the ToR), the lesser the impact to applications in terms of loss of bandwidth and number of flows that need re-termination on existing links. For proper host-level fault tolerance in NSX, it is recommended to provide at least dual uplinks from each host; typically 2 x 10 Gbps or 4 x 10 Gbps NIC configurations. Rack-level redundancy can provide additional fault-tolerance if a highly available design where a failure of a rack is not acceptable.

### Jumbo MTU

When leveraging encapsulation technologies, it is important to increase the MTU supported both on ESXi hosts as well as on all interfaces of the devices deployed in the physical network. VXLAN adds 50 Bytes to each original frame, leading to the recommendation to increase the MTU size to at least 1600 bytes. The MTU on the ESXi server, both for the internal logical switch and for the VDS uplinks, is automatically tuned to 1600 bytes when provisioning VXLAN to hosts from the NSX Manager UI.

### Quality of Service (QoS)

Virtualized environments must carry various types of traffic – including tenant, storage and management – across the switching infrastructure. Each traffic type has different characteristics and applies different demands on the physical switching infrastructure. Although management traffic typically is low in volume, it can be critical for controlling physical and virtual network state. IP storage traffic typically is high in volume and generally stays within a data center. The cloud operator might be offering various levels of service for tenants. Different tenants’ traffic carries different quality of service (QoS) values across the fabric.

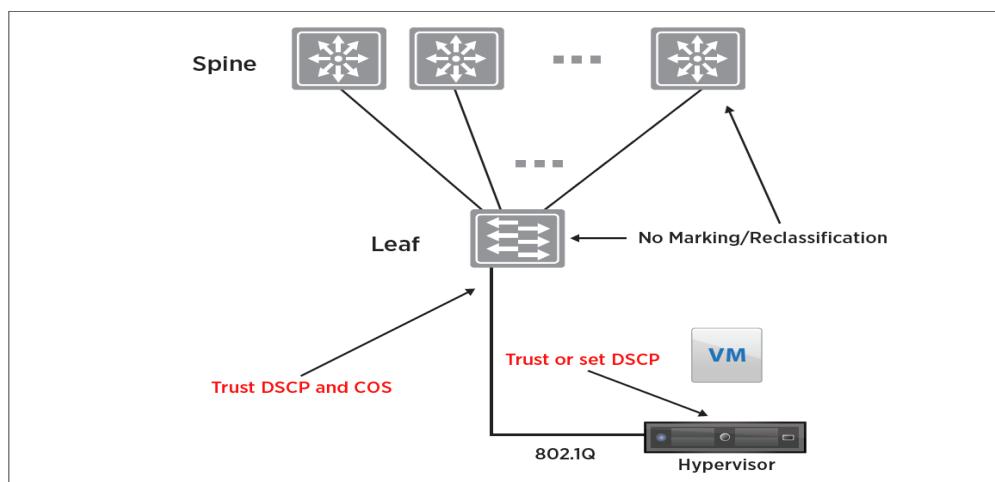


Figure 96 - Quality of Service (QoS) Tagging

For virtualized environments, the hypervisor presents a trusted boundary, setting the respective QoS values for the different traffic types. The physical switching infrastructure is expected to trust these values. No reclassification is necessary at the server-facing port of a leaf switch. If there were a congestion point in the physical switching infrastructure, the QoS values would be evaluated to determine how traffic should be sequenced – and potentially dropped – or prioritized.

There are two types of QoS configuration supported in the physical switching infrastructure; one is handled at L2, and the other at the L3 or IP layer. L2 QoS is sometimes referred to as “Class of Service” (CoS) and the L3 QoS as “DSCP marking”.

NSX allows for trusting the DSCP marking originally applied by a virtual machine or explicitly modifying and setting the DSCP value at the logical switch level. In each case, the DSCP value is then propagated to the outer IP header of VXLAN encapsulated frames. This enables the external physical network to prioritize the traffic based on the DSCP setting on the external header.

## 5.2 VLAN Connectivity with NSX

In any topology, physical ports facing the servers inside a rack have a minimal repeatable configuration. The 801.Q trunk from ESXi hosts carries only four VLANs: VXLAN guest traffic, management, storage, and VMware vSphere vMotion traffic for compute. The need for this L2 trunk is independent from the specific server’s uplinks configuration. Details are discussed further in the “VDS Uplinks Connectivity ” section. The Edge cluster has additional VLANs for peering with physical routers; bridging traffic and is discussed in “[Edge Design and Deployment Considerations](#)”

The modular design shown in Figure 96 represents a classical datacenter multi-tier design, where multiple pods can be interconnected via an L3 datacenter core. Each pod represents an aggregation block that limits the extension of VLANs within each pod.

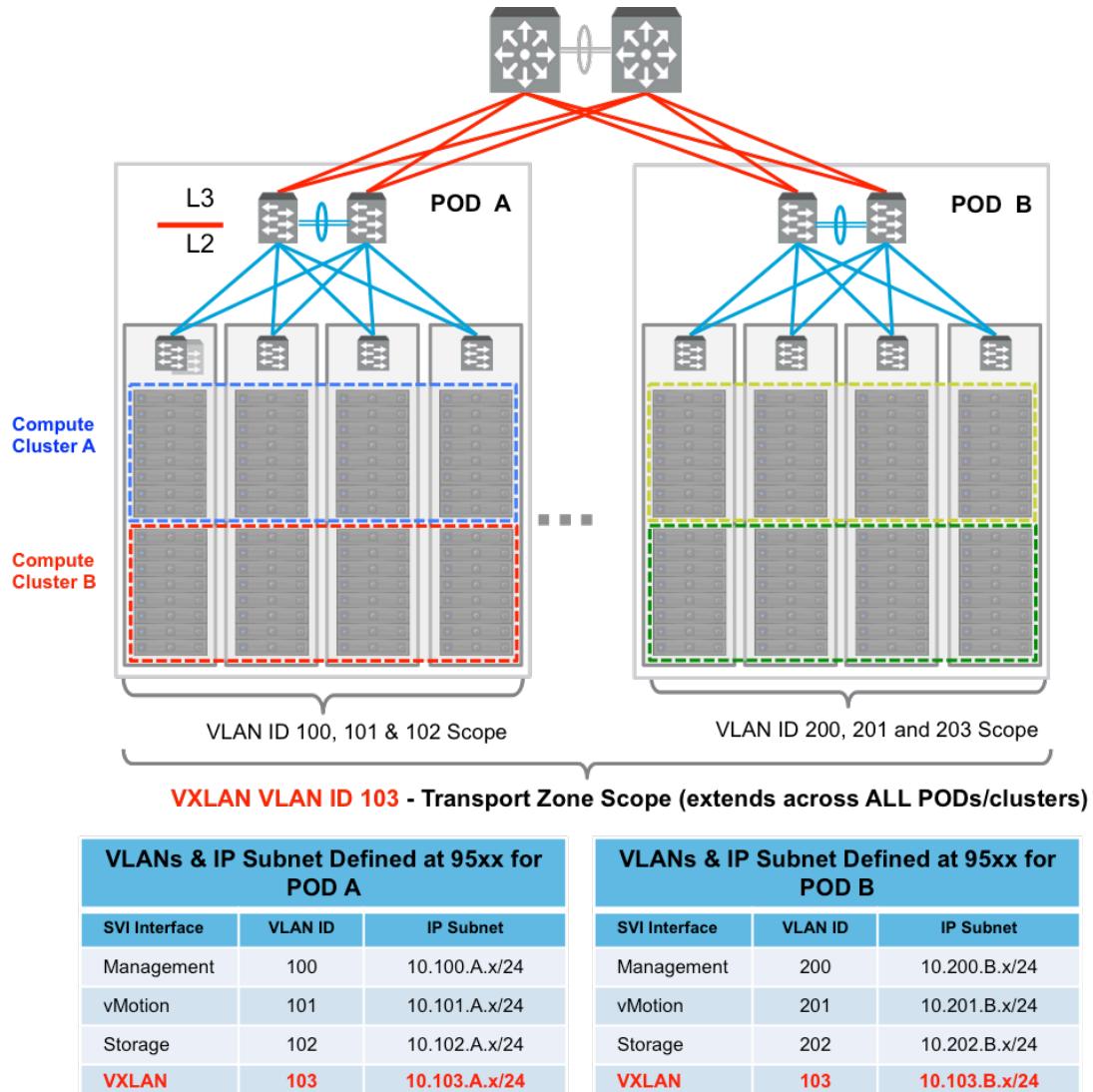


Figure 97 - Classical Access/Aggregation/Core Datacenter Network

The aggregation layer devices of each pod are the demarcation line between L2 and L3 network domains. This is done to limit the extension of L2 broadcast and failure domains. As a result, any connectivity requiring L2 adjacency (e.g., application, VM live migration) must be deployed inside a given pod. This limits the mobility of VM connectivity to a pod. NSX decouples this restriction by providing the logical connectivity inside the hypervisor, providing an overlay network across the pod. The VXLAN transport VLAN 103 in Figure 96 spans both pods, however the rest of the VLANs, 100 – 102 and 200 – 202, are local to the pod as shown.

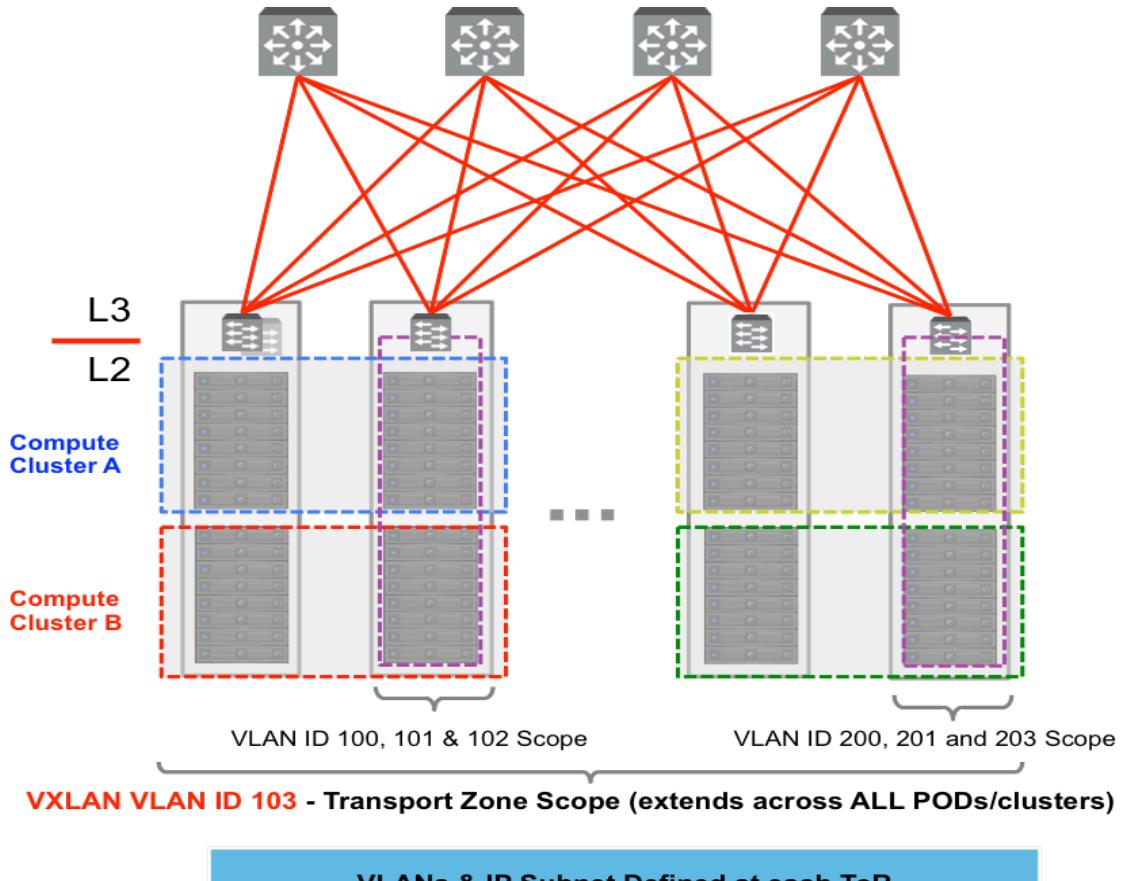
One of the design requirement for VDS is that its uplink configuration must be consistent; in this case the VXLAN transport VLAN ID. Typically in a layer-2 topology, the VLAN ID must only be unique to the layer-2 domain. In this example, two distinct layer-2 pods each have locally unique VLAN ID for non-VXLAN traffic. The VXLAN transport zone which defines the mobility and the

scope of the VXLAN enabled cluster spans both pods. The VXLAN-enabled VDS that spans both pods (e.g., hosts/clusters) must have consistent VLANs on its uplinks. This implies that the VLAN ID for VXLAN transport has to be the same for both the pods. In other words, map the VLAN designated for VXLAN transport with two different subnets for the same VLAN ID. This is depicted with the VLAN ID for VXLAN transport set to 103 in both pods, though the subnet that it maps to is unique at each aggregation layer. For each pod, VXLAN VTEP that maps to this VLAN 103 will have a unique subnet, but not a unique VLAN.

In the recommended design, compute and edge clusters each have a dedicated VDS. While each VDS can have a unique VLAN for VXLAN transport, use of a common VLAN ID for the NSX transport is recommended for operational simplicity and ease of configuration.

The multi-pod case is similar to a spine-leaf routed data center design. In a spine-leaf routed datacenter, L3 demarcation starts at the access layer.

Classical modular network design has evolved into a routed datacenter or leaf-spine design as shown in Figure 97. The routed fabric design offers the same characteristics of low oversubscription, high east-west bandwidth availability, performance predictability, and configuration uniformity. It enhances overall resiliency and device interoperability since switches from multiple vendors can be interconnected. The main challenge of a routed design is the inability to support applications that require L2 adjacency between different nodes, given that the extension of an L2 domain is limited behind each leaf switch (e.g., ToR). This is where NSX and network virtualization come to the rescue. The decoupling of logical and physical networks allows for flexible connectivity in the logical space, independent from how the underlay network is configured.



VLANs & IP Subnet Defined at each ToR		
SVI Interface	VLAN ID	IP Subnet
Management	100	10.100.R_ID.x/24
vMotion	101	10.101.R_ID.x/24
Storage	102	10.102.R_ID.x/24
<b>VXLAN</b>	<b>103</b>	<b>10.103.R_ID.x/24</b>

Figure 98 – Routed Datacenter Design

In a routed datacenter (e.g., leaf-spine) design, L3 is terminated at the ToR, thus all the VLANs originating from ESXi hosts terminate on ToR. This typically means the VLAN ID is irrelevant; thus it be kept unique or remains the same for a given type of traffic per rack, as long as the VLAN ID maps to the unique subnet.

The VXLAN transport VLAN requirement is exactly same as described in the classical pod design in Figure 96. VLAN ID must be the same for VXLAN transport even though it maps to unique subnet. In other words, for the VTEP, the VLAN ID remains the same for every ToR; however, the subnet that maps to VLANs is unique per ToR. The VLAN ID can be kept the same in every rack for the rest of the traffic types to aid in simplicity, or it can be made unique for troubleshooting purposes. Keeping the same VLAN ID simplifies the configuration for every rack and only requires configuration once. This is depicted in the table in Figure 97, which can be repeated for each ToR configuration with the unique subnet identified by rack ID.

NSX can be deployed in any topology, new or existing deployment, with a choice of physical switch vendor. The rest of the design selects topology specifics for illustrative purposes, however the principles and techniques are agnostically applicable to any topology.

### 5.3 NSX Deployment Considerations

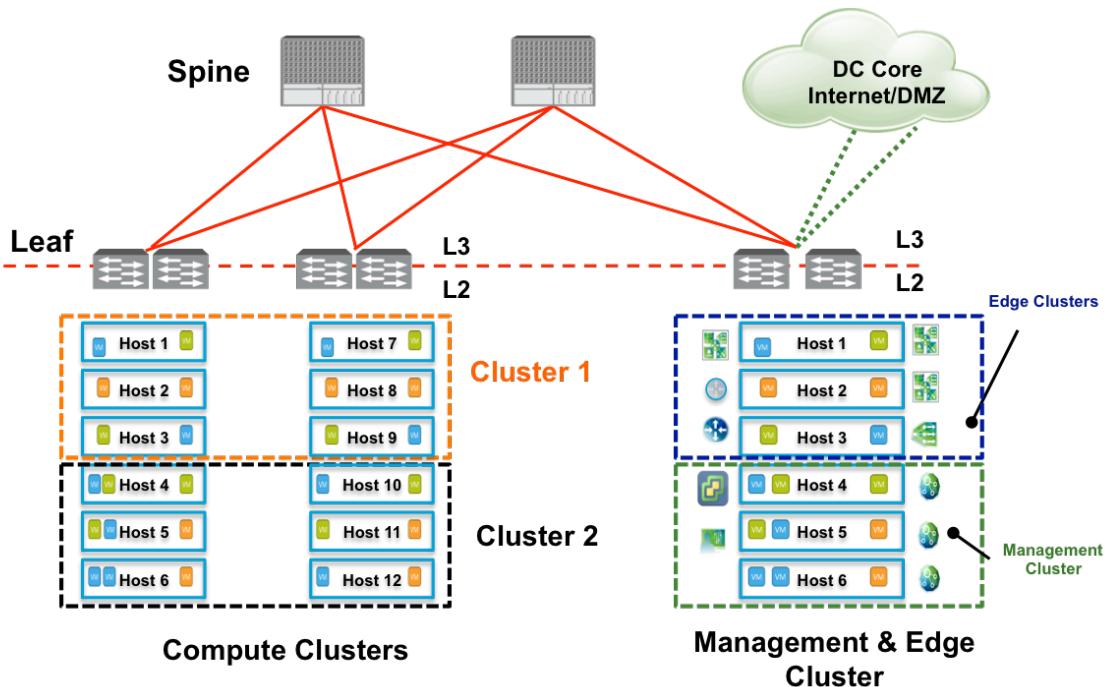
NSX network virtualization consists of three major aspects: decouple, reproduce and automate. All three aspects are vital in achieving the desired efficiencies. This section focuses on decoupling, which is key to simplifying and scaling the physical infrastructure.

When building a new environment, it is essential to choose an architecture that allows for future growth. This approach works for deployments that begin small and scale out while keeping the same overall architecture.

In an NSX enabled datacenter, it is desirable to achieve logical separation and grouping of the ESXi hosts providing specific functions such as compute, management, and edge services. This separation is logically arranged and provides the following advantages to the datacenter architect:

- Flexibility of expanding and contracting resources for specific functions.
- Ability to isolate and develop span of control over various control plane components such as control VM, Edge VM deployment, as well other technology integration
- Managing the lifecycle of certain resources for specific functions (e.g., number of socket/core, memory, NIC, upgrade, migration). This is critical for Edge cluster resource as well VXLAN offload capable NIC providing line rate performance for every hosts
- High availability based on functional connectivity needs, (e.g., DRS, FT, Edge P/V and HA).
- Automation control over areas or functions that require frequent changes (e.g., app-tier, security tags, policies, load balancer)

Figure 99 highlights the compute, edge, and management component connected to leaf-spine topology.



**Figure 99 – Compute, Edge, and Management Racks**

While the separation for each type of cluster is important, in smaller deployments it may necessitate to consolidate NSX, management, and infrastructure functions in the same physical racks or combine them into single cluster. The NSX is flexible in adapting various sizes depending on workload requirements.

The datacenter cluster configuration choices and NSX sizing consideration for various types of workloads is considered last in “[DC Cluster Configurations & Sizing with NSX](#)” as it would require deeper considerations on following key components of virtualized solution:

- Datacenter cluster types and its characteristics
- vCenter integration
- VDS design NSX
- VDS uplinks connectivity
- ESXi host traffic types
- Edge design and deployment considerations

### 5.3.1 Cluster Types and Characteristics

**Management Cluster:** The management cluster hosts the management components – including vCenter Server, NSX manager, NSX controller, Cloud Management Systems (CMS), and other shared components. Compute and memory requirements for hosts and resources are typically pre-identified based on the required scale and minimum supported configurations. Capacity and

resource requirements are fairly consistent. The availability of components is important, and the management host can improve its resiliency by enabling LACP. Additionally, rack redundancy is not a requirement in a small to medium environment, but it may become necessary at scale.

The ESXi hosts in the cluster do not require VXLAN provisioning, however in small and medium designs where management cluster is collapsed with Edge cluster it is necessary to prepare the management cluster for VXLAN.

**Compute Cluster:** The compute cluster consists of the part of the infrastructure where workloads are provisioned and tenant virtual machines connectivity is enabled via logical networks. Compute clusters are designed with following considerations:

- Rack based vs. multi-rack/horizontal striping is based on following criteria:
  - Host density per rack and automation dependencies (auto-deployment per rack or across rack)
  - Availability and mobility of workload
  - Connectivity – either single VTEP vs. multiple VTEPs
  - Topology implication (layer 2 vs layer 3) on IP addressing for VTEP and VMkernel
- Lifecycle of the workload drives the consideration for:
  - Growth and changes in the application
  - Multi-rack, zoning (e.g., PCI, tenancy)
  - Same four VLANs are required for each rack enable streamlined repetition of rack connectivity
- Workload centric resources allocation, compliance, and SLA adherence can be met via:
  - Cluster separation
  - Separate VXLAN transport zone
  - Per tenant DLR and Edge routing domains
  - DRS and resource reservation

**Edge Cluster:** The edge cluster is a key to developing design and capacity for NSX enabled workloads. It provides critical interaction with the physical infrastructure. The edge cluster has following characteristics and functions:

- Provide on-ramp and off-ramp connectivity to physical networks (i.e., north-south L3 routing on NSX Edge virtual appliances).
- Allow communication with physical devices connected to VLANs in the physical networks through NSX L2 bridging.
- Hosts the Control VM for DLR routing.

- May have centralized logical or physical services (e.g., firewall, load balancers, VPN monitoring components, log insight VMs).
- NSX Controllers can be hosted in an edge cluster, when a dedicated vCenter is used to manage the compute and edge resources.
- Edge cluster resources have anti-affinity requirement to protect the active-standby configuration or to maintain the bandwidth availability during failure.
- Edge workloads have specific connectivity needs and characteristics:
  - Edge VM forwarding is CPU-centric with consistent memory requirements.
  - Additional VLANs are required for north-south routing connectivity and bridging. Edge resources require external connectivity to physical network devices, possibly constraining physical location placement in an effort to minimize VLAN spread.
  - Recommended teaming option for Edge hosts is “route based on SRC-ID”. Use of LACP is highly discouraged due to vendor specific requirements of route peering over LACP. This recommendation is discussed in detail in the [“Edge Design and Deployment Considerations”](#) section.

### 5.3.2 vCenter Design with NSX

The vCenter design for an NSX enabled data center falls into two broad categories, small/medium and large-scale deployments. In both scenarios, the following considerations are valid when deploying NSX:

- The vCenter and NSX manager have a one-to-one mapping relationship. There is only one NSX Manager (i.e., NSX domain) per given vCenter. This implies that the scalability limit of vCenter governs the scalability of the overall NSX deployment.
- As part of the installation process, the NSX controllers must be deployed into the same vCenter Server where NSX manager is connected.
- The difference in small/medium vs. large-scale designs is usually determined by the number of vCenter deployed and their mapping to NSX manager.

In a small/medium datacenter deployment, a single vCenter is usually deployed for managing all the NSX components, as shown in Figure 100.

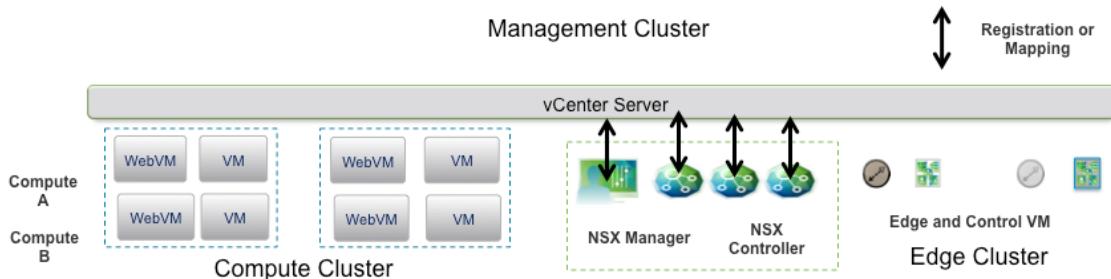


Figure 100 - Single vCenter for Managing All NSX Components

The design consideration either to separate or collapse clusters in small/medium design will be discussed in "[DC Cluster Configurations & Sizing with NSX](#)" section.

Most enterprises deploy multiple vCenters for administrative and workload segmentation. Merger, divestiture, and consolidation practicalities may also mandate the use of a dedicated vCenter management cluster to servicing multiple vCenters. Figure 101 shows multiple vCenters and the associated NSX manager mapping.

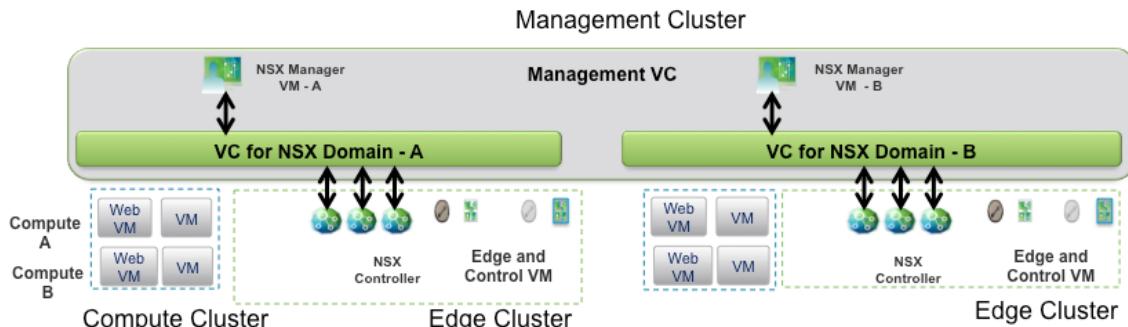


Figure 101 - Dedicate vCenter Servers for Managing NSX Resources

In this example, the NSX manger VM resides with the management vCenter while its configuration points to another vCenter where a set of compute and edge clusters are attached servicing specific NSX domain. There are several advantages in adopting such approach:

- Avoid circular dependencies – the management cluster does not need to be prepared for VXLAN and normally resides outside of the domain it manages
- Mobility of management cluster for remote datacenter operation.
- Integration with an existing vCenter.
- Ability to deploy more than one NSX domain and upgrade independent of each other.
- Upgrade of management vCenter does not directly impact the NSX domains.
- SRM and other explicit state management are possible.

- Ability to develop workload management policy with multiple vCenters while providing migration, consolidation and separation with new multi-VC features available in vSphere 6.0 and NSX release 6.2.

### 5.3.3 VDS Design in an NSX Domain

The vSphere Virtual Distributed Switch (VDS) is a foundational component in creating VXLAN logical segments. The span of VXLAN logical switches, an L2 broadcast domain, is governed by the definition of a transport zone; each logical switch can span across separate VDS instances over the entire transport zone.

Figure 102 depicts the relationship between VDS, ESXi clusters, and a transport zone. It recommends a design leveraging separate VDS switches for the compute and the edge clusters.

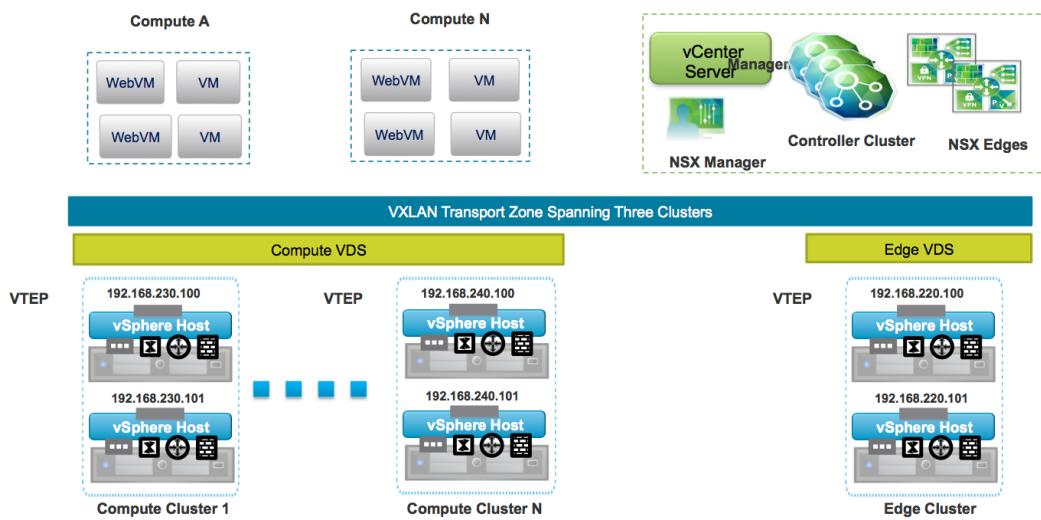


Figure 102 - Cluster, VDS and Transport Zone

Although a design with a single VDS spanning both compute and edge cluster is possible, there are several advantages in keeping separate VDS for compute and edge:

- Flexibility in span of operational control. Compute/virtual infrastructure administration and network administration are typically separate groups, allowing each to manage the cluster specific tasks. These benefits are already a factor in designing a dedicated cluster and rack for specific services and further substantiated by the VDS design choices. However, in a small or medium size design it is possible to consolidate clusters and provide the access control for Edge resources with policy such that only network administrators can have operational responsibility.
- Flexibility in managing uplink connectivity on computes and edge clusters. See "[VDS Uplinks Connectivity NSX Design Considerations](#)" section on uplink design and the recommendation of using different teaming options for compute and edge clusters.

- The VDS boundary is typically aligned with the transport zone, allowing VMs connected to logical switches to span the transport zone. The vMotion boundary is always limited by the extension of a VDS, so keeping a separate VDS for compute resources ensures that those workloads will never be moved – either by mistake or by choice – to ESXi hosts dedicated to other services.
- Flexibility in managing VTEP configuration.
- Avoid exposing VLAN-backed port groups used by the services deployed in the edge racks (e.g., NSX L3 routing and NSX L2 bridging) to the compute racks, thus enabling repeatable streamlined configuration for the compute rack.

The VDS used for infrastructure services is not part of the VXLAN as the management cluster typically represents an independent entity providing functions that goes beyond serving a given NSX domain.

#### 5.3.4 VDS Uplinks Connectivity NSX Design Considerations

The general design criteria used for connecting ESXi hosts to the ToR switches for each type of rack takes into consideration:

- The type of traffic carried – VXLAN, vMotion, management, storage. Specific focus in this case is on VXLAN traffic as it is the specific additional traffic type found in NSX deployments.
- Type of isolation required based on traffic SLA – dedicated uplinks (e.g., for vMotion/Management) vs. shared uplinks.
- Type of cluster – compute workloads, edge, and management either with or without storage.
- The amount of bandwidth required for VXLAN traffic (single vs multiple VTEP).
- Simplicity of configuration – LACP vs. non-LACP.
- Convergence and uplink utilization factors - flow-based vs. MAC-based.

VDS leverages a special port-group called dvUplink for uplink connectivity. Table 6 shows the teaming options supported for the VXLAN port-group dynamically created when NSX is deployed on an ESXi cluster. The teaming option for that port-group must be specified during the VXLAN provisioning process.

Teaming and Failover Mode	NSX Support	Multi-VTEP Support	Uplink Behavior 2 x 10G
<b>Route Based on Originating Port</b>	✓	✓	Both Active
<b>Route Based on Source MAC Hash</b>	✓	✓	Both Active

LACP	✓	✗	Flow based
<b>Route Based on IP Hash (Static EtherChannel)</b>	✓	✗	Flow based
<b>Explicit Failover Order</b>	✓	✗	Only one link is active
<b>Route Based on Physical NIC Load (LBT)</b>	✗	✗	✗

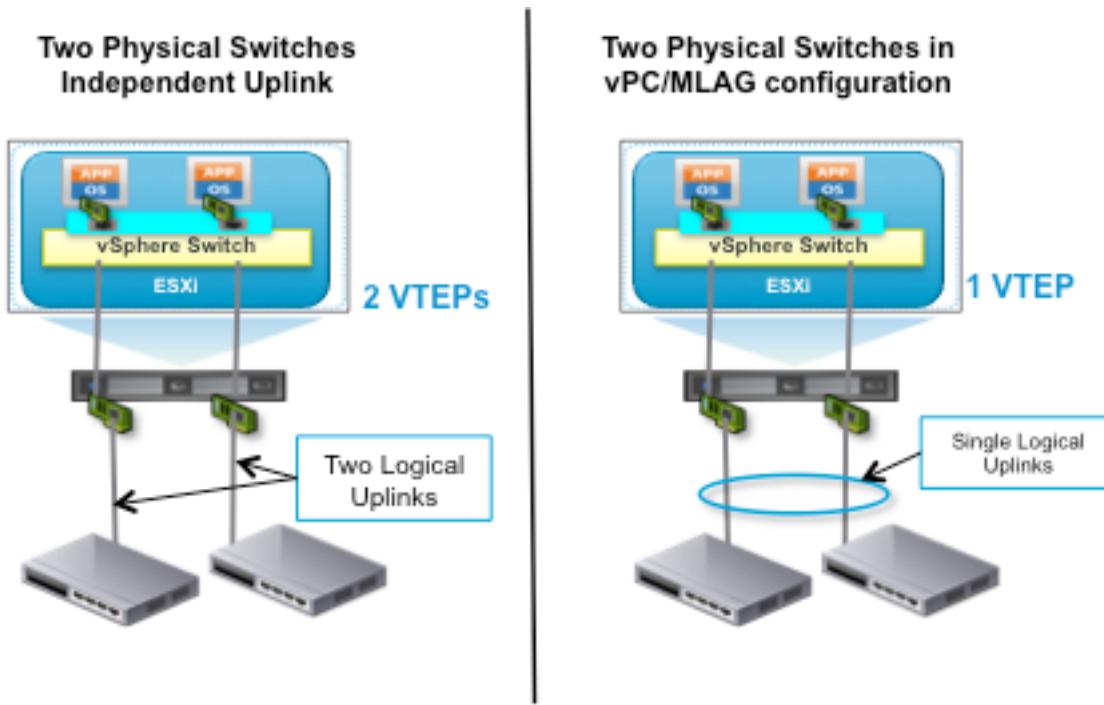
**Table 6 – Teaming Options for Uplink Connectivity in NSX**

The only teaming option not supported for VXLAN traffic is Load-Based Teaming (LBT). All the other flavors can be utilized, and the specific choice made will impact how VXLAN encapsulated traffic is sent and received on the VDS uplinks, as well as the number of VMkernel (VTEP) interfaces created.

Additional considerations for how the selected uplink connectivity impacts the flexibility of creating port-groups and how it is inherited across the cluster or across the transport zone include:

- VDS offers great flexibility for uplink connectivity. Every port-group defined as part of a VDS could make use of a different teaming option, depending on the requirements of the traffic associated to that specific port-group.
- The teaming option associated to a given port-group must be the same for all the ESXi hosts connected to that specific VDS, even if they belong to separate clusters. For the case of the VXLAN transport port-group, if the LACP teaming option is chosen for the ESXi hosts part of compute cluster 1, this same option must be applied to all the other compute clusters connected to the same VDS. Selecting a different teaming option for a different cluster would not be accepted and will trigger an error message at the time of VXLAN provisioning.
- If LACP or static EtherChannel is selected as the teaming option for VXLAN traffic for clusters belonging to a given VDS, then the same option should be used for all the other port-groups/traffic types defined on the VDS. The EtherChannel teaming choice implies a requirement of additionally configuring a port-channel on the physical network. Once the physical ESXi uplinks are bundled in a port-channel on the physical switch or switches, using a different teaming method on the host side may result in unpredictable results or loss of communication. This is one of the primary reasons that use of LACP is highly discouraged, along with restriction it imposes on Edge VM routing connectivity with proprietary technology such as vPC from Cisco.
- The VTEP design impacts the uplink selection choice since NSX supports single or multiple VTEPs configurations. The single VTEP option offers operational simplicity. If traffic requirements are less than 10G of VXLAN

traffic per host, then the Explicit Failover Order option is valid. It allows physical separation of the overlay traffic from all the other types of communication; one uplink used for VXLAN, the other uplink for the other traffic types. The use of Explicit Failover Order can also provide applications a consistent quality and experience, independent from any failure. Dedicating a pair of physical uplinks to VXLAN traffic and configuring them as active/standby will guarantee that in the event of physical link or switch failure, applications would still have access to the same 10G pipe of bandwidth. This comes at the price of deploying more physical uplinks and actively using only half of the available bandwidth.



**Figure 103 - Single and multi VTEP Uplink Options**

Figure 103 shows both single and multiple VTEP options. A single VTEP is provisioned in the port-channel scenarios on the right, despite the fact there are two active VDS uplinks. This is because the port-channel is considered a single logical uplink since traffic sourced from the VTEP can be hashed on a per-flow basis across both physical paths.

Since the single VTEP is only associated in one uplink in a non port-channel teaming mode, the bandwidth for VXLAN is constrained by the physical NIC speed. If more than 10G of bandwidth is required for workload, multiple VTEPs are required to increase the bandwidth available for VXLAN traffic when the use of port-channels is not possible (e.g., blade servers). Alternately, the VDS uplink configuration can be decoupled from the physical switch configuration. Going beyond two VTEPs (e.g., four uplinks) will result into four VTEP configurations for the host, may be challenging while troubleshooting, and will require a larger IP addressing scope for large-scale L2 design.

The number of provisioned VTEPs always matches the number of physical VDS uplinks. This is done automatically once the SRC-ID or SRC-MAC option is selected in the “VMKNic Teaming Policy” section of the UI interface shown in Figure 104.

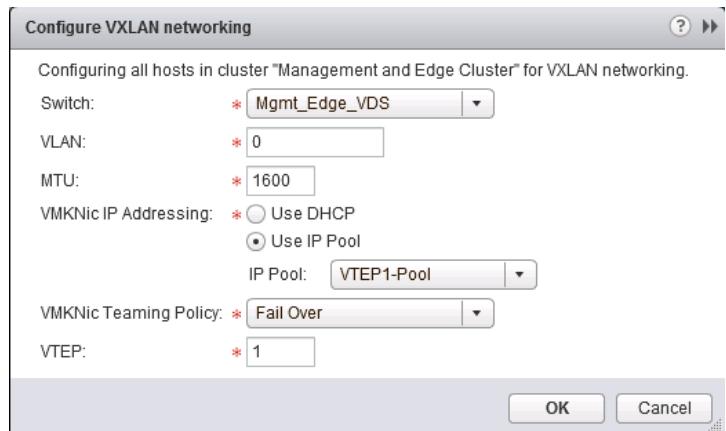


Figure 104 - Configure VXLAN Networking

The recommended teaming mode for the ESXi hosts in edge clusters is “route based on originating port” while avoiding the LACP or static EtherChannel options. Selecting LACP for VXLAN traffic implies that the same teaming option must be used for all the other port-groups/traffic types which are part of the same VDS. One of the main functions of the edge racks is providing connectivity to the physical network infrastructure. This is typically done using a dedicated VLAN-backed port-group where the NSX Edge handling the north-south routed communication establishes routing adjacencies with the next-hop L3 devices. Selecting LACP or static EtherChannel for this VLAN-backed port-group when the ToR switches perform the roles of L3 devices complicates the interaction between the NSX Edge and the ToR devices.

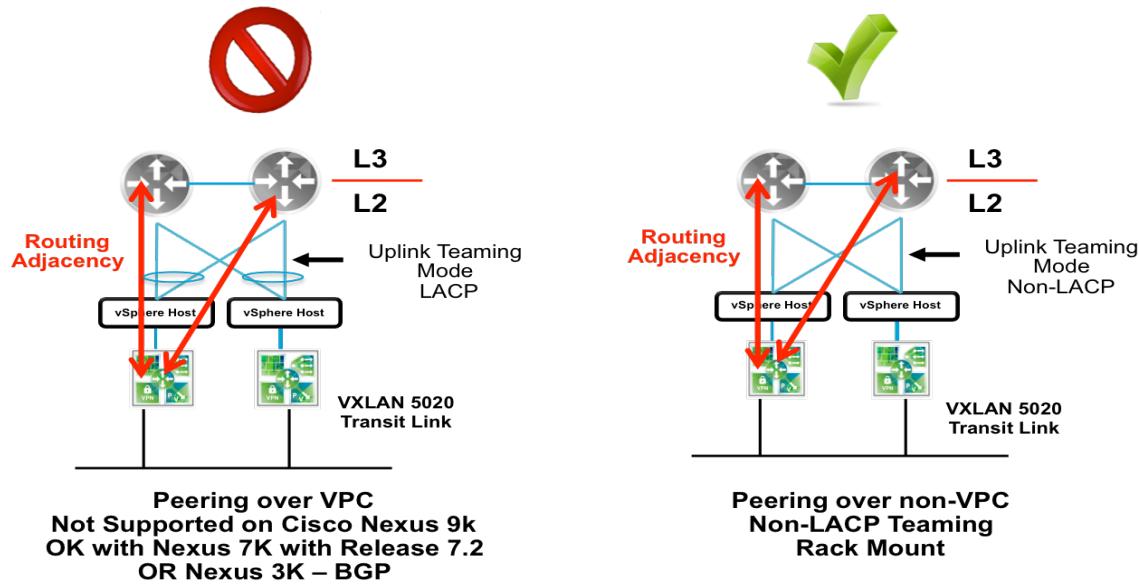


Figure 105 - Routing Peering Over Multi-Chassis EtherChannel

As shown in Figure 105, the ToR switches not only need to support multi-chassis EtherChannel functionality (e.g., vPC or MLAG) but must also be capable of establishing routing adjacencies with the NSX Edge on this logical connection. This creates an even stronger dependency from the underlying physical network device; this may be an unsupported configuration in specific cases. As a consequence, the recommendation for edge clusters is to select the SRC-ID/SRC-MAC hash as teaming options for VXLAN traffic. An architect can also extend this recommended approach for the compute cluster to maintain the configuration consistency, automation ease, and operational troubleshooting experience for all hosts' uplink connectivity.

In summary, the “route based on originating port” is the recommended teaming mode for VXLAN traffic for both compute and edge cluster.

### 5.3.5 ESXi Host Traffic Types

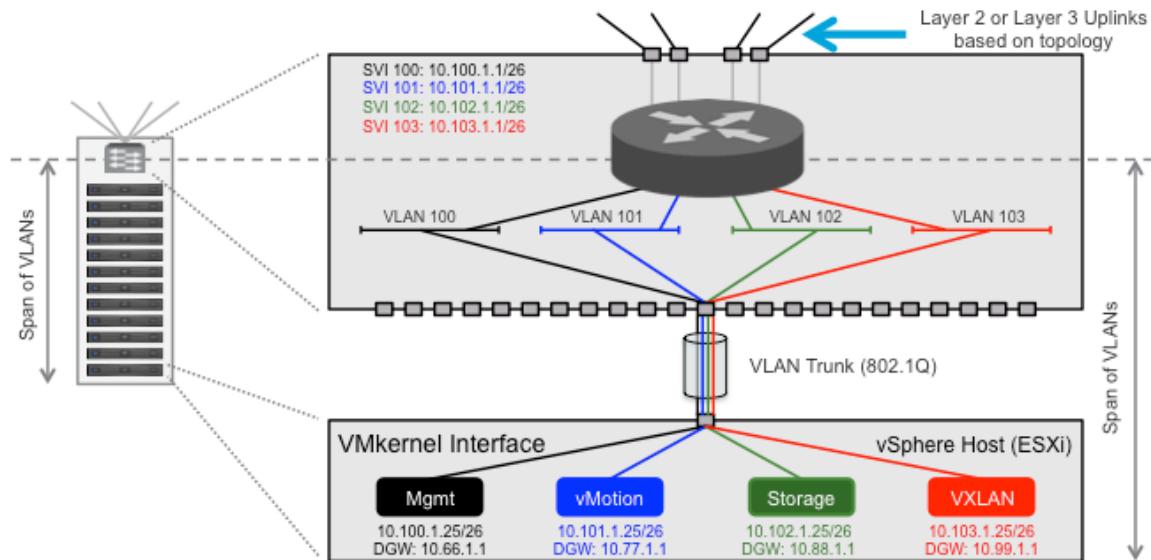
ESXi hypervisors deployed in an NSX domain typically source different types of traffic. Common traffic types of interest include overlay, management, vSphere, vMotion and storage traffic. The overlay traffic is a new traffic type that carries all the virtual machine communication and encapsulates it in UDP (VXLAN) frames. VXLAN traffic is usually sourced by ESXi host's part of compute and edge clusters, but not part of the management clusters. The other types of traffic are usually leveraged across the overall server infrastructure.

Traditional designs called for the use of different 1G NIC interfaces to carry different types of traffic in and out of the ESXi hypervisor. With the ever-increasing adoption of 10G interfaces in the data center, consolidation of different types of traffic on a common pairs of uplinks is becoming more popular.

Different traffic types can be segregated via VLANs, enabling clear separation from an IP addressing standpoint. In the vSphere architecture, specific internal interfaces are defined on each ESXi host to source those different types of traffic.

Those are called VMkernel interfaces and are assigned discrete VLANs and IP addresses.

When deploying a routed data center fabric, each VLAN terminates at the leaf switch (i.e., Top-of-Rack device), so the leaf switch will provide an L3 interface for each VLAN. Such interfaces are also known as SVIs or RVIs. Figure 106 diagrams this model.



**Figure 106 - Example of Host and Leaf Switch Configuration in a Rack**

In the L2 topology, these VLANs are terminated at aggregation and the configuration requirement for NSX is consistent regardless of topology. In this case the VM default gateway is DLR interface (LIF). The guest VM traffic will be carried by VXLAN VMkernel – VLAN 103 in Figure 105 – to other hosts or Edge VMs over VXLAN overlay.

### 5.3.5.1 VXLAN Traffic

After the vSphere hosts have been prepared for network virtualization using VXLAN, a new traffic type is enabled on the hosts. Virtual machines connected to one of the VXLAN-based logical L2 networks use this traffic type to communicate. The traffic from the virtual machine is encapsulated and sent out as VXLAN traffic. The external physical fabric never detects the virtual machine IP and MAC address.

The VXLAN Tunnel Endpoint (VTEP) IP address associated with a VMkernel interface is used to transport the frame across the fabric. In the case of VXLAN, the tunnels are initiated and terminated by a VTEP. For east-west traffic, both the source and destination VTEPs are situated in hypervisors located in compute and edge racks. North-south traffic leaving the data center will flow between a tenant virtual machine and an NSX edge.

When deploying NSX over a routed fabric infrastructure, the VTEP interfaces for hosts connected to different compute and edge racks must be deployed as part

of different IP subnets (i.e., VXLAN transport subnets). The combination of this with striping compute and edge clusters across separate racks creates an interesting deployment challenge. Provisioning VXLAN on ESXi hosts is performed at the cluster level, and during this process the user must specify how to address the VTEP interfaces for the hosts belonging to that cluster. This is referenced in Figure 102.

The challenge of assigning the VTEP interfaces IP addresses in different subnets can be addressed in two ways:

- Leveraging the “Use DHCP” option. In this manner, each VTEP will receive an address in the proper IP subnet, depending on the specific rack where it is connected. This is the recommended approach for production deployment scenarios.
- Statically assigning IP addresses to the VTEPs on a per host basis, either via ESXi CLI or via the vSphere Web Client UI. In this recommendation, select the DHCP option when provisioning VXLAN to an ESXi cluster, wait for the DHCP process to time-out, and statically assign the VTEP address. This can be done only for smaller scale deployments.

### 5.3.5.2 Management Traffic

Management traffic is sourced and terminated by the management VMkernel interface on the host. It includes the communication between vCenter Server and hosts as well as communication with other management tools such as NSX manager.

A single VDS can span multiple hypervisors that are deployed beyond a single leaf switch. Because no VLANs can be extended beyond a leaf switch, the management interfaces of hypervisors participating in a common VDS and connected to separate leaf switches are in separate IP subnets.

### 5.3.5.3 vMotion Traffic

During the vSphere vMotion migration process, running virtual machine state is transferred over the network to another host. The vSphere vMotion VMkernel interface on each host is used to move this virtual machine state. Each vSphere vMotion VMkernel interface on the host is assigned an IP address. Depending on the speed of the physical NIC, the number of simultaneous virtual machine vSphere vMotion migrations is decided. On a 10GbE NIC, eight simultaneous vSphere vMotion migrations can be performed.

From a VMware support point of view, the historical recommendation has always been to deploy all the VMkernel interfaces used for vMotion as part of a common IP subnet. This is not possible when designing the network for virtualization using L3 in the access layer, as it is mandatory to select different subnets in different racks for those VMkernel interfaces. Until this design is supported by VMware, it is recommended that users go through the RPQ process and allow VMware validate designs on a case-by-case basis.

#### 5.3.5.4 Storage Traffic

A VMkernel interface is used to provide features such as shared or non-directly attached storage. This refers to storage that can be attached via an IP connection (e.g., NAS or iSCSI) rather than FC or FCoE. From an IP addressing standpoint, the same rules that apply to management traffic apply to storage VMkernel interfaces - the storage VMkernel interface of servers inside a rack connected to a leaf switch is part of the same IP subnet. This subnet cannot span beyond this leaf switch, therefore the storage VMkernel interface IP of a host in a different rack is in a different subnet.

#### 5.3.5.5 Host Profiles for VMkernel Interface IP Addressing

vSphere host profiles allow automation of the provisioning of IP addresses to the VMkernel NICs for each traffic type. The host profile feature enables users to create a reference host with properties that are shared across the entire deployment. After this host has been identified and basic configured has been performed, a host profile can be created from that host and applied across the other hosts in the deployment. This approach allows for rapid configuration of large numbers of hosts.

In a sample configuration, the same set of four VLANs – storage, vSphere vMotion, VXLAN, management – is usually provided in each rack. The VLAN IDs associated with those VLANs are also common across racks since the VDS spans clusters of hosts that are striped horizontally. This implies that with an L3 fabric design, IP subnets associated to the same VLAN ID in different racks must change. This is detailed in Table 7.

IP ADDRESS MANAGEMENT AND VLANs <sup>1</sup>		
Function	Global VLAN ID	IP Address
Storage	66	10.66.R_id.x/26
vMotion	77	10.77.R_id.x/26
VXLAN/VTEP	88	10.88.R_id.x/26
Management	99	10.99.R_id.x/26

<sup>1</sup> Values of VLANs, IP addresses, and masks are an example (not prescriptive to the design)

**Table 7 – IP Address Management and VLAN IDs**

The following are among the configurations required per host:

- VMkernel NIC (vmknic) IP configuration per traffic type in the respective IP subnet.
- Static route configuration per subnet to handle proper traffic routing to the respective gateways.
- The static routing configuration needed certain type of traffic based on version of ESXi is deployed – 5.5 or 6.0 and newer.

#### VMware vSphere 5.5 based design

ESXi 5.5-based configuration allows two TCP/IP stacks:

- **VXLAN:** This is dedicated to traffic sourced from the VMkernel VTEP interface. A dedicated default-route 0.0.0.0/0 can be configured on this stack for each ESXi pointing to the gateway deployed on the local ToR. This allows communication with all the remote VTEPs deployed in different transport subnets.
- **Default:** This stack is used for all the other traffic types, including vMotion, management, and storage. It is typical to leverage a default route for management purposes since connectivity to the vmk0 management interface could be originated from many remote IP subnets. This then requires static routing configuration to support inter-subnet communication for the other types of traffic.

## VMware vSphere 6.0 based design

vSphere 6.0-based configurations allow additional choices in TCP/IP stack choice with a dedicated vMotion TCP/IP stack. This TCP/IP stack is optimized for speed and latency tolerance, supporting a latency up to 150ms between hosts providing vMotion. The default stack used for rest of the traffic, excluding VXLAN and vMotion, and still requires static routing

In the example from Table 7, a given host 1 in rack 1 would have the following vmknic configuration:

- A storage vmknic with IP address 10.66.1.10.
- A vSphere vMotion vmknic with IP address 10.77.1.10 for ESXi 5.5 based deployments.
- A management vmknic with IP address 10.99.1.10.

The default gateway configuration on host 1 for the default TCP/IP stack is in the management vmknic subnet 10.99.1.0/26. To support proper routing for other subnets, the following static routes are configured as part of the host 1 preparation:

- Storage network route
  - `esxcli network ip route ipv4 add -n 10.66.0.0/16 -g 10.66.1.1`
- vSphere vMotion network route for ESXi 5.5 based deployments
  - `esxcli network ip route ipv4 add -n 10.77.0.0/16 -g 10.77.1.1`

After host 1 of rack 1 has been configured, a host profile is created and applied to other hosts in the rack. While applying the profile to the hosts, new vmknics are created and the static routes are added, simplifying the deployment.

The VXLAN vmknic would have an IP address 10.88.1.10 and leverage a dedicated default gateway configuration in that subnet. As previously mentioned, the VTEP IP address would normally be assigned leveraging DHCP at the time of provisioning of the VXLAN configuration to the compute/edge clusters.

In the case of a vSphere Auto Deploy environment, the PXE boot infrastructure, along with the Auto Deploy server and vCenter Server, supports the host-booting process, helps automate the deployment, and upgrades the ESXi hosts.

### 5.3.6 Edge Design and Deployment Considerations

The NSX Edge VM services provide an entry point between the logical and the physical networks. This could include north-south routing and VXLAN-VLAN bridging along with any servers enabled at the Edge VM (e.g., VPN, NAT, firewall).

The deployment of L2 and L3 network services imposes the requirement of extending a certain number of VLANs across the edge racks. Those VLANs represent the interface used to enable communication at L2 or L3 between the logical networks and the external physical infrastructure.

The logical connectivity between ToR and Edge cluster is dependent on the type of topology deployed. In routed topology it will directly peer with ToR or for L2 topology it peers to aggregation layer where the L3 boundary resides. In this section only routed fabric for overlay is discussed.

The separation of links from ToR is shown in Figure 107 for VLAN and VMkernel logical connectivity. While not presented here, these two functionalities can be consolidated into same uplinks for the L2 topology.

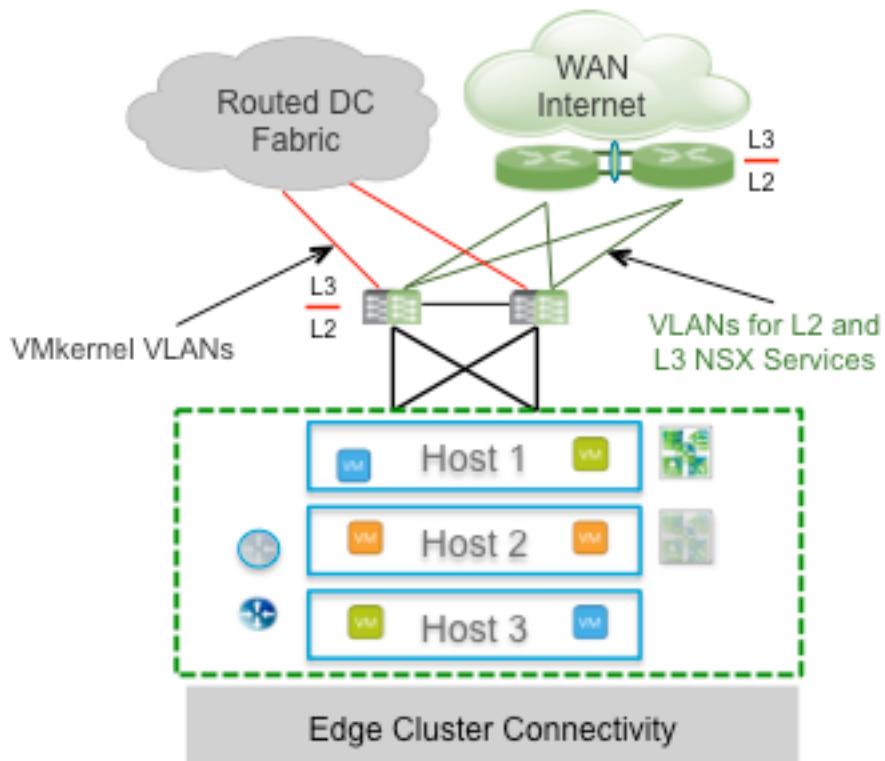


Figure 107 – Connectivity of Edge Racks Toward the Physical Network

In this deployment model, pairs of Top-of-Rack (ToR) switches are connected play a dual role:

1. Function as L3 leaf nodes connecting with L3 point-to-point links to the fabric spine devices. In doing so, the ToR switches offer the L2/L3 boundary and operate as the default gateway for all the local VMkernel VLANs used by the servers (e.g., management, VXLAN, vMotion, storage). The span of those VLAN(s) is limited to ToR, similar to the compute and management racks).
2. The ToR switches also represent the interface between the datacenter fabric and the external physical network infrastructure, and are sometimes referred to as border leafs. In this case they only provide L2 transport functionality for the VLANs that are used to interconnect to the physical network space (i.e., L2 and L3 NSX services). The default gateway for devices connected to those VLANs is usually positioned on a dedicated pair of aggregation routers. A separate VLAN or set of VLANs would be used for each deployed tenant, in case of multi-tenant design.

### 5.3.7 NSX Edge Deployment Considerations

The NSX Edge services gateway can be configured in two modes – active-standby services mode or ECMP. Figure 107 highlights the reference topology that will be used for describing the various HA models. It leverages the NSX Edge between the DLR and the physical network as previously discussed in the “Enterprise Routing Topology” section.

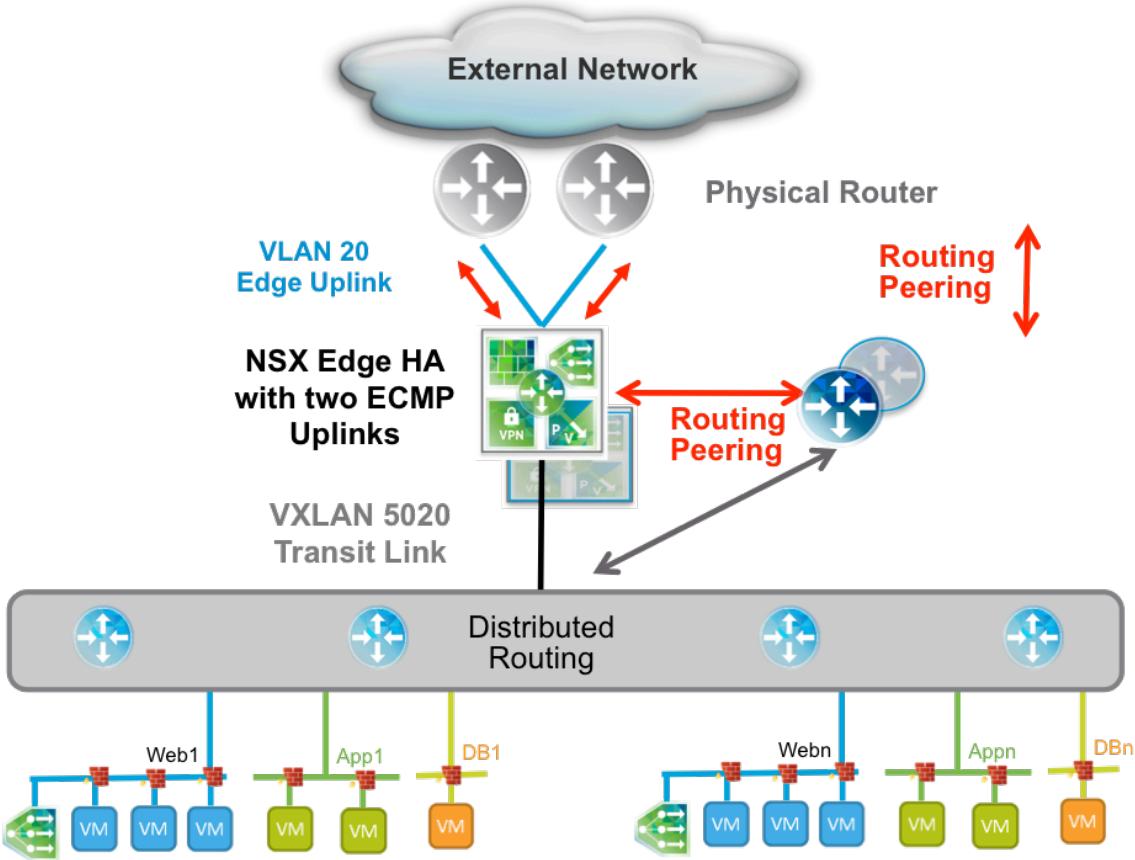


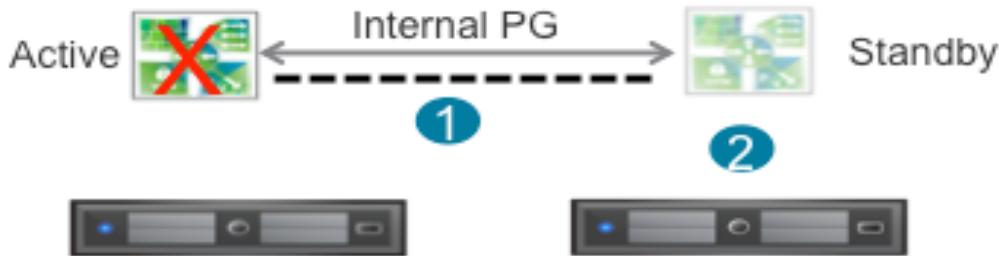
Figure 108 - NSX Logical Routing Reference Topology

This document analyzes in detail these Edge models, specifically focusing on both how to provide resiliency to the NSX Edge and DLR Control VM functional components and what is the impact of their failure on north-south data plane communications.

#### 5.3.7.1 Active/Standby Mode - Stateful Edge Services

This is the redundancy model where a pair of NSX Edge Services Gateways is deployed for each DLR; one Edge functions in active mode, actively forwarding traffic and providing the other logical network services. The second unit is in standby state, waiting to take over should the active Edge fail. To support the active-standby stateful failover model, a heartbeat is exchanged between active and standby systems. Router peering and exchange of forwarding information is only performed with the active Edge; this occurs with both the physical router and control VM as shown in Figure 107.

It is mandatory to have at least one internal interface configured on the NSX Edge to exchange keepalives between active and standby units. Deleting the last internal interface would break this HA model.



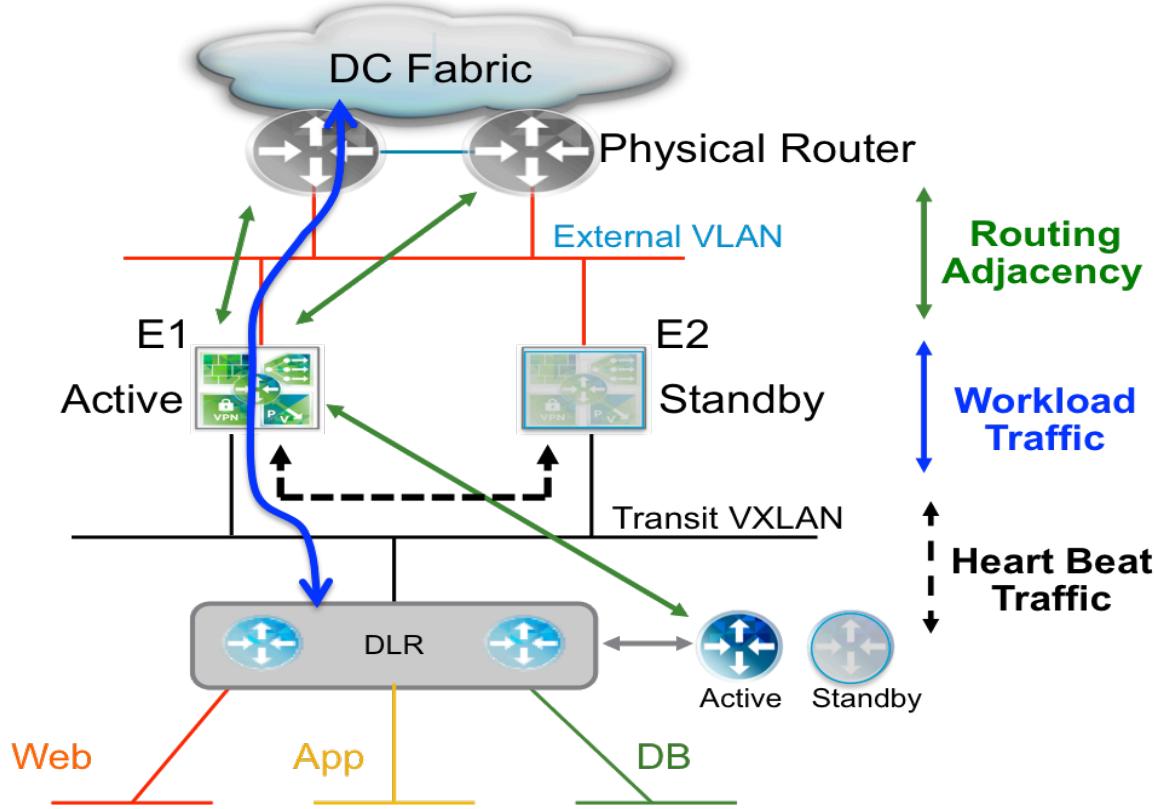
**Figure 109 – NSX Edge Active/Standby Deployment**

NSX Edge HA is based on the behavior highlighted in Figure 108.

- NSX Manager deploys the pair of NSX Edges on different hosts; anti-affinity rules are automatic.
- Heartbeat keepalives are exchanged every second between the active and standby edge instances to monitor each other's health status. The keepalives are L2 probes sent over an internal port-group. These leverage the first vNIC deployed on the NSX Edge unless a different interface has been explicitly selected. A VXLAN L2 segment can be used to exchange the keepalives, so it may happen over routed physical network infrastructure. The VLAN-backed port-group mandates the vNICs of the NSX Edges to be L2 adjacent (i.e., connected to the same VLAN), whereas the VXLAN-backed prot-group provides more topological flexibility and is recommended for deployment.
- If the active Edge fails, either due to ESXi server fails or user intervention (e.g., reload, shutdown), at the expiration of a “Declare Dead Time” timer, the standby takes over the active duties.
- The default value for this timer is 15 seconds. The recovery time can be improved by reducing the detection time through the UI or API. The minimum value that is supported is 6 seconds. Care must be taken to balance the deployment consideration for types of services and amount of state requiring syncing to standby VM vs. detecting false positives. Setting the timer to 9 seconds is a safe best practice.
- When the previously active Edge VM recovers on third ESXi host, the existing active Edge VM remains active.

The messages exchanged on the internal port-group between the active and standby NSX Edges are also used to sync up state information required for the logical services (e.g., forwarding information base (FIB), NAT, firewall connection state) to providing this service statefulness.

Figure 110 highlights how the active NSX Edge is active both from control and data plane perspectives. The routing adjacencies are formed with the physical router on a common “external VLAN” segment and also the DLR control VM on a common “transit VXLAN” segment. Traffic between logical segments connected to the DLR and the physical infrastructure always flows only through the active NSX Edge appliance.



**Figure 110 - NSX Edge Active/Standby HA Model**

In active-standby design, two uplinks can be enabled from the Edge over which two routing adjacencies establish to two physical routers. This provides two ECMP paths from an active Edge. This capability improves overall resiliency and is available from NSX release 6.1 onward. In this instance it is recommended to follow the two external VLAN design guidance similar to an ECMP-based design.

If the active NSX Edge fails (e.g., ESXi host failure), both control and data planes must be activated on the standby unit that takes over the active duties. Figure 111 displays traffic flows in this failed state.

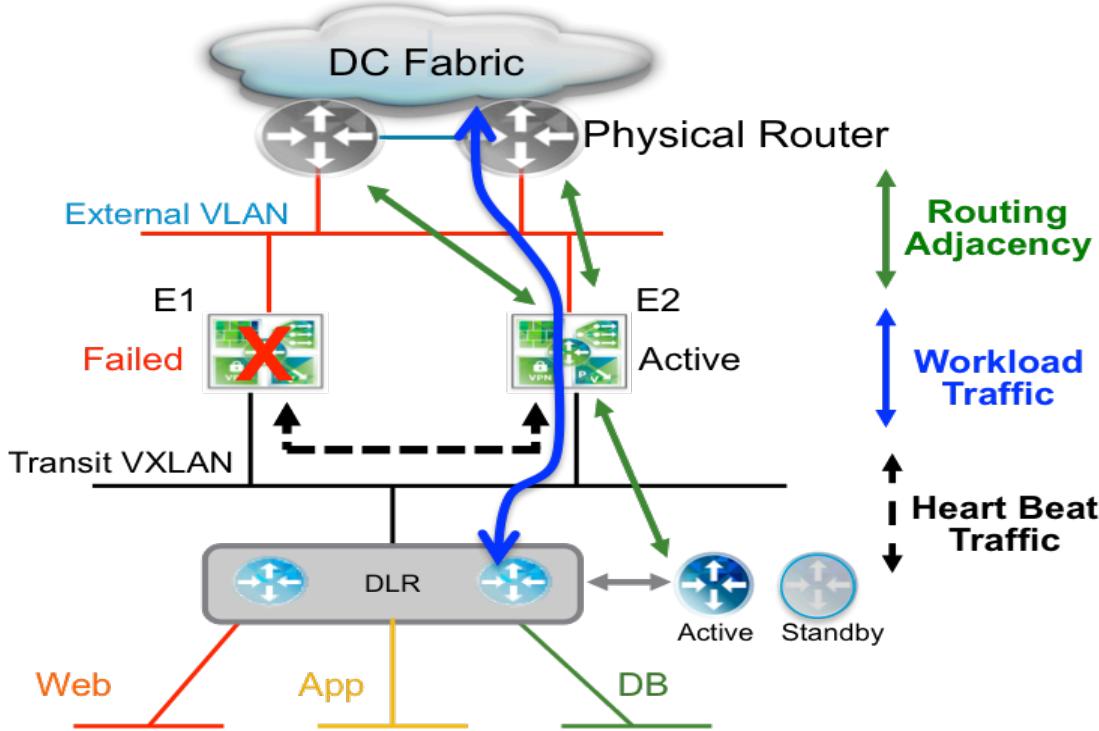


Figure 111 - Traffic Recovery after Active NSX Edge Failure

The following are some important design and deployment considerations relative to the behavior of this HA model:

- The standby NSX Edge leverages the expiration of a specific “Declare Dead Time” timer to detect the failure of its active peer. This timer is configured by default to 15 seconds and can be tuned down to a minimum value of 6 seconds via the UI or an API call. It is the main factor influencing the traffic outage experienced with this HA model. For a typical deployment the heartbeat value should not be set lower than 9 seconds to reduce the possibility of false positive.
- Once the standby is activated, it starts all the services that were running on the failed Edge. While the services are restarting, traffic will still be forwarded leveraging the information in the NSX Edge forwarding table that was kept in sync between the active and the standby Edge units. The same applies to the other logical services, since the state is synchronized and available also for FW, LB, NAT, etc.
- In order for north-south communication to be successful, it is required that the DLR on the south side of the NSX Edge and the physical router on the north side start sending traffic to the newly activated Edge.

There are a number of interactions and features involved in graceful acceptance of a newly active Edge:

- At the control plane/routing protocol level, the DLR active control VM and the physical router should remain unaware of the fact that a switchover has happened, maintaining the previously established routing adjacencies.
- Once the newly activated NSX Edge has restarted its routing control plane, it is ready to start sending hellos to the DLR Control VM and to the physical router. This is where the Graceful-Restart (GR) capabilities of the NSX Edge come into play. With GR, the NSX Edge can refresh adjacency with the physical router and the DLR Control VM while requesting them to continue using the old adjacencies. Without GR, these adjacencies would be brought down and renegotiated on reception of the first hello from the Edge and this would ultimately lead to a secondary traffic outage. This can be achieved by setting the hold-time timers to a sufficiently long value to ensure that the newly activated NSX Edge can restart the routing protocol before the hold-time on the DLR and the physical router expires. If the hold-time expires, the DLR and the physical router will time out the routing adjacencies, removing from the forwarding tables the routing information learned from the NSX Edge. The recommendation is to configure OSPF hold and BGP dead timers to at least 120 seconds between devices. The hello timer should be adjusted to a reasonable value, which is typically specific to a routing protocol. Only OSPF timers typically need modification since BGP default values are adequate, as shown in Table 8. The timers need to be matched at the DLR control VM and physical routers.

Routing Protocol Timers	Default Timers in Seconds	Recommended Timers Seconds
<b>OSPF Hello Interval</b>	10	30
<b>OSPF Dead Interval</b>	40	120
<b>BGP Keepalive</b>	60	60
<b>BGP Hold Down</b>	180	180

Table 8 – Protocol Timer Default and Recommendation for Active-Standby Edge

The timers are not a determining factor for the primary traffic outage. The recommended values are set quite high to cover the worst case scenario where restarting an NSX Edge may extend the time required to activate the network services.

At the data-plane level, the DLR kernel modules and the physical router keep sending traffic to the same next-hop IP address of the NSX Edge. However the MAC address used by the newly activated NSX Edge is different from the one that was used by the failed unit. In order to avoid traffic black holing, the new NSX Edge must send gratuitous ARP requests (GARPs) on both the external VLAN and the transit VXLAN segments to update the MAC to IP mapping on those devices.

In addition to the active-standby HA functionality, it is recommended to enable vSphere HA to protect the edge cluster where the NSX Edge services gateway VM resides. It is required to deploy at least three ESXi hosts in the edge cluster to ensure that a failed NSX Edge can be restored to a different ESXi host accordingly to the anti-affinity rule and become the new standby unit.

Finally, it is important to consider the impact of DLR active control VM failure to the convergence of workload traffic:

- The DLR Control VM leverages the same active-standby HA model as described for the NSX Edge Services Gateway. The main difference is that the control VM only runs the DLR control plane, while the data-plane is fully distributed in the kernel of the ESXi host part of the NSX domain.
- The DLR control-plane active-standby VM also runs a heartbeat timer similar to Edge services VM. This timer is typically not a determining factor in traffic recovery with the exception of bridging traffic recovery. Additional details on this caveat can be found in the bridging section
- The hold/dead timer setting (e.g., OSPF-120, BGP-180) for the routing timers required between the DLR and the NSX Edge to deal with the NSX Edge failure scenario previously described also allows handling the control VM failure case, leading to a zero seconds outage. While the standby control VM detects the failure of its active peer and restarts the routing control plane
- Traffic in the south-to-north direction continues to flow since the forwarding tables in the kernel of the ESXi hosts remain programmed with the original routing information received from the Edge.
- Traffic in the north-to-south direction keeps flowing since the NSX Edge does not time out the routing adjacency with the DLR Control VM established to the “Protocol Address” IP address. It continues sending data plane traffic to the forwarding address IP identifying the DLR data plane component. The use of a different IP address for control and data plane operations implies also that there is no need to generate GARP requests from the newly activated control VM.
- Once the control VM is ready to start sending routing protocol hellos, the graceful-restart functionality on the DLR side comes into the picture to ensure that the NSX Edge can maintain adjacency and continue forwarding the traffic.

### 5.3.7.2 ECMP Edge

NSX software release 6.1 introduces support for ECMP mode, which offers high bandwidth and faster convergence, making it the recommended deployment mode for most enterprise north-south connectivity.

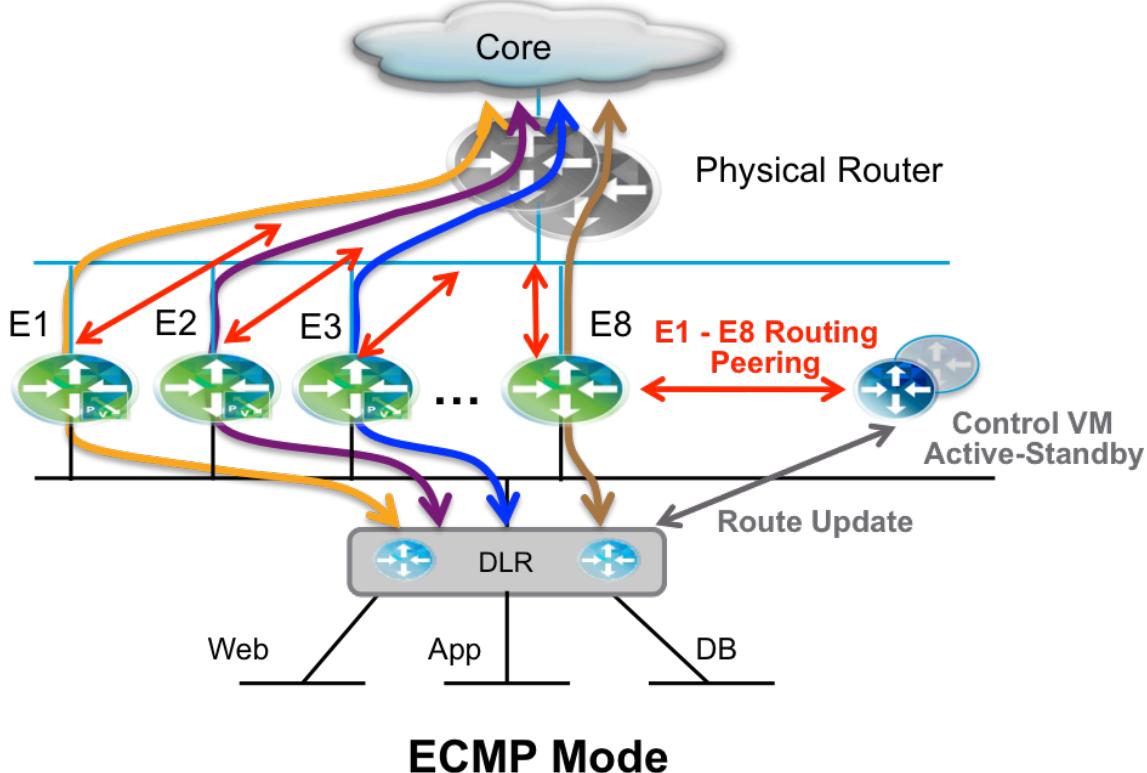


Figure 112 - NSX Edge ECMP HA Model

In the ECMP model, the DLR and the NSX Edge functionalities have been improved to support up to 8 equal cost paths in their forwarding tables. Focusing for the moment on the ECMP capabilities of the DLR, up to 8 active NSX Edges can be deployed at the same time and all available control and data planes will be fully utilized, as shown in Figure 112.

This HA model provides two main advantages:

- An increased available bandwidth for north-south communication, up to 80 Gbps per tenant.
- A reduced traffic outage in terms of % of affected flows for NSX Edge failure scenarios.

As shown in Figure 111, traffic flows are very likely to follow asymmetric paths, where different NSX Edge VM handle the north-to-south and south-to-north legs of the same flow. The DLR distributes south-to-north traffic flows across the various equal cost paths based on hashing of the source and destination IP addresses of the original packet sourced by the workload in logical space. The process used by the physical router to distribute north-to-south flows depends on the specific HW capabilities of that device.

In this mode, the NSX supports aggressive tuning of the routing hello/hold timers, down to the minimum supported values of 1 second hello and 3 seconds of dead timers on the NSX Edge and the DLR control VM. To achieve lower traffic loss, this is required because the routing protocol hold-time is now the main factor influencing the severity of the traffic outage. When E1 fails, the physical router and the DLR continue sending traffic to the failed unit until the expiration of the hold-time timer. At that point, they bring down the adjacency and update their forwarding table. The flows that were hashed through E1 will now use other ECMP edge nodes as next-hop. The similar detection and re-hash of the flows occurs for DLR to remaining ECMP nodes. This failover is shown in Figure 113.

	Default Seconds	Supported up to Seconds
OSPF Hello Interval	10	1
OSPF Dead Interval	40	3
BGP Keepalive	60	1
BGP Hold Down	180	3

Table 9 – Protocol Timer Default and Minimum Configuration Support for ECMP Edge

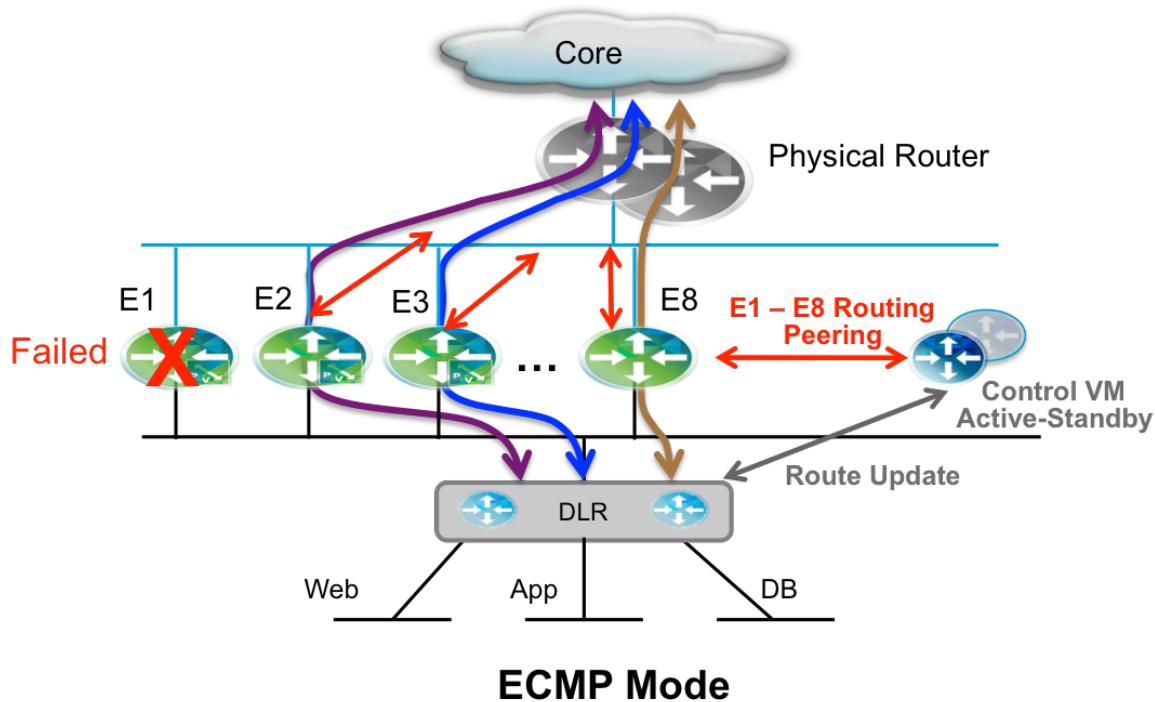


Figure 113 - Traffic Recovery after ECMP NSX Edge Failure

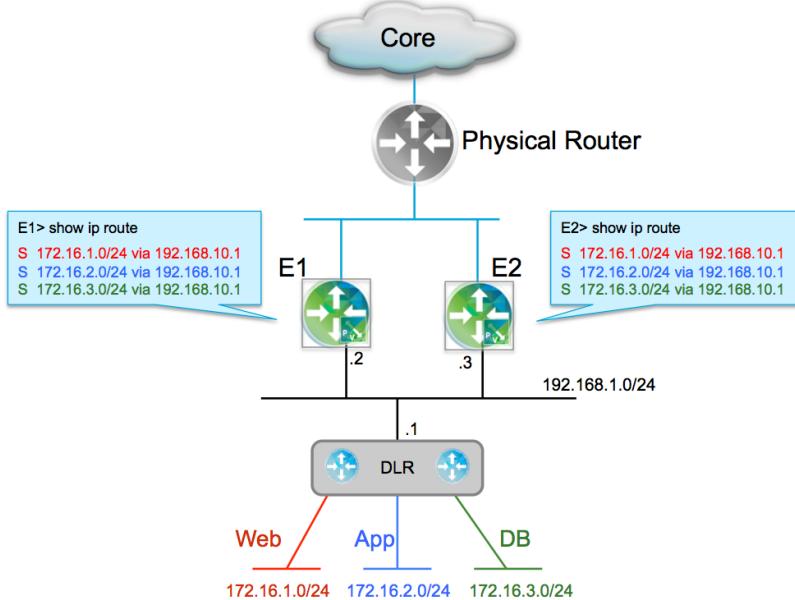
Some deployment considerations specific to this ECMP HA model:

- To maintain minimum availability, at least two ECMP edge VMs are required. Both VMs should not be in the same host to avoid a total loss of the connectivity. The anti-affinity rules are not automatically enforced and thus needs to be implemented.
- The length of the outage is determined by the speed of the physical router and the DLR control VM timing out the adjacency to the failed Edge VM. It is possible to expedite this process by aggressively tuning the hello/hold-time timers as low as 1/3<sup>rd</sup> of a second.
- The failure of one NSX Edge now affects only a subset of the north-south flows – the ones that were handled by the failed unit. This is an important factor contributing to an overall improved recovery functionality with the ECMP model.
- When deploying multiple active NSX Edges, the stateful services cannot be deployed with ECMP mode. It is recommended to leverage DFW for security protection and dedicated service Edge VM (for one-arm load balancer and VPN) or two tier Edge design to cover both scale and services requirements of workload.
- The diagram in Figure 112 showed a simplified view of the physical network infrastructure north of the NSX Edge Gateways – active-standby and ECMP. It is expected that a redundant pair of physical routers would always be deployed in a production environment. The section “[Edge VM Connectivity and Availability Design](#)” addresses best practices of connecting the NSX Edges to those routers from a logical perspective.
- The vSphere HA and DRS are important to maintain full availability of bandwidth in the case of a host failure.

#### 5.3.7.3 Control VM Failure & Recovery

The aggressive setting of routing protocol timers as applied to ECMP mode for faster recovery has an important implication when dealing with the specific failure scenario of the active control VM. This failure would now cause ECMP Edges to bring down the routing adjacencies previously established with the control VM in less than 3 seconds. This means that ECMP Edges flush their forwarding tables, removing all the prefixes originally learned from the DLR. This would cause the north-south communications to stop.

To mitigate this failure of traffic, static route with a higher administrative distance than the dynamic routing protocol used between ESG and DLR are needed. This configuration is shown in Figure 113.



**Figure 114 - Configuring Static Routes on the Active NSX Edges**

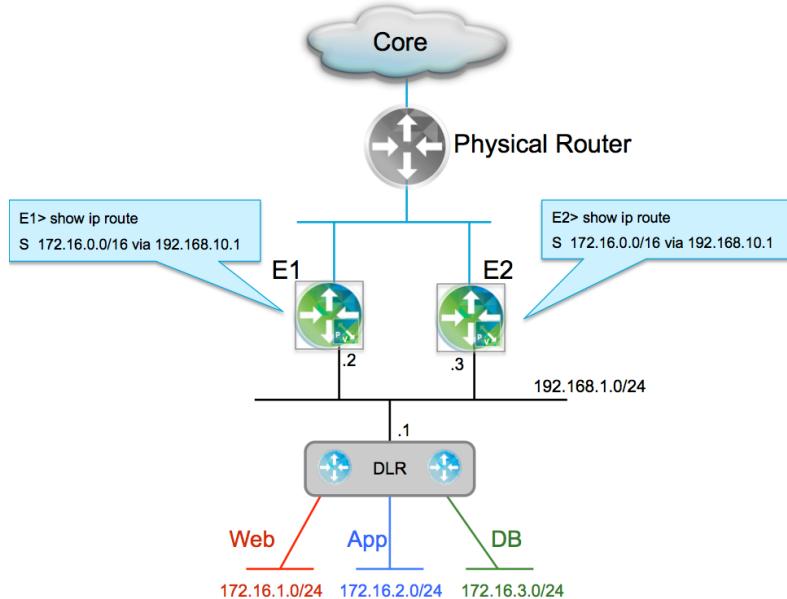
With this design, when the adjacencies time out and the dynamically learned routes associated with the Logical Switches are removed, traffic from north to south will continue to flow based on static routing information. For this to be possible, the NSX Edges must also redistribute the static routes into their routing protocol of choice to advertise them toward the physical routers. This configuration optimization is covered with routing protocol choices in “[Routing Capabilities in NSX](#)” section.

There is no requirement to configure static routing information on the DLR side. If the active control VM fails and the standby takes over and restarts the routing services, the routing information in its forwarding table remains unchanged. It is independent from the fact that the adjacencies between the DLR control VM and the Edge ECMP VMs are lost. All the traffic directed to the northbound IP prefixes will continue being hashed across the equal cost paths available via the NSX Edges. Once the control VM restarts its routing services, the adjacencies with the NSX Edges will be reformed and IP prefixes will be dynamically exchanged with them. New routing information received from the Edges, if present, will be pushed to the controller and into to the ESXi host kernels.

Despite the fact that static routes are configured on the NSX Edge devices to handle the Control VM failure scenarios, it is still critical to use a routing protocol between the Edges and the DLR Control VM. This is required as the control VM leverages the exchange of routing hellos to detect the failure of an Edge device, avoiding the traffic black-holing that would occur when deploying static routes on the DLR side.

The task of installing static routes on the Edge can be automated. This ensures that every time a new logical switch is created, a corresponding static route is added on both NSX Edge VMs, and the route is subsequently removed if the logical switch is deleted or disconnected from the DLR. In deployment scenarios

where the IP addressing of the logical networks can be properly planned, a summarized static route can be used on the Edge gateways to completely cover the logical address space.



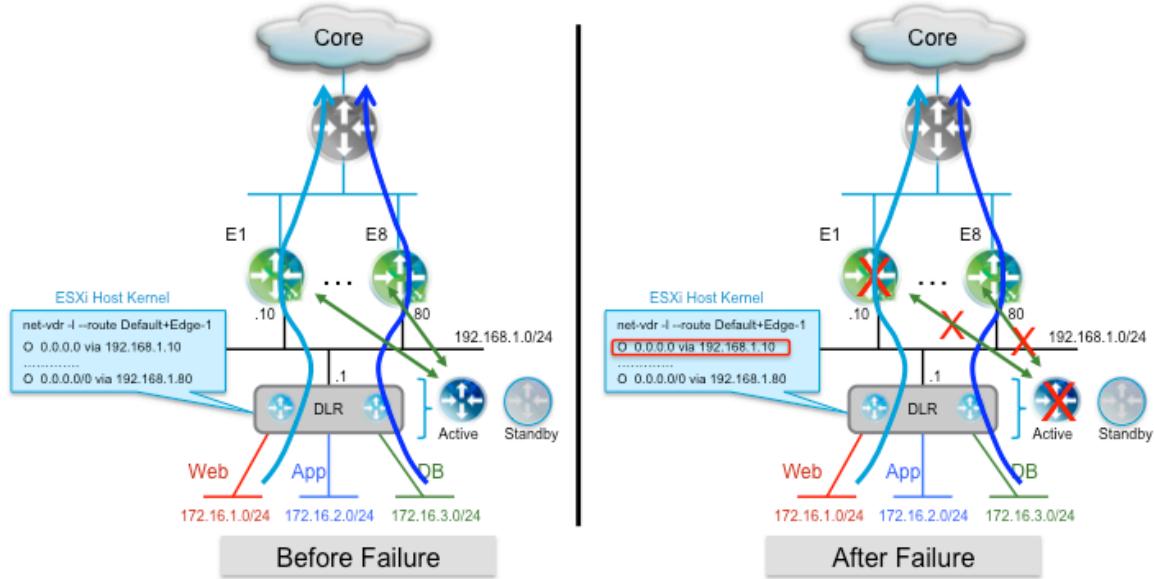
**Figure 115 - Use of a Summarized Static Route**

This configuration, shown in Figure 114, simplifies the deployment from an operational point of view. It does not require modifying the static route every time a new Logical Switch is added or removed, thus reduces the route churn.

It is important to emphasize that the static route is only required in ECMP mode and if the aggressive protocol timers are used for a faster recovery. The active-standby Edge VM deployment does not require static routes since the protocol timers are long enough (as required to recover standby) to obviate the need. In either case the control VM active-standby HA timers has no impact on the recovery.

### Avoiding Dual Failure of Edge and Control VM

There is an additional important failure scenario for NSX Edge devices in ECMP mode; the failure of the active control VM caused by the failure of the ESXi host where the control VM is deployed together with one of the NSX Edge router. This event is pictured in Figure 115.



**Figure 116 - Concurrent Failure of DLR Control VM and NSX Edge**

In this failure event, the control VM cannot detect the failure of the Edge since it has also failed and the standby control VM is still being activated. As a consequence, the routing information in the ESXi host's kernel still shows a valid path for south-to-north communication via the failed Edge device. This is Edge VM E1 (192.168.1.10) in Figure 115. This state will exist until the standby control VM is activated, it restarts its routing services, and reestablishes adjacencies with the Edges. This will lead to a long outage for all the traffic flows that are sent via the failed Edge unit.

In order to address this outage, ensure that the control VMs are never deployed on the same ESXi server hosting a tenant NSX Edge device. This can be achieved in different ways:

- Increasing the number of ESXi hosts available in the Edge cluster and configuring anti-affinity rules between the control VMs and the NSX Edge VMs.
- Where there is not enough compute capacity on the Edge cluster, the control VMs can instead be deployed as part of the compute clusters. The control plane communication between the control VMs and the NSX Edges happens on a transit VXLAN segment, so deploying the control VMs in the compute cluster does not mandate any specific requirement in terms of spanning of L2 VLANs across racks connected to the physical network infrastructure; only IP connectivity is required between those NSX components.
- Deploying the Control VMs in the management cluster. This is only viable in small design where clusters are collapsed for efficient host utilization and thus it requires deploying NSX. This entails installing the VIBs and configuring VXLAN).

Each Edge model and its interaction with control VM along with various high availability guidelines is summarized in Table 10.

Edge VM Model	Active-Standby Edge	ECMP	Control VM
Minimum Host	2	=> 2 Depending on BW Requirements	2
Anti-affinity & DRS	Automatic	Manual with DRS group based on VM density per Host	Automatic Active VM should not be with ECMP Edge VM
Routing Protocol Timer Tuning	Required based on protocol - 30/120 for OSPF 60/180 for BGP	Not required but can be tuned up to 1/3	Match based on Edge VM model and tuning
HB Timers Tuning	Default 15 Tune based on requirement up to 9 second	Not Applicable	Default 15 Rarely need tuning
Size	Depending on performance and stateful services need	Depending on BW need	Not applicable

Table 10 – Summary of Edge Models and Control VM Configuration Choices

#### 5.3.7.4 Edge VM Connectivity and Availability Design

The design consideration that is applicable to both modes of Edge VM are as follows:

- VLAN design for peering with physical routers
- Availability, oversubscription, and Edge VM density consideration

## VLAN Design for Peering with Physical Routers

An Edge VM needs a VLAN facing physical routers over which the routing protocol adjacency is established. The central design consideration here is to keep connectivity simple, predictable, and fast-converging during failure. Simplicity keeps the VLAN configuration loop free. Predictable means keeping the routing protocol peering mapping to only distinctly 1:1 path. Fast convergence ensures no other underlying protocol interaction impacts the detection and recovery of forwarding.

The simple rule to follow is to map each element to unique paths as follows:

**Edge Uplink = VDS uplink = Host NIC Uplink = Router = Adjacency**

This mapping is mirrored for redundancy from Edge uplink to physical routers. This convention is primarily adapted for dual 10 Gbps NIC, however it can be extended for the host carrying more than 2 x 10 Gbps links. A complete peering diagram is provided in Figure 116.

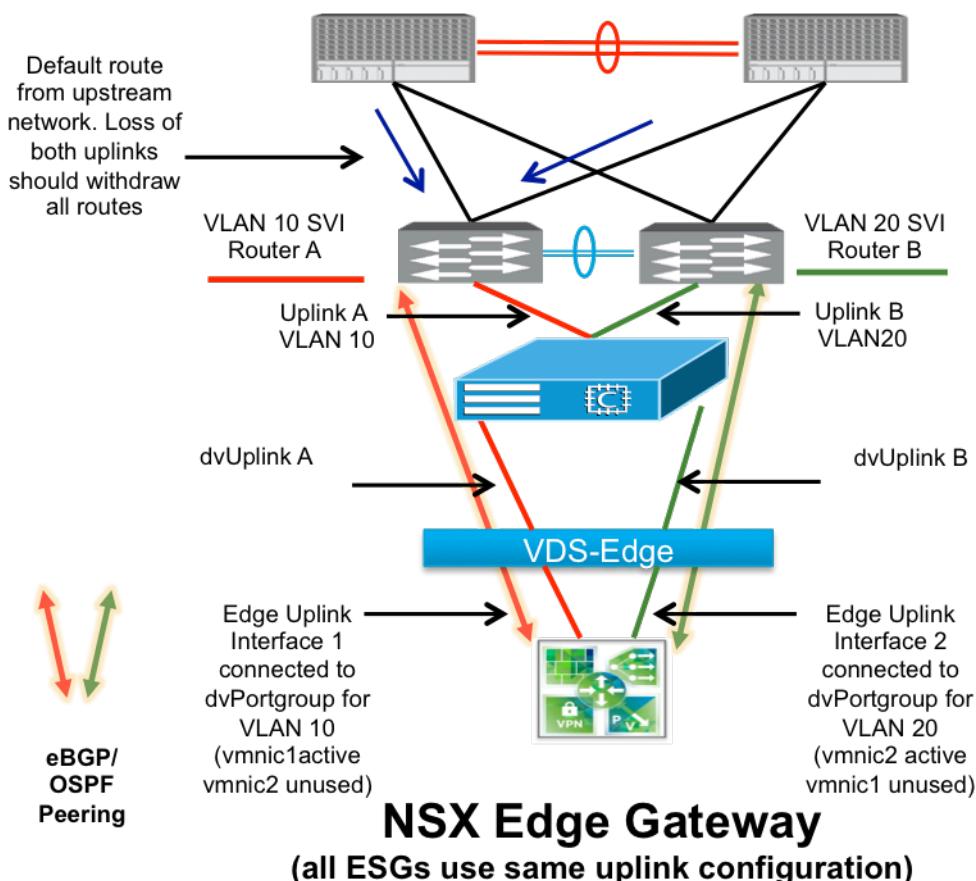


Figure 117: Edge VM to Physical Routers Mapping for Routing Connectivity

The recommended design is to map the number of logical uplinks to the number of VDS uplinks available on the ESXi servers hosting the NSX Edges. Since an NSX Edge logical uplink is connected to a VLAN-backed port-group, it is required

to use two external VLAN segments to connect the physical routers and establish routing protocol adjacencies. Each external VLAN should be carried on one ESXi uplink. In Figure 116, VLAN10 is carried on the uplink toward router A and VLAN20 on the uplink toward router B. The switch virtual interfaces (SVIs) for VLANs 10 and 20 exist only in router A and router B respectively. They do not span across the two routers, thus avoiding spanning tree related failure and convergence. This simplifies the design and is predictable so that under normal circumstances both ESXi uplinks can be concurrently utilized to send and receive north-south traffic, even without the creation of a port-channel between the ESXi host and the ToR devices. With this model, a physical failure of an ESXi NIC would correspond to a logical uplink failure for the NSX Edge running inside that host. In that case the Edge would continue sending and receiving traffic leveraging the second logical uplink (i.e., the second physical ESXi NIC interface).

In Figure 116 the connectivity of all the deployed active NSX Edge gateways E1-E8 are running on ESXi hosts connected to the Top-of-Rack switches via two VLANs mapped uniquely to R1 and R2.

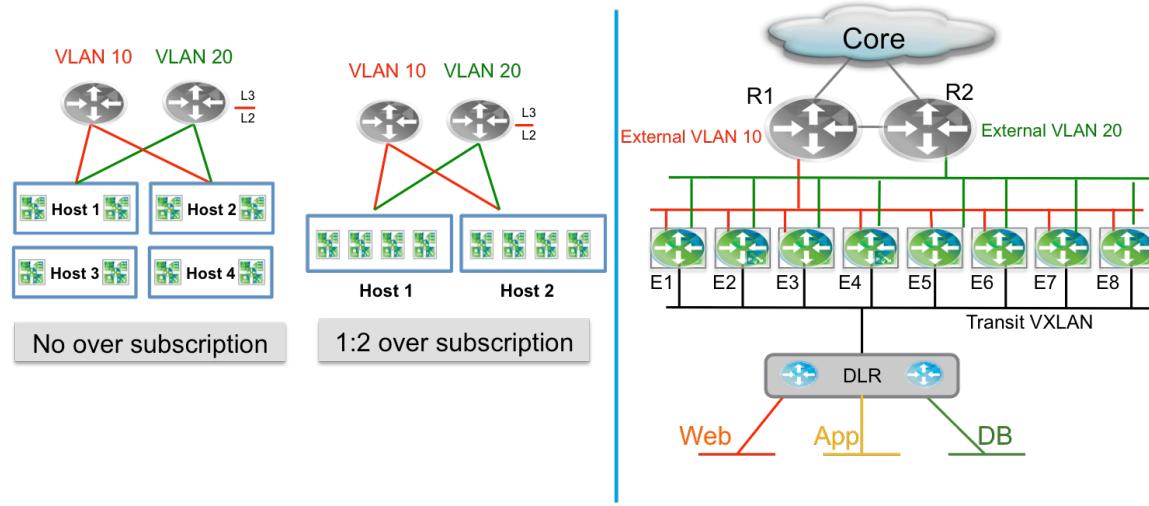


Figure 118: Connecting ECMP NSX Edges to Redundant Physical Routers

### Availability, Oversubscriptions and Edge VM Density Considerations

The number of Edges and type of Edge deployed varies based workload requirements. The NSX model is flexible in supporting a single active-standby pair for small designs to the ECMP multi-tenant model where each DLR domain (i.e., tenant) can scale up to 80 Gbps. Thus edge scaling is elastic and meets bandwidth need from small to large datacenter need. The right side of Figure 117 depicts the 80 Gbps (eight Edge VMs) connectivity for a sample tenants and left side of the picture offers a choice in oversubscription based design.

While designing Edge cluster, architect often faces a challenge in estimating bandwidth and availability of Edge VM. Understanding how much north-south bandwidth is required for tenants or workload is first step in adequate design.

Often north-south bandwidth is a fraction of (10 to 15 percent) east-west bandwidth. Thus most workload does not require 80 Gbps from get go.

Second factor is the understanding of oversubscription that exists in critical points in network. A proper design must consider the oversubscription that is present in every layer between end-user to workload (i.e., north-south) and between workload end-point (i.e., east-west). As described in physical design section, cross-sectional bandwidth of the physical fabric, number of uplinks from access-layer switch, and host density per access-layer switch affects both east-west and north-south traffic.

The most critical oversubscription point for north-south traffic, which is where Edge VM design is relevant, is connectivity to rest of the network either at the aggregation layer for multi-tier or at the border-leaf in spine-leaf topology in routed or proprietary fabrics.

In a typical aggregation design the bandwidth to the datacenter core ranges from 40 to 160 Gbps. For the leaf-spine design, the problem becomes more acute as the pair of border-leaf is the common exit point. The bandwidth from spine to border-leaf and bandwidth from border-leaf uplinks to rest of the data centers is centrally important. The number of ECMP Edge VMs (and thus the bandwidth) should be aligned with the border-leaf uplinks to core networks. In other words, in a non-oversubscribed design, even after providing a full line rate forwarding from the Edge to ToR as shown in Figure 117 (the two Edge VMs per host matching 2x10 Gbps) it is important to make sure that ToR to upstream bandwidth is adequate.

Adding further, that not all workload and tenants burst at the same time, one can consider an oversubscribed model with four Edge VMs per host which has 20 (2x10) Gbps bandwidth from host to ToR. Thus offering 50% oversubscriptions with an assumption to match the number of ToR uplinks, otherwise a further increase in oversubscription will result.

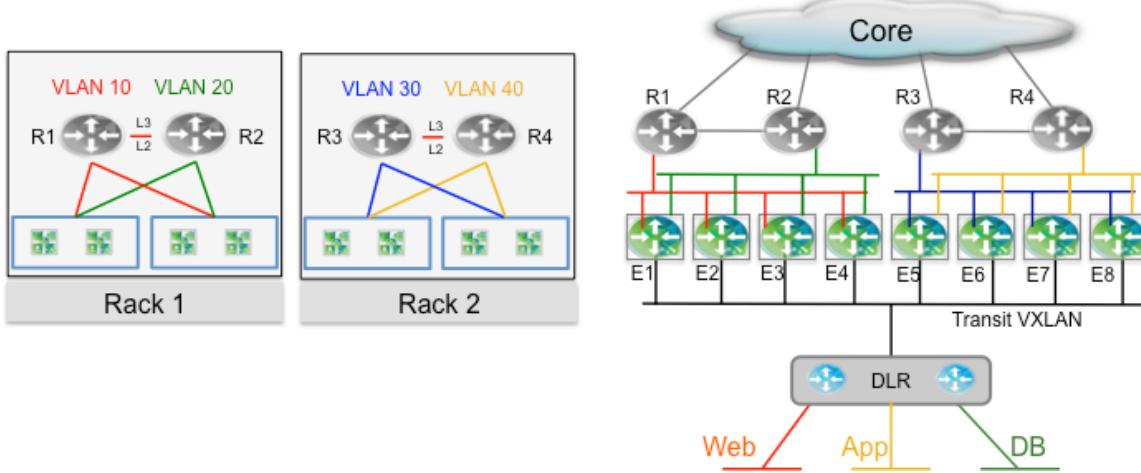
Alternatively, consider four 10 Gbps NICs from the host to accommodate higher Edge VMs per hosts to match the uplink bandwidth. The availability consideration dictates splitting the Edge VMs to more than one host so that a host failure does not reduce the available bandwidth to zero until the Edge VM recovers to another host.

It is recommended to use hosts with higher core densities to support the required Edge VM sizing (2 vCPU to 4 vCPU). Higher clock speed should also be considered for VXLAN offload while NIC choices made based on the need for higher bandwidth. Further design consideration is discussed in "[DC Cluster Configurations & Sizing with NSX](#)".

## Multi Rack Connectivity Option

In order to build a resilient design capable of tolerating the complete loss of an edge rack, it is a design choice to deploy two sets of four Edge VMs in two separate edge racks. This mandates the extension the external VLANs between the two edge racks. This can be achieved by leveraging an external L2 physical

network. The need to span these external VLANs between the edge racks can be eliminated by adopting the second deployment option, shown in Figure 119.



**Figure 119 - Connecting ECMP NSX Edges to Redundant Physical Routers (Option 2)**

The relevant design points in this scenario are:

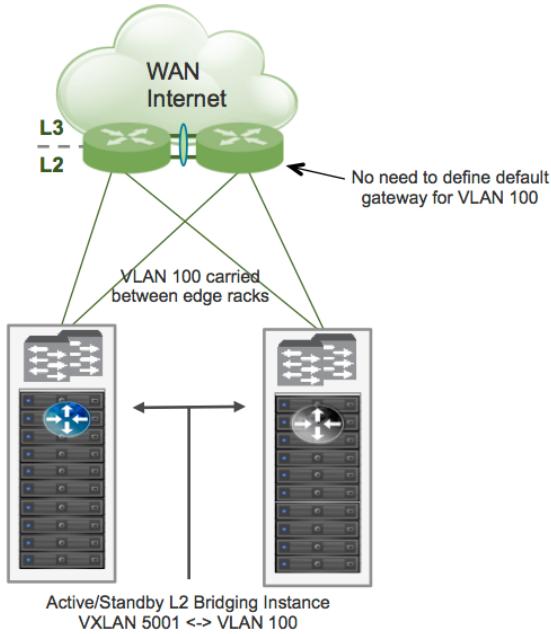
- Two separate VLANs are defined in each edge rack to connect the local NSX Edges to the physical routers.
- The span of those VLANs is limited to each rack. It is now possible to deploy two independent pairs of physical routers and position them as Top-of-Rack devices instead of connecting the edge racks to an external L2 infrastructure. NSX Edges deployed in edge rack 1 establish routing peering with R1 and R2, while NSX Edges deployed in edge rack 2 peer with R3 and R4.
- The VLAN IDs associated with the peering segments in separate edge racks could be the same, however this is not recommended for troubleshooting as operational confusion may arise from the same VLAN IDs in different racks or in layer 2 design where distinct VLANs termination at the aggregation is desired.
- As VLANs will not be extended between racks, the recommendation is to ensure that NSX Edge VMs can be dynamically moved (e.g., vSphere HA) between ESXi hosts belonging to the same edge rack, but never across racks.

### 5.3.7.5 NSX Layer 2 Bridging Deployment Considerations

The NSX L2 bridging functionality is used to provide logical networking access to physical devices that are connected to VLANs defined in the physical infrastructure. An NSX L2 bridge must be able to terminate and decapsulate VXLAN traffic, originated by a virtual machine belonging to a given Logical Switch, and to bridge it on a specific VLAN.

The NSX bridging functionality is described in the “Unicast Traffic (Virtual to Physical Communication)” section. This section covers the design consideration of deploying L2 bridging functions.

The L2 bridging function for a given VXLAN-VLAN pair is always defined under the DLR configuration menu. This bridging instance is enabled on the ESXi host where the active control VM is running. This does not mean bridging traffic is passing through an active control VM; instead, it simply defines where this bridging instance needs to run. The bridge traffic is forwarded via a specially defined sync port in the kernel at the line rate



**Figure 120 - Active/Standby NSX L2 Bridging Instance**

As shown in Figure 120, all the L2 bridging instances defined on a given DLR would be active on the ESXi host in edge rack 1 where the corresponding DLR Active Control VM is running. It is possible to create multiple sets of bridging instances and associate them with different DLRs. This allows for spreading of the bridging load across different ESXi hosts.

Some important deployment considerations include:

- If two edge racks are deployed to increase the resiliency of the solution, it is recommended to connect the active and standby control VMs in those separate racks. This implies that the VLAN where VXLAN traffic is bridged must extend between the two edge racks to cover the scenario where the active control VM moves from edge rack 1 to edge rack 2. Using an external L2 physical infrastructure is a viable option to extend VLANs between the edge racks.
- The keepalive messages between the active and standby control VMs can be exchanged leveraging VXLAN (internal Edge interfaces), therefore it is not required to extend an additional VLAN for that purpose. The keepalive

message can be tuned down to improve bridging traffic recovery. This change must be performed via the API; it cannot be done in the GUI. The recommendation is not to tune below 9 seconds.

- It is recommended to connect the bare metal servers requiring connectivity to the logical networks to the edge racks. This is important to limit the extension of the bridged VLANs to only their associated racks and not to the other compute racks connected to different ToR switches.

When building the separate L2 physical infrastructure depicted in Figure 119, there is the option of deploying a dedicated set of racks for bare-metal servers, with their own ToR switches directly connected to the switched physical network.

#### 5.3.7.6 NSX Distributed Routing and Layer 2 Bridging Integration

Prior to the NSX 6.2 release, bridging could not be combined with distributed logical routing; it was not possible to enable optimized east-west forwarding and simultaneously bridge the same logical segment to external VLAN segment for migration or temporary connectivity. Figure 120 details the limitation in which a database segment could be utilizing distributed routing and thus had to use centralized NSX Edge for routing and bridging.

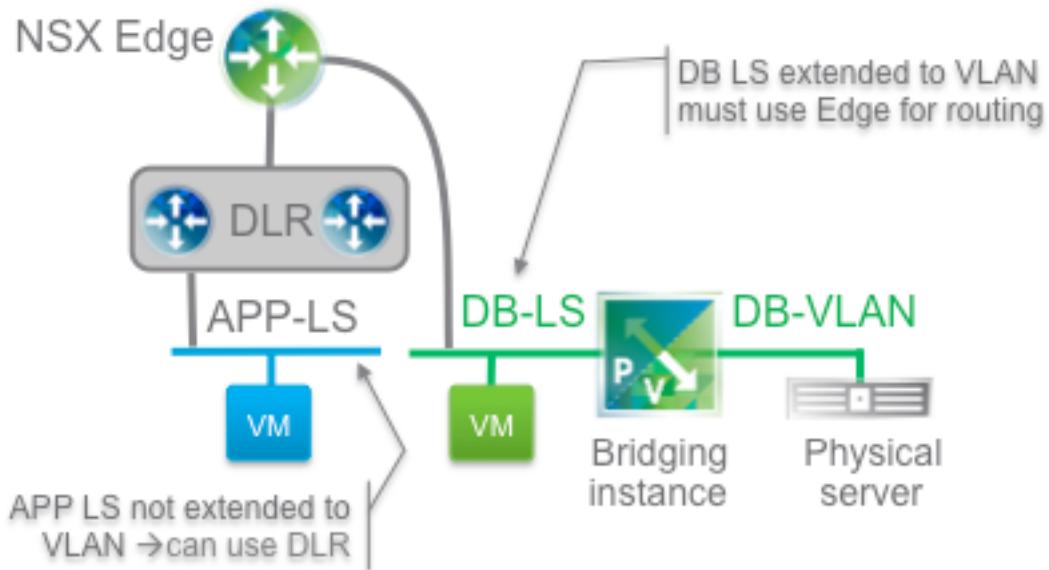
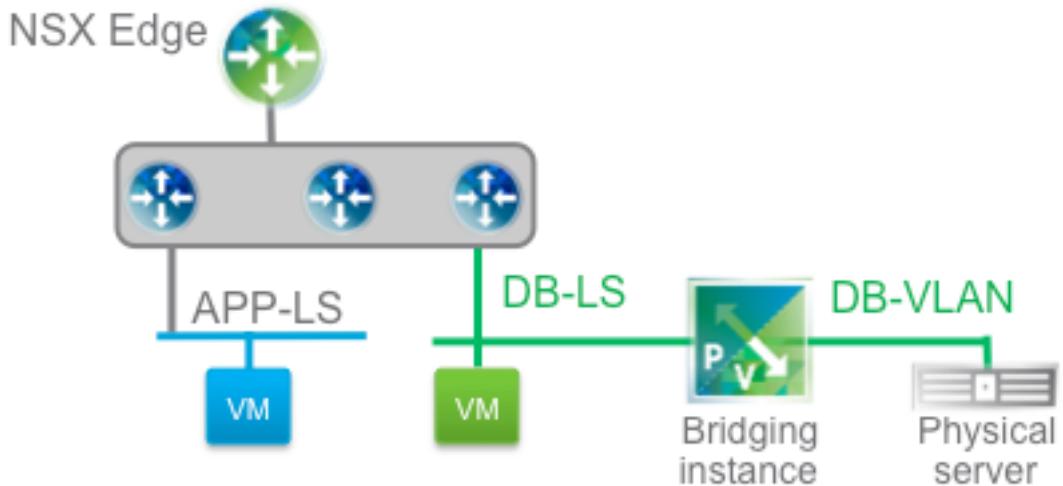


Figure 121 – Bridging and Distributed Routing in before NSX 6.2

With advances in NSX release 6.2, a given logical segment (i.e., switch) can now participate in distributed routing as well as bridging as shown in Figure 121.



**Figure 122 - Bridging and Distributed Routing in after NSX 6.2**

As a result, the traffic for the bridging segment (e.g., the database segment in Figure 121) does not need to flow through a centralized Edge VM; it can now switch to the other segment via distributed forwarding. The routing between segments is now completely local to the kernel. For a given logical switch can only be extended to a single VLAN with one active bridge instance.

### 5.3.8 DC Cluster Configurations & Sizing with NSX

NSX provides modularity which allows design to scale based on requirements. The requirement gathering is an important part of sizing and cluster design. The requirements consist of identifying workload types, size of the compute, network services, network bandwidth required. The design considerations and approaches for enabling NSX in a particular environments range from only few hosts to hundreds of hosts, reaching to the upper limit of vCenter scalability. The component sizing (i.e., small to extra-large Edge) and configuration flexibility in the platform allows adoption of NSX in across a wide scale of environments. These guidelines attempt to categorize the deployment into three broad size categories – small, medium, and large. The recommendations are provided as a broad; actual implementation choices will vary based upon specific requirements.

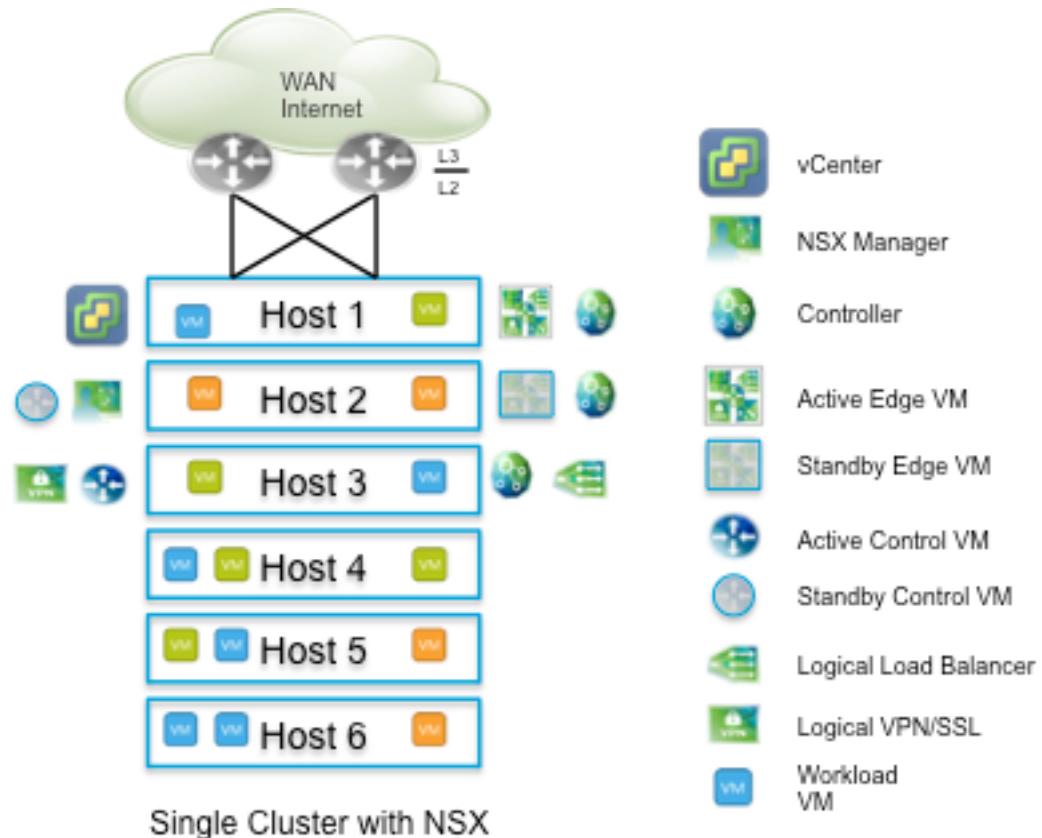
Common factors affecting the sizing and configuration are as follows:

- The number of hosts in deployment – small (e.g. 3 – 10), medium (10 – 100), or large (> 100). Growth requirements can be a significant factor in the design choice.
- Workload behavior and selection of NSX components mixed with regular workload.
- Multiple vCenter is not the requirements, though offers great flexibility and cross-VC mobility with NSX 6.2 and ESXi 6.x release
- NSX component placement restrictions depends on vCenter design, collapsed clustering options and other SLAs. Some essential attributes are:

- Controller must exist in a vCenter where the NSX manager's registers to. This means location of controllers should be in the management cluster when used with a single vCenter
- Controller should reside in an Edge cluster when a dedicated vCenter is used to manage the compute and edge resources. This is a multi vCenter deployment model
- Must consider Edge component placement and properties as described in [Edge Design & Deployment Considerations](#) as well the Edge vCPU requirements (see Figure 15 under section [NSX Edge Services Gateway section](#)).

## Small Design

A small-scale design can start with single cluster having few ESXi hosts, consisting of management, compute, and edge components along with the respective workloads. This is commonly called a “datacenter in a rack” design and consists of pair of network devices directly connected to WAN or Internet servicing few applications.



[Figure 123 - Single Cluster Small Datacenter Design](#)

Typical design considerations for single cluster small design are:

- The standard edition vSphere license is assumed which now supports VDS for NSX deployments only. See KB article:

---

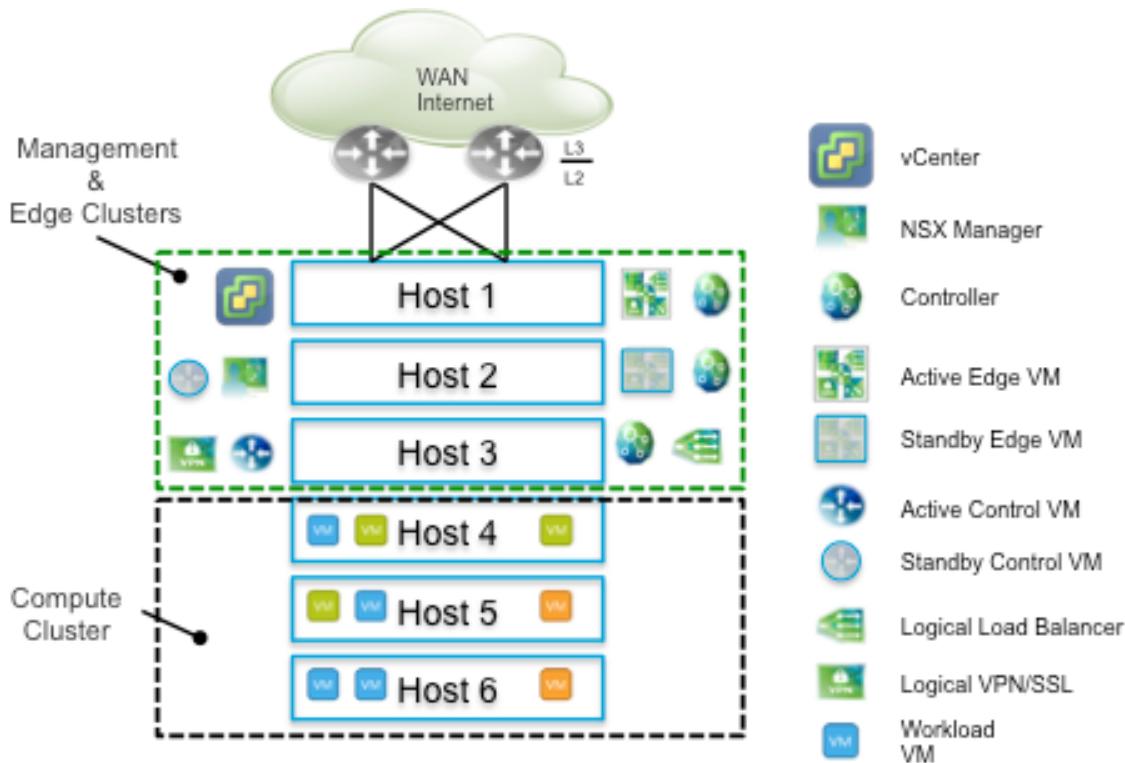
[http://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=2135310](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2135310)

---

- This is a single cluster design and thus all hosts are prepared for VXLAN. Management workload (vCenter and any other non-NSX components) must be excluded from DFW stateful firewall list to avoid loss of connectivity.
- The Edge gateway can serve holistic need of small business - integrated VPN, SSL client access and firewall services.
- The bandwidth requirement is low, typically less than 10 GB, therefore the NSX Edge VM can be deployed in active-standby mode. Using DRS anti-affinity rules, the Edge VM's should be automatically placed on two different hosts for availability. The large size (2 vCPU) of Edge VM is recommended, and it can grow to quad-large (4 vCPU) if line rate throughput is desired.
- Need of control VM can be avoided if the static routes are enabled between DLR and Edge VM, however if the growth is considered then dynamic routing is preferred.
- Resource reservation is a key to maintaining the SLA requirements of CPU reservation for NSX components, specifically for the Edge services gateway.
- If a growth is planned, expansion plans should separate the compute workload into separate cluster. This is applicable even if the compute cluster is in the same rack.

## Medium Design

Medium size design considers the possibility of future growth. It assumes more than one cluster, so a decision is required whether any clusters can be mixed while optimally utilizing underlying resources. Mixing edge and compute workloads require close monitoring of CPU resources since edge workload is more CPU centric while compute workload could be unbalanced and may require on-demand changes. Growth of compute is more likely, thus managing the resources to protect the edge workload gets complex. Mixing edge and compute clusters is not recommended, especially when compute workload growth is expected on same set of hosts.



**Figure 124 - Medium Datacenter Design**

Mixing either edge or compute with management requires the cluster be prepared for VXLAN. This may make expansion or decoupling of management component difficult if not impossible. As shown in Figure 124, edge and management components can be mixed together in a single cluster. Collapsing them with proper resource reservation can optimize the resources, while the size and scope could be still a single rack. This allows for compute growth. The design considerations for medium size deployment are as follows:

- The standard edition vSphere license is assumed which now supports VDS for NSX deployments only.
- The Edge gateway can serve holistic need of medium business – load balancer, integrated VPN, SSL client access, and firewall services
- Bandwidth requirements will vary though are typically not greater than 10GB. Edge VM can be active-standby automatically deployed with automatic anti-affinity in two different hosts for availability. The large size (2 vCPU) of Edge VM is recommended and can grow to quad-large (4 vCPU) if line rate throughput is desired.
- If throughput greater than 10 GB is desired, convert the active-standby gateway to ECMP mode with the same vCPU requirement, however stateful services cannot be enabled. Instead, enable the load balancer service or VPN on an Edge in the compute cluster and use DFW with micro-segmentation.

- Active DLR control VM should not be hosted where the active Edge services gateway is running for ECMP based design or reevaluate DRS and anti-affinity rules when Edge VM is converted from active-standby to ECPM mode. And consider tuning routing peer for faster convergence.
- Resource reservation is a key to maintain SLA of CPU reservation for NSX components, specifically the Edge services gateway.
- Recommendations for separate management, compute, and edge should be considered for the following conditions:
  - Future host growth potential.
  - Multi-site expansion.
  - Multiple vCenters managing distinct sets of virtualized workloads.
  - Use of Site Recovery Manager (SRM) for disaster recovery.

## **Large Scale Design**

Large-scale deployment has distinct characteristics that mandate separation of clusters, diversified rack/physical layout, and consideration that is beyond the scope of this document. In the context of NSX design impact, the following factors play critical role:

- Workload characteristics and variability
- Higher degree of on-demand compute.
- Compliance standards – Banking, HIPPA and PCI.
- Automation.
- Multiple vCenters managing production, development, and QA segments.
- Migration of workloads across multiple vCenters.
- Multi-site and disaster recovery as baseline requirements
- Multi 10GB traffic pattern for both east-west and north-south traffic.

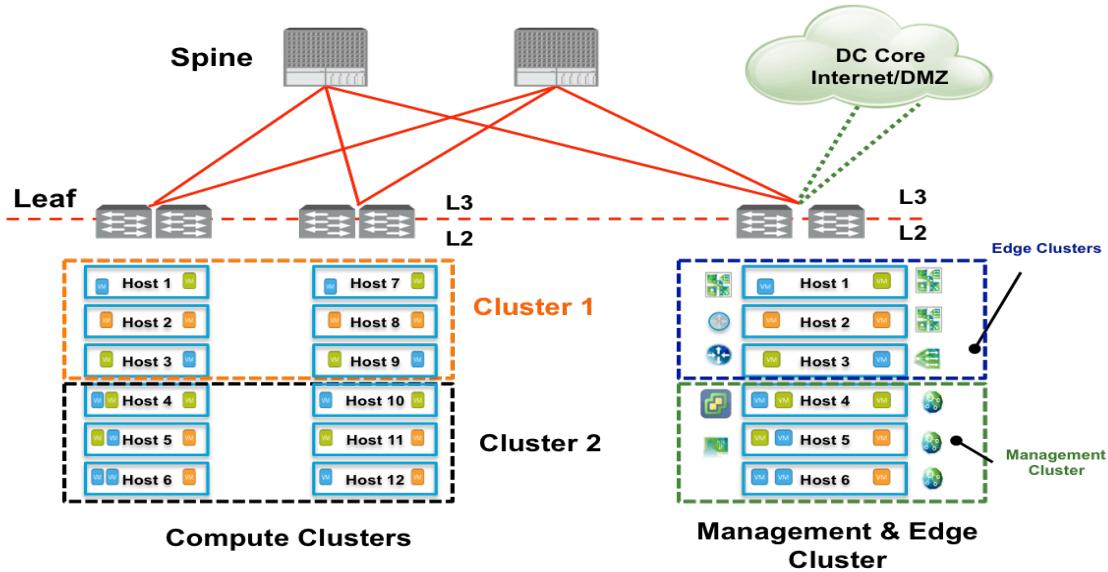
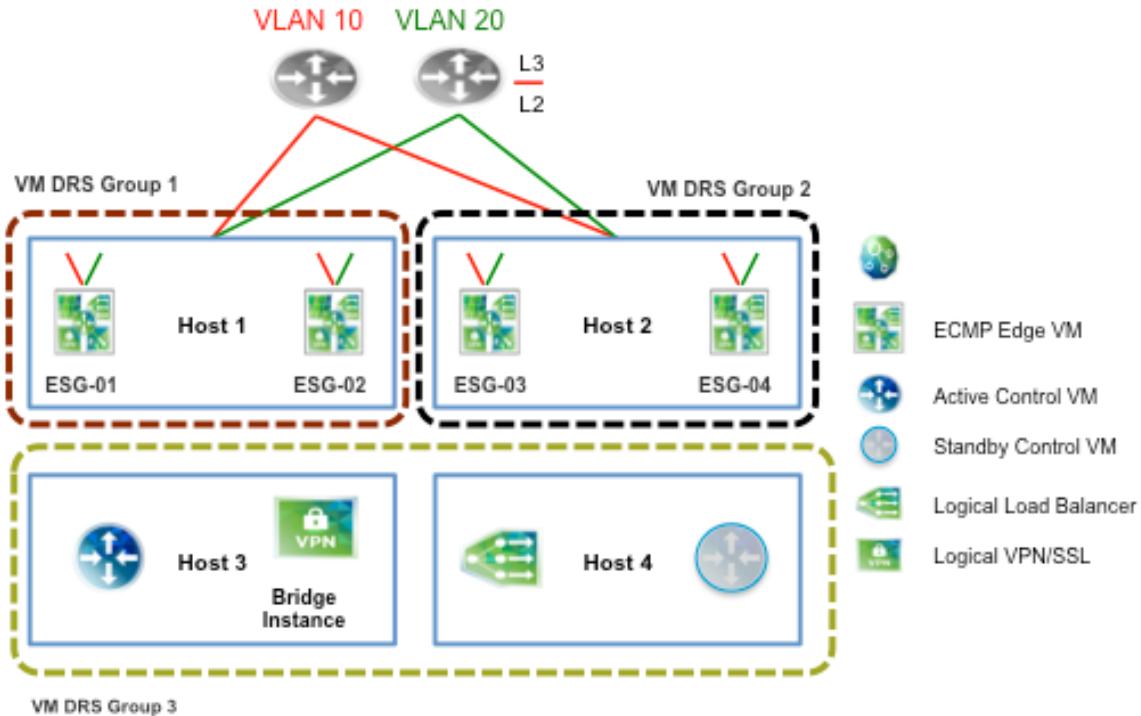


Figure 125 - Large Datacenter Design

The large-scale design requires further consideration of edge cluster design as follows:

- vSphere enterprise edition is assumed
- Enterprise class design assumes higher north-south bandwidth, typically greater than 10 Gbps, requiring ECMP-based Edge deployment.
- The bandwidth availability considerations are discussed in detail in “[Edge Design and Deployment Considerations](#)”. The design properties are narrated below with an assumption of understanding of that section.
  - A sample design assumes 40 Gbps north-south with growth up to 80 Gbps. Two ECMP Edge VMs per host providing line rate bandwidth of 20 Gbps per host. Thus total two hosts used for north-south traffic.
  - The active and standby control VMs are located on separate hosts with automatic anti-affinity rules enabled.
  - Need anti-affinity rule between host holding ECMP Edge VMs pair so that not all ECMP VMs do not end up on a single host. Enabled DRS protection between ECMP Edge VMs host and active control VM hosts to avoid traffic outage as described in [Avoiding Dual Failure Of Edge and Control VM](#).
  - The hosts containing standby control VM can be used as back up by either hosts containing ECMP VMs.
  - Thus minimum number of hosts recommended is four as shown in Figure 125 below



**Figure 126 – DRS and Protection Group for ECMP and Control VM**

- Additionally, Figure 125 depicts additional NSX services that can be deployed along with control VM. These services are optional and location may vary based on workload requirements, either near to logical switch or in the Edge cluster as a centralize service.
- Consider oversubscription ratio based on uplink bandwidth of the ToR where the Edge hosts connects.
- The edge cluster can be deployed in a single rack leveraging a pair of ToR switches or striped across two racks with a pair of ToR switches in each to survive a rack failure

### Host Specification Guideline for Edge Cluster

The following guideline should be followed when it comes to host CPU, memory and NIC for the Edge cluster:

- Always use uniform hosts specification for all hosts in Edge cluster to have consistent and predictable response.
- Higher CPU clock is preferred for attaining consistent ECMP Edge VMs performance
- Allow for growth in vCPU in case of Edge form factor upgrade from large to quad-large
- Though Edge VM does not require high memory, other workload in Edge cluster may need it, hence consider appropriate size.
- Highly recommended to use NIC that supports VXLAN TSO offload and RSS support for desired line rate performance.

## 5.4 Design Consideration for NSX Security Services

NSX provides a platform for security services to protect workloads in a datacenter. Security services can either be built in as part of the NSX product or integrated by 3<sup>rd</sup> party vendors. This section examines design scenarios and best practices that allow for a zero-trust model in the datacenter using the NSX security platform to deliver micro-segmentation. It will cover both brownfield and greenfield environments.

### Three Pillars of Micro-Segmentation

Micro-segmentation consists of three parts: network isolation, workload traffic segmentation, and advanced services application.

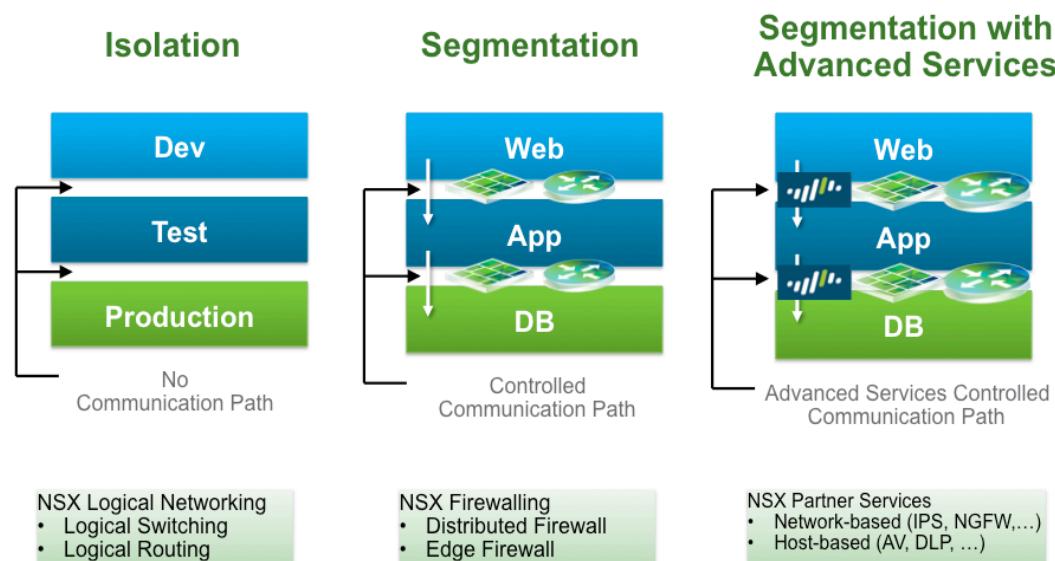


Figure 127 – Pillars of Micro-segmentations

Vendors that integrate with the NSX platform provide advanced services. These services are not available natively in NSX. They include:

- Network based security services (e.g., L7 firewall, IPS, IDS).
- Endpoint services (e.g., anti-virus, anti-malware, file integrity monitoring).
- Vulnerability management.
- Network based monitoring technologies (e.g., Gigamon) providing packet aggregation and brokering services.
- Enhanced RBAC provided by partners Hytrust (<http://www.hytrust.com/solutions/access-control-for-nsx/>)

Designing network isolation using physical network underlays and logical networking features was previously covered; this section will focus on segmentation and advanced services using capabilities discussed in [Introduction to Service Composer](#).

The ideal process for deploying segmentation and advanced services is as followed:

- Preparing the security fabric for built-in and partner vendor services.
- Determining the policy model.
- Creating groups, policies, and automated workflows.

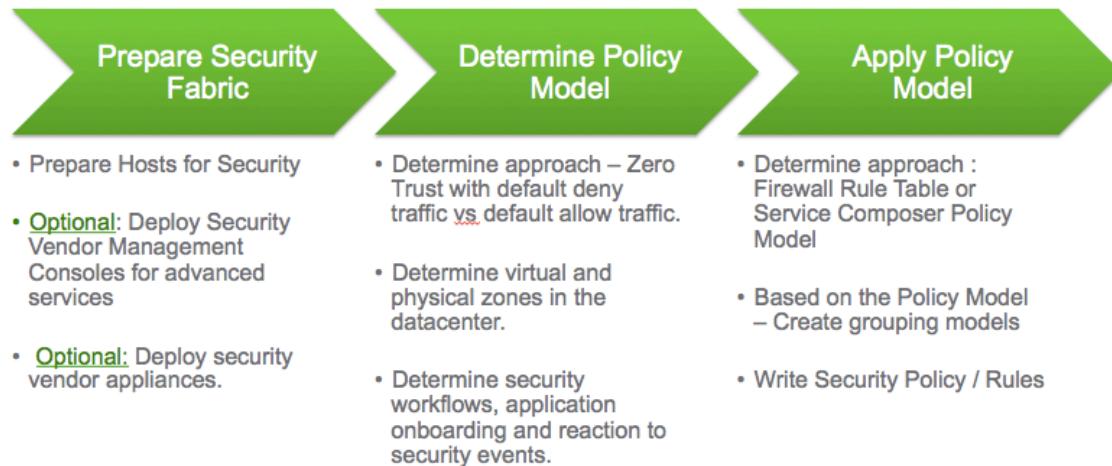


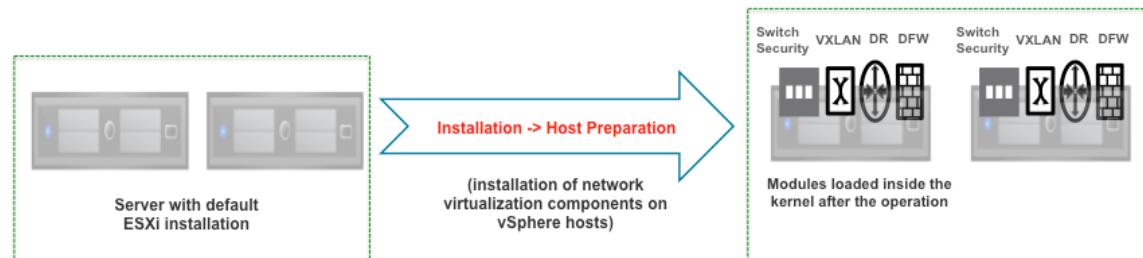
Figure 128 – Process of Developing Policy and Micro-Segmentation

#### 5.4.1 Preparing Security Services for Datacenter

NSX security services consist of built-in services including distributed firewall, edge firewall along with the extensibility frameworks for enabling host and network based advanced services from third party vendors. We will first look at the built-in services like distributed firewall deployments. We will then cover deployments of extensible frameworks in the datacenter.

##### 5.4.1.1 Deploying Distributed Firewall

Distributed firewall rules are enabled on the NSX Manager by default, with a single permit rule. This allows activation of the distributed firewall in each ESXi host, irrespective of whether it is a brownfield or greenfield environment. To enable distributed firewall in a host, deploy the VIBs to that host; this will enable a single DFW instance per vNIC on the VM.



**Figure 129 – NSX Installation and DFW initiation**

NSX manager deploys VIBs to specified clusters. If a cluster is already prepared, NSX manager automatically prepares any new host subsequently added to the cluster. NSX components – NSX manager, controllers, and Edge appliances – are automatically excluded from distributed firewall. Third party security virtual appliances (SVAs) deployed via NSX are also automatically excluded from the distributed firewall. Host preparation does not require a reboot of the host. NSX manager will show the status of distributed firewall on each host. Alerts and error messages are provided that offer visibility in case host has a problem.

The following best practices will help you cover various scenarios in your virtualized data center.

### **Prepare Hosts in the Compute and Edge Clusters**

Datacenter designs generally involve 3 different types of clusters – management, compute and edge. VDI deployments may also have desktop clusters. Compute, desktop and edge clusters must be prepared for the distributed firewall.

The management cluster enables operational tools, management tools, and security management tools to communicate effectively with all guest VMs in the datacenter. If east-west communication paths between these tools and the datacenters are completely known, then enable distributed firewall on these clusters. This ensures that management tools are not locked out of the datacenter environment.

A separation of security policy between management components and workload is desired. The policy and guidance for management cluster security is not discussed in this design guide. The management cluster can be prepared for security VIB in small environment where edge and management clusters can be combined. It is recommended to exclude management components and security tools from the DFW policy to avoid the lockout. Security tools contain management servers must talk to their virtual appliances to provide signature updates, rule sets, and initiate command and controls (e.g., scans). Asset management tools may need to deploy patches on the virtual machines. In these scenarios, if the communication and data paths are known then add DFW rules, however it may be easier to exclude them from DFW to avoid disruption of essential controls.

### **Account for vMotion of guest VMs in the Datacenter**

Preparing the hosts does not only involve the clusters/hosts that are currently running workloads, but also clusters that may receive a workload from a vMotion event. In general, there are two types of vMotion in a vSphere 5.x environment and an additional type in a vSphere 6.x environment:

- **vMotion Across Hosts in the Same Cluster (vSphere 5.x, 6.0):** Ensure that when new hosts are added to a cluster, they are monitored for VIB

- deployments. If DRS rules restrict guest VMs vMotion to hosts on the same cluster and hosts are not properly prepared, guest VMs will lose protection.
- **vMotion Across Clusters (vSphere 5.x, 6.0):** Ensure that clusters which are part of vMotion in the same vSphere datacenter construct for workloads are also prepared for distributed firewall.
  - **vMotion Across vCenters. (vSphere 6.0 only):** With vSphere 6.0, it is possible to vMotion guest VMs across vCenter domains. Ensure clusters that are part of your vMotion rules between vCenter domains are prepared for deployment for distributed firewall. This will ensure guest VMs continue to receive protection as they cross vCenter domains.

## Enable Specific IP Discovery Mechanisms for Guest VMs

NSX allows distributed firewall rules to be written in terms of grouping objects that evaluate to virtual machines rather than IP addresses. NSX automatically converts the rules that contain virtual machines to actual IP addresses. To automatically update firewall rules with IP addresses, NSX relies on 5 different mechanisms to provide IP addresses.

Enable at least one of these IP discovery mechanisms:

- Deploy VMtools in guest VM.
- Enable DHCP snooping (NSX 6.2.1 and above).
- Enable ARP snooping (NSX 6.2.1 and above).
- Manually authorize IP addresses for each VM.
- Enable trust on first use of IP address for a VM.

---

**Note:** All IP discovery mechanisms are available for both IPv4 and IPv6 addressing.

---

VMtools and ARP snooping enable you to learn IP addresses of the virtual machine in all cases. Where static IP addresses are configured manually on the guest VM, this is very effective. For DHCP-based IP address management, DHCP snooping is the right choice. DHCP snooping also keeps track of lease expiry timers and will make IP addresses stale. DHCP snooping uses both DHCPv6 and ND Snooping for IPv6 addresses.

Where identity based firewalling is used, VMtools deployment is essential. VMtools detects not just the IP address of the VM but also the users logging in.

VMtools and DHCP snooping are the recommended IP discovery mechanisms for guest VMs. Both of these mechanisms report changes in IP addresses as soon as they occur. They are also comprehensive in any environment, covering both dynamic and static addressing scenarios. If VMtools is used, care should be taken to ensure that it is running at all times. ARP snooping is also effective, but it should be used with SpoofGuard to avoid ARP poisoning.

The recommended approaches in selecting IP discovery are as follows:

- If VMtools is deployed, use VMtools and DHCP snooping for discovery of IP Addresses.
- If VMtools is not deployed, use DHCP and ARP snooping for discovery of IP Addresses.
- If Identity Firewall is required, VMtools is a strict requirement.

NSX provides a SpoofGuard feature to avoid spoofing of IP addresses. There are two SpoofGuard mechanisms: trust on first use and manual authorization.

Trust on first use will trust the first IP address reported to the NSX manager via any of the described methods. For manual authorization, it will present the set of IP addresses discovered via any of the methods for approval by users. Users are still able to edit and add a different IP address.

Note that VMs initially get a link local IP address 169.\*.\*.\*. Please ensure that is added to the list of trusted address via SpoofGuard policy.

In a DHCP environment, trust on first use is not recommended the frequent IP address changes will cause operational challenges. Similarly, manual authorization of IP addresses will require automation in a large dynamic environment.

Policies are used to enable SpoofGuard for virtual machines. These policies are tied to logical switches or distributed port-groups. For a single policy, all IP addresses approved must be unique. If overlapping IP addresses are desired, the virtual machines should be part of different SpoofGuard policies, i.e., should be part of different logical switches or distributed port-groups.

#### **5.4.1.2 Extensibility framework for Advanced Security Services**

NSX security fabric includes two extensible frameworks that allow for the datacenter to use built-in and third party vendor advanced security services. This will extend NSX built-in segmentation and provide additional security services. Without deploying NSX extensible frameworks, third party services will not be available to provide security.

Extensible framework Guest Introspection enables host based security services like anti-virus, anti-malware, and file integrity monitoring. Deploy this requires the Guest Introspection service appliance. As part of the Guest Introspection service appliance, NSX automatically deploys vShield Endpoint VIBs. The deployment considerations are the same as with distributed firewall.

Extensible framework Network Introspection enables network security services like L7 firewall and IDS/IPS along with other monitoring services. This is deployed as part of the distributed firewall deployments. The deployment considerations are the same as with distributed firewall.

#### **5.4.1.3 Deploying NSX Built-in Advanced Services**

NSX provides two advanced detection services – data security and activity monitoring.

##### **Data Security**

Data security allows detection of sensitive data in the guest virtual machines. Examples of sensitive data are credit card information, social security numbers, PCI data, PII data, and driver's license numbers. NSX comes with 100+ sensitive data detection policies that can be used in various scenarios.

##### **Activity Monitoring**

Activity monitoring allows detection of network connections for each individual application inside a guest virtual machine, regardless of its anywhere inside the datacenter. If NSX is integrated with Active Directory, activity monitoring provides users who are using specific applications to talk to other applications on the network.

To enable these services, the Guest Introspection extension framework must be deployed. Additionally, for data security, the data security service needs to be deployed on the cluster. Both these services generate reports that can be exported out to syslog servers or as CSV files.

#### **5.4.1.4 Deploying Third Party Vendor Security Services**

Deployment of third party vendor security services like anti-malware, anti-virus, L7 firewall, or IPS/IDS will require considerations in addition to preparing the hosts for the distributed firewall. NSX integrated third party security services contain two parts: the management console/server and a Security Virtual Appliance (SVA). NSX manager requires connectivity with the third party management server when creating policies. NSX manager also requires connectivity to web servers where the OVF files for the security virtual appliances are stored.

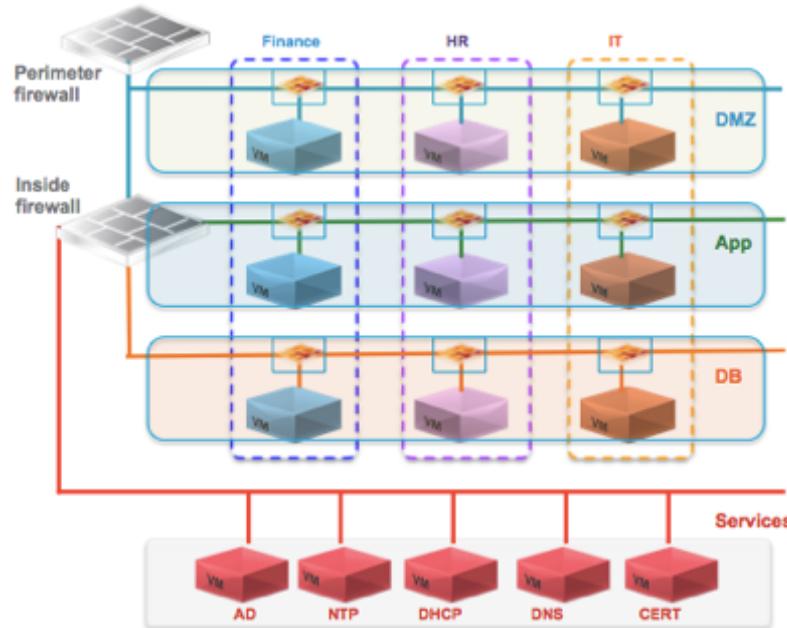
Third party vendor security managers should be deployed in the management cluster in the datacenter design. NSX manager deploys SVAs across the datacenter on every host. The deployment considerations for SVAs are the same as deploying the distributed firewall and preparing hosts across the datacenter. All vMotion considerations included for distributed firewall are also applicable for SVA deployments.

#### **5.4.2 Determining Policy Model**

Policy models in a datacenter are essential to achieve optimized micro-segmentation strategies. They are required in order to enable optimum groupings and policies for micro-segmentation.

The first criteria in developing a policy model is to align with the natural boundaries in the data centers such, as tiers of application, SLAs, isolation requirements, and zonal access restrictions (e.g., production/development, internal/external). Associating a top-level zone or boundary to a policy helps apply the consistent and yet flexible control. Global changes for a zone can be

applied via single policy, however within the zone there could be a secondary policy with sub-grouping mapping to a specific sub-zone. An example production zone might itself be carved into sub-zones like PCI or HIPPA, or there may be multiple zones for VDI deployments based on user types. Various types of zones can be seen in Figure 129, where the connectivity and access space is segmented into DMZ, app, and database tier zones. There are also zones for each department as well as shared services. Zoning creates relationships between various groups, providing basic segmentation and policy strategies.

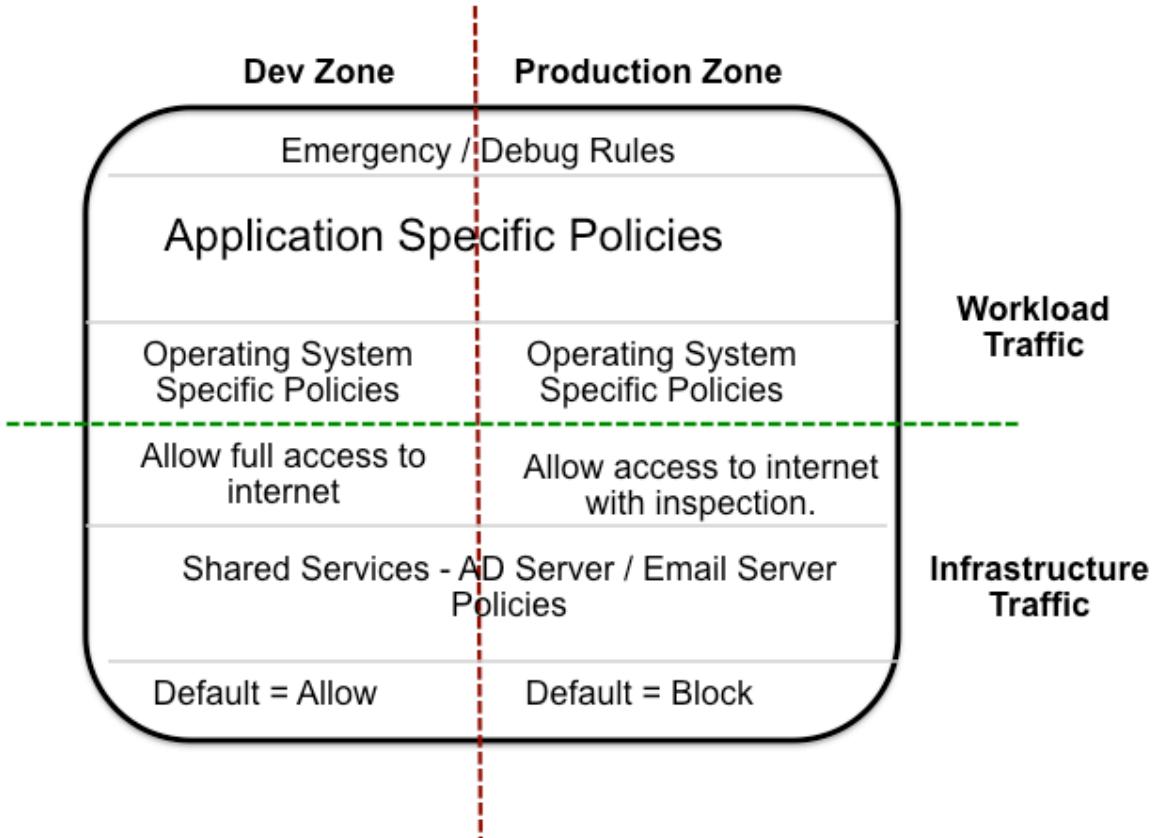


**Figure 130 – An Example of Various Zones**

A second criterion in developing policy model identifying reactions to security events and workflows. If vulnerability is discovered, what are the mitigation strategies? Where is the source of the exposure – internal vs external? Is the exposure limited to a specific application or operating system version?

The answering for all these questions help shape a policy model. Policy models should be flexible enough to address ever-changing deployment scenarios rather than simply be part of the initial setup. Concepts such as intelligent grouping, tags, policy inheritance, global vs. local policy, and hierarchy provide flexible and agile response capability for both steady state protection and during instantaneous threat response.

A sample policy model is shown in Figure 130.



**Figure 131 – Policy Model for Different Zone and Workload**

To effectively protect datacenter from attacks, use a zero-trust model where any access to the resources and workload are by default not trusted unless specifically permitted by a policy. NSX micro-segmentation along with distributed firewall allows creation of this zero-trust model. In a zero trust model, traffic is whitelisted and allowed. The default rule for such firewall policies should have a default deny of any traffic. This model ensures that the rules will allow only specific traffic, however this is also the hardest to implement operationally. If there is little insight into east-west traffic, then this option will cause disruption of existing datacenter traffic.

The opposite of a zero-trust model is also a design choice. In this case, all traffic is by allowed default with administrators specifically restricting/blacklisting certain traffic. This model requires the administrator to close down traffic that is viewed as unnecessary. It is very easy to execute operationally, but very difficult to ensure the best security posture. The goal should be start at a default allow model and quickly move to a default block posture.

The quickest and the most efficient way to deploy the policy is to create big perimeters with light restrictions, but provide enough protection with sets of rules around zones. Then proceed to segment or unite sets while continuing to shrink the perimeter size with more restrictive rules.

Sample with large perimeters could include:

- Allow or block access to the Internet.
- Allow shared services access to particular ports.
- Specific generic operating system policies.

In the example of zones in Figure 129, suggested starting steps would be:

- Determine zones that are allowed access to the external world or Internet. (e.g., restrict Internet access to the DMZ zone).
- Determine whether all the zones need access to the shared services.
- Determine if each of the departments need to talk to each other.
- Determine if DMZ zones need to talk to app and/or database zones.

Once these broader zones and rules are created, each of the individual zones can be further analyzed to understand what specific rules are required.

In this example, each of the zones can have a security group, and each security group can have a separate security policy with a distinct set of rules.

Once the rules for the larger perimeters are completed, create a new perimeter around the application. Profile the application for its traffic pattern, then create more granular rules to control its traffic. Repeat this process for each application. This is an efficient strategy for speeding along the micro-segmentation process.

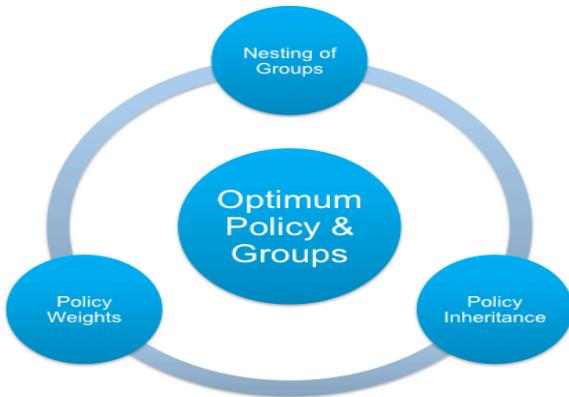
Application profiling is the essential part of micro-segmentation deployment. The aim of micro-segmentation is to enact a policy model based on either an open or zero trust model, where workloads are protected from threats arising both internally and externally. A variety of tools and methods available to expedite this process. Well-known applications contain hardening guides that specify various east-west and north-south communication paths. Collecting flow data using NetFlow can be helpful. Enabling DFW and logging all flows that can be analyzed at a later date can provide a detailed profile of the application.

### **5.4.3 Consideration for creating Groups and Policies**

This section examines the various design considerations for creating groups and policies along with various tweaks and nuances in deploying policy models to achieve higher efficacy.

#### **5.4.3.1 Optimal grouping and policy**

There are three factors to an effective security policy for a datacenter, combining grouping strategy, policy model strategy, and policy weights. Optimized policy and grouping requires a balance for all the three factors. This helps to better maintain the security posture of the datacenter in the long run. These factors are depicted in Figure 131.



**Figure 132 – Factors Affecting Optimum Policy & Groups**

### **Making Grouping Simple**

To make grouping simple, policy models and policy weights must be well thought out. A simpler grouping strategy creates multiple groups per application or zone. VMs in this model would ideally reside in mutually exclusive security groups, as they are not nested. This would require the weight of each policy group receive careful consideration to determine proper precedence. Additionally, it is likely policy rules will be complicated and sprawl similar rules across different policies.

### **Making Policy Weights Simple**

The fewer the policy groups, the simpler the weight assignment. This strategy will include a lot of nested groups. In this approach, virtual machines have to reside in multiple security groups, increasing complexities in grouping as well as in policy rule creation.

### **Making Policy Rules Simple**

Creation of minimum policy rules for protecting a given zone or application would ideally be operationally simple to understand. To make policy rules simple, the nesting of groups and policy weights must be very well designed.

In the example case, web, app and database tiers require access to shared services (e.g., DNS, AD, SSO, NTP). If the policy is not nested, then all three tiers require a distinct security group, which then must be individually updated. Nesting of a group implies that a VM may reside in multiple groups at the same time. This may cause rule evaluation for each VM can become overly cumbersome when the nesting is too deep; thus nesting should be limited to 3 to 5 levels of depth.

With multiple policies, execution order has to be efficient as policy weights translate to firewall precedence. Policy weights become crucial to determine the order in which the rules will be applied. Too much nesting may make policy simpler to administer, but the VM may reside in multiple security groups which will increase the complexity of policy weight. A well-balanced nesting depth and limited policy grouping will be the optimal solution.

### **5.4.3.2 Group Creation Strategies**

The most basic grouping strategy is creation of a security group around every application that is on-boards in the NSX environment. Each 3-tier, 2-tier, or single tier applications should have its own security group; this will enable faster operationalization of micro-segmentation. When combined with a basic rule that says, “No Application can talk to another except for shared essential services like DNS, AD, DHCP servers”, this enforcement of granular security inside perimeter. Once this basic micro-segmentation is achieved, and then writing rules per application will be desirable.

Creation of security groups gives more flexibility as the environment changes over time. Even if the rules contain only IP Addresses, NSX provides a grouping object called an IPSet that can encapsulate IP Addresses. This can then be used in Security Groups.

This approach has three major advantages:

- Rules stay more constant for a given policy model, even as the datacenter environment keeps changes. The addition or deletion of workloads will affect group membership alone, not the rules.
- Publishing a change of group membership to the underlying hosts is more efficient than publishing a rule change. It is faster to send down to all the affected hosts and cheaper in terms of memory and CPU utilization.
- As NSX adds more grouping object criteria, the group criteria can be edited to better reflect the datacenter environment.

### **Use Grouping to enhance visibility**

A virtual machine can be part of multiple groups. Groups can be used for visibility and categorization of the datacenter as well as a way to apply security rules to workloads. A security group can contain all virtual machines that have Windows 2003 operating system; there might not be a security rule for all virtual machines for that operating system, but this enhances the visibility of workloads in the datacenter. In this example, migration plans can be developed when the operating system is at its end of life or a specific vulnerability policy can be developed based on an announced or discovered security exposure.

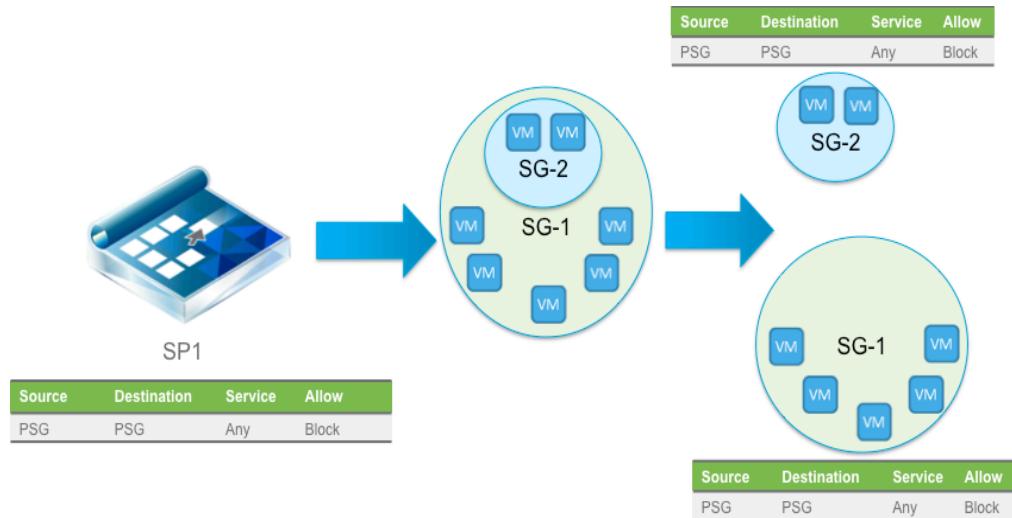
### **Efficient Grouping considerations**

Calculation of groups adds a processing load to the NSX manager. Different grouping mechanisms add different types of loads. Static groupings are more efficient than dynamic groupings in terms of calculation. At scale, grouping considerations should take into account the frequency of group changes for a virtual machine. A large number of group changes applied to a virtual machine frequently means the grouping criteria is sub-optimal.

### **Using Nesting of Groups**

Groups can be nested. A security group may contain multiple security groups or a combination of security groups and other grouping objects. A security rule

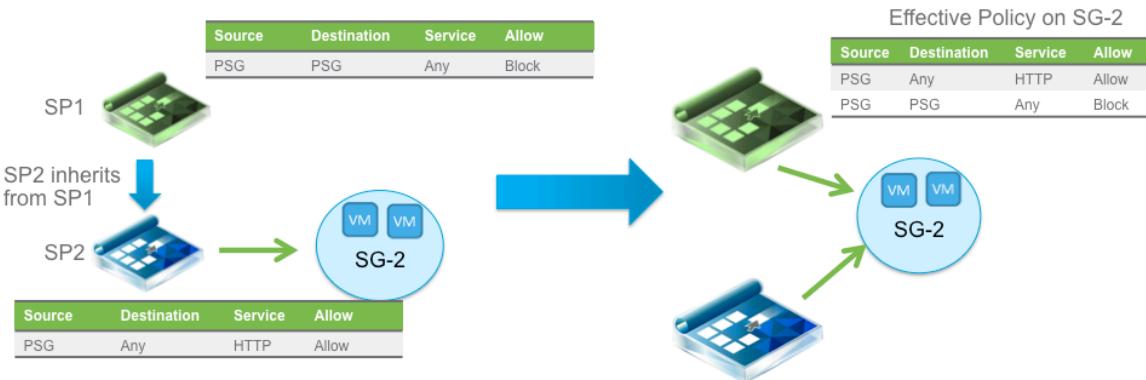
applied to the parent security group is automatically applied to the child security groups.



**Figure 133 – Policy Nesting Example**

### Using Policy Inheritance

Policy inheritances are generally costly operations for the NSX manager. They use more processing power to compute the effective policy applied to the individual virtual machines, there are legitimate use cases for them. If a service provider/tenant model is used, the base policies developed by the service provider will provide the guardrails for the tenants, while the child policies can be given to the tenants to create their own policies.



**Figure 134 – Policy Inheritance Example**

#### 5.4.3.3 Policy Creation Strategies

This section details the considerations behind policy creation strategies, help determine which capability of NSX platform should be exercised and how various grouping methodologies and policy strategies can be adopted for a specific design.

## Traditional Approach vs. NSX Security Policy Approach

One of the biggest questions that arise about deploying security with NSX platform involves continued use of traditional security rules vs. migrating to the NSX security policy method.

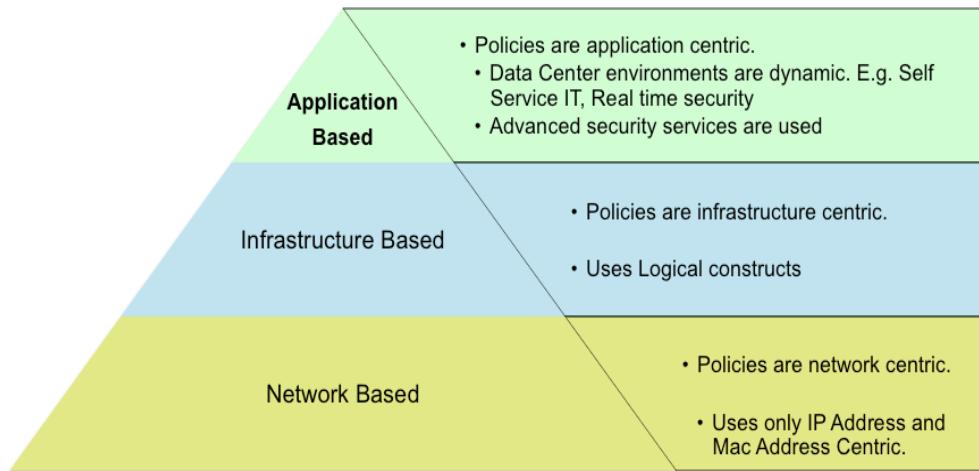


Figure 135 – Context and Hierarchy to Policy Creation

There are three general approaches for policy models showing in Figure 134. The NSX provides flexibility of suitable model with specific advantages and disadvantages.

**Network Topology based Policy Models:** This is the traditional approach of grouping based on L2 or L3 elements. Grouping can be based on MAC addresses, IP addresses, or a combination of both. NSX supports this approach of grouping objects. Network topology models include both physical topologies as well as logical topologies.

**Advantages:** This method of grouping works great for migrating existing rules from a different vendor's firewall environment.

**Disadvantages:** The security team needs to aware of the network topology to deploy network-based policies. There is a high probability of security rule sprawl, as grouping based on vCenter objects, NSX objects, or virtual machine attributes are not used. The rules cannot be targeted to the workload; they are spread everywhere in the datacenter.

**Infrastructure based Policy Models:** In this approach, policies are based on SDDC infrastructure like vCenter clusters, logical switches, and distributed port groups. An example would mark cluster 1 to cluster 4 for PCI applications, with grouping done based on cluster names and rules enforced based on these groups. Another example would be to connect all VMs pertaining to a particular application to a specific Logical Switch.

**Advantages:** Security rules are more comprehensive. The security team needs to work closely with the administrators that manage compute, networking, and storage in the datacenter. Workload abstraction using grouping is better than the network topology policy model.

**Disadvantages:** The security team still must understand the logical and physical boundaries of the datacenter. Unlike the earlier policy model, it will be imperative to understand the compute and storage models. Workloads have physical restrictions on where they can reside and where they can move.

**Application based Policy Model:** These policy models are developed for environments that are extremely dynamic in nature. Multiple physical and logical network topologies may be created and removed on demand, potentially through use of automation or cloud consumption models to deploy workloads. Examples of this deployment models include self-service IT, service provider deployments, and enterprise private/public cloud.

**Advantages:** Security posture and rules are created for the application, independent of the underlying topology of compute, storage and network. Applications are no longer tied down to either network constructs or SDDC infrastructure, thus security policies can move with the application irrespective of network or infrastructure boundaries. Policies can be turned into templates and reused. Security rule maintenance is easier as the policies live for the application life cycle and can be destroyed when the application is decommissioned. This practice helps in moving to self-service or hybrid cloud models.

**Disadvantages:** The security team must to be aware of the interactions of the application that it is trying to secure.

The following recommendations are provided to assist in developing optimal policy and grouping strategies.

- Nesting levels should be only 3 to 5 levels deep.
- Virtual machines should not be part of more than 10 mutually exclusive security groups.
- Virtual machines should not change security groups frequently.
- Policy inheritances should be kept at minimum around 3 – 5 and serve only to create guardrails.
- Base policies in an inheritance should have fairly static criteria.
- Policy weights and ranks should be kept fairly simple so tracking and debugging of issues is easier.

#### 5.4.4 Deployment Models

Deployment models required additional considerations for design, as they generally indicate how users and applications going to access the datacenter. There are multiple possibilities for workload location and migration; common use cases include:

1. L3 to L7 policies across zones.
2. Secure user environment using VDI/mobile infrastructure.
3. Tenancy model with service provider.
4. Protection between physical-to-virtual workloads.
5. Disaster recovery sites to back up data from primary datacenters.
6. Remote Office/Branch Office (ROBO).

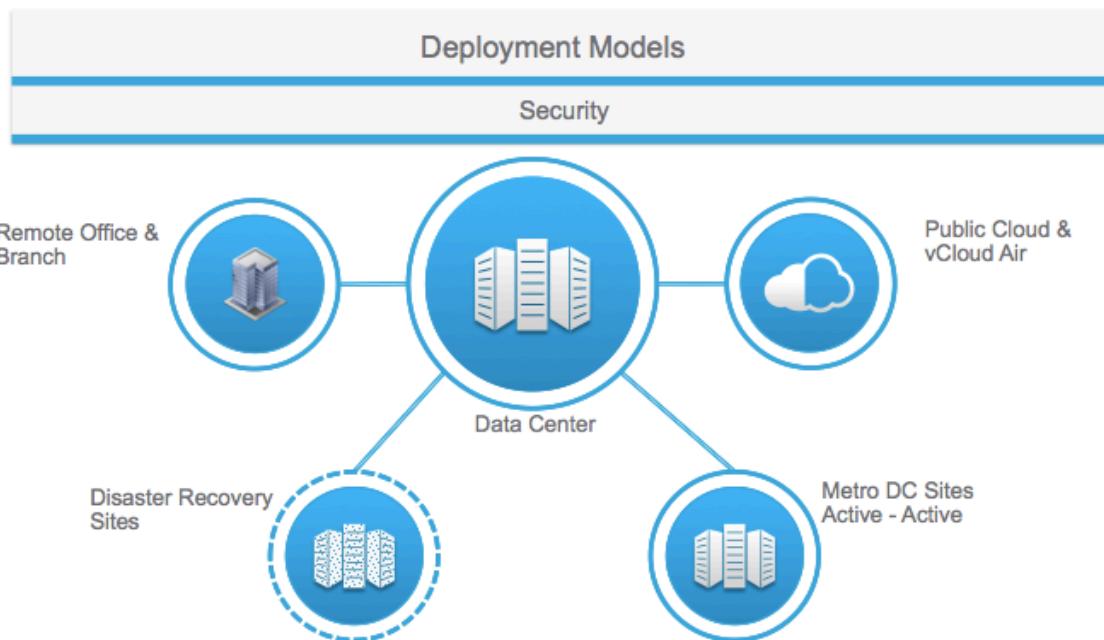


Figure 136 – Application of Security Models

#### Considerations for deployment models

There are various tweaks that can be performed to deliver greater efficiency in the design of various deployment models.

#### L3 to L7 Policies across Zones

A datacenter typically includes various zones for regulation (e.g., PCI, HIPPA), workloads (e.g., production, test, DMZ, application), users (e.g., developers, contractors, knowledge workers), or geographies (e.g., USA, EMEA, APJ).

Regulatory requirements or enhanced security mandates may not only distribute firewalling using L2-L4 rules, but additionally deploy L7 firewall rules and packet introspection technologies like IPS.

When traffic moves across zones, it is recommended to use advanced security services from NSX partners to provide L7-based security technologies. Grouping strategies should identify zones and create them to provide logical demarcations. Policy strategies should include zone specific policies that incorporate both built-in NSX security modules and advanced services from vendors for enhanced security.

### **Secure User Environment using VDI / Mobile Infrastructure**

This model is generally pervasive in a virtual desktop or desktop as a service deployment. With NSX, a firewall can be adapted to a particular user logging into the virtual machine. NSX allows intelligent grouping that can be based on Active Directory user groups. NSX also allows detection and determination of users logging into the virtual machine that allows for adding or removing virtual machines from active directories.

If Active Directory integration is not desirable in a datacenter, then security tags can be used to determine the users logged in. For example, if Horizon View deploys and allocates a virtual desktop to a particular user, the virtual machine could be tagged with a security tag specifying the user details. Security grouping can be based on the tags that provide user related information

### **Tenancy model with service provider**

In this model, the service provider provides guardrails into what the tenant can do. The tenant may be free to create its own rules within the sandbox provided.

In a traditional approach, service providers can create sections for tenants. Tenants can create rules in those sections and publish individual sections. The rule itself can be tagged with keywords that enable logging to be provided only to the tenants for those rules.

In a policy approach, service providers can provide child policies to the tenants that inherit base policies. This will ensure that the guardrails are covered in the base policy while tenants can update the child policies. If policy inheritance approach is not suitable in the design, then tenant policies can be of lower weights compared to service provider policies.

### **Protecting Traffic from Physical workloads**

Physical workloads are represented by their IP addresses in the NSX domain. Communications with physical workloads can be controlled in two different places in the NSX domain. Rules can be provisioned on the NSX Edge firewall or in the distributed firewall.

### **Controlling communication between VM across multiple datacenters**

With NSX 6.2, NSX allows universal firewall rules that can be created and synced between vCenters. This is in addition to the local distributed rules that are allowed per vCenter.

### **Remote Office / Branch Office (RoBo) Model**

In this model, since vCenter encompasses both the central office and the remote office. Creating sections in the firewall that represent the remote/branch office is recommended. The section reduced the load on NSX manager and only publishes the rules in the section and thus its best approach in any deployments.

## 6 Conclusion

VMware NSX network virtualization solution addresses current challenges with physical network infrastructure, bringing flexibility, agility, and scale through VXLAN-based logical networks. Along with the ability to create on-demand logical networks using VXLAN, the NSX Edge Services Gateway helps users deploy various logical network services such as firewall, DHCP, NAT and load balancing on these networks. This is possible due to its ability to decouple the virtual network from the physical network and reproduce these properties and services in the virtual environment.

NSX reproduces in the logical space network services traditionally delivered by the physical infrastructure, such as switching, routing, security, load balancing, virtual private networking. It extends connectivity into the logical space for physical devices connected to the external network (Figure 137).



Figure 137 - Logical Network Services Provided by NSX

### References

NSX @ VMware.com

<https://www.vmware.com/products/nsx/>

NSX Documentation

[https://www.vmware.com/support/pubs/nsx\\_pubs.html](https://www.vmware.com/support/pubs/nsx_pubs.html)

VMware vSphere 6 Documentation

<https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-6-pubs.html>

[VMware vSphere 5.5 Documentation](#)

<https://pubs.vmware.com/vsphere-55/index.jsp>