VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

Intended Audience

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

Overview

IT organizations have gained significant benefits as a direct result of server virtualization. Server consolidation reduced physical complexity, increased operational efficiency, and the ability to dynamically re-purpose underlying resources to quickly and optimally meet the needs of increasingly dynamic business applications are just a handful of the gains that have already been realized.

Now, VMware's Software Defined Data Center (SDDC) architecture is 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 and storage. In much the same way that server virtualization programmatically creates, snapshots, deletes and restores software-based virtual machines (VMs), VMware NSX network virtualization programmatically creates, snapshots, deletes, and restores software-based virtual networks. The result is a completely transformative approach to networking that not only enables data center managers to achieve orders of magnitude better agility and economics, but also allows for a vastly simplified operational model for the underlying physical network. With the ability to be deployed on any IP network, including both existing traditional networking models and next generation fabric architectures from any vendor, NSX is a completely non-disruptive solution. In fact, with NSX, the physical network infrastructure you already have is all you need to deploy a software-defined data center.

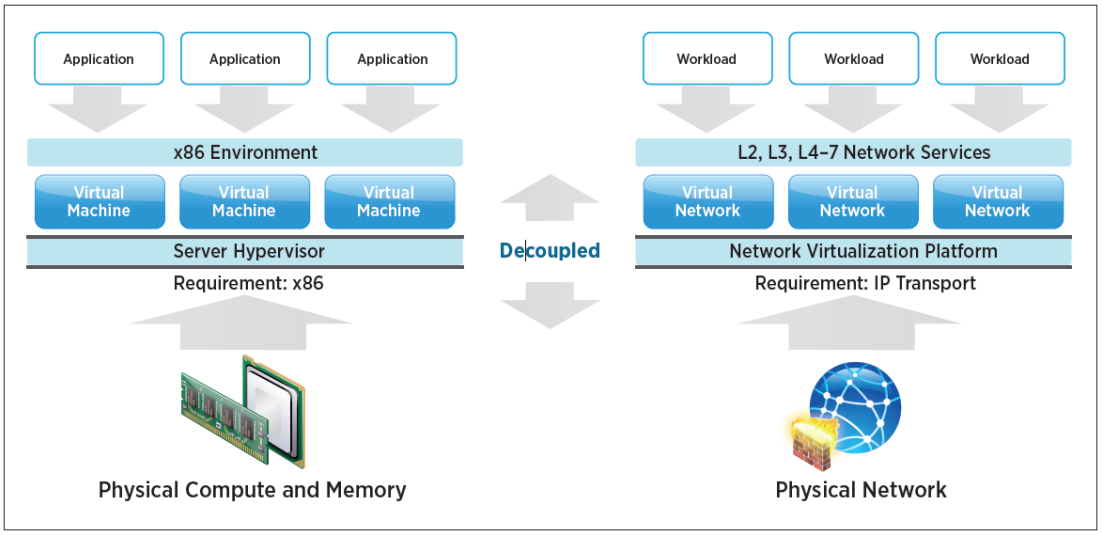


Figure 1 - Server and Network Virtualization Analogy

Figure 1 draws an analogy between compute and network virtualization. With server virtualization, a software abstraction layer (server hypervisor) reproduces the familiar attributes of an x86 physical server (e.g., CPU, RAM, Disk, NIC) in software, allowing them to be programmatically assembled in any arbitrary combination to produce a unique virtual machine (VM) in a matter of seconds.

With network virtualization, the functional equivalent of a "network hypervisor" reproduces the complete set of layer 2 to layer 7 networking services (e.g., switching, routing, firewalling and load balancing) in software. As a result, these services can be programmatically assembled in any arbitrary combination, to produce unique, isolated virtual networks in a matter of seconds.

Not surprisingly, similar benefits are also derived. For example, just as 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 and allow IT to treat the physical network as a pool of transport capacity that can be consumed and repurposed on demand. 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, you already have the network you need to deploy a next-generation software defined data center. This paper will highlight the various functionalities provided by the VMware NSX for vSphere (NSX-v) architecture and the design factors you should consider to fully leverage your existing network investment and optimize that investment with VMware NSX-v.

Note: in the rest of this paper the terms “NSX” and “NSX-v” are used interchangeably and they always refer to the deployment of NSX with vSphere.

Introduction to Network Virtualization

Overview of NSX-v Network Virtualization Solution

An NSX-v deployment consists of a data plane, control plane and management plane, as shown in Figure 2.

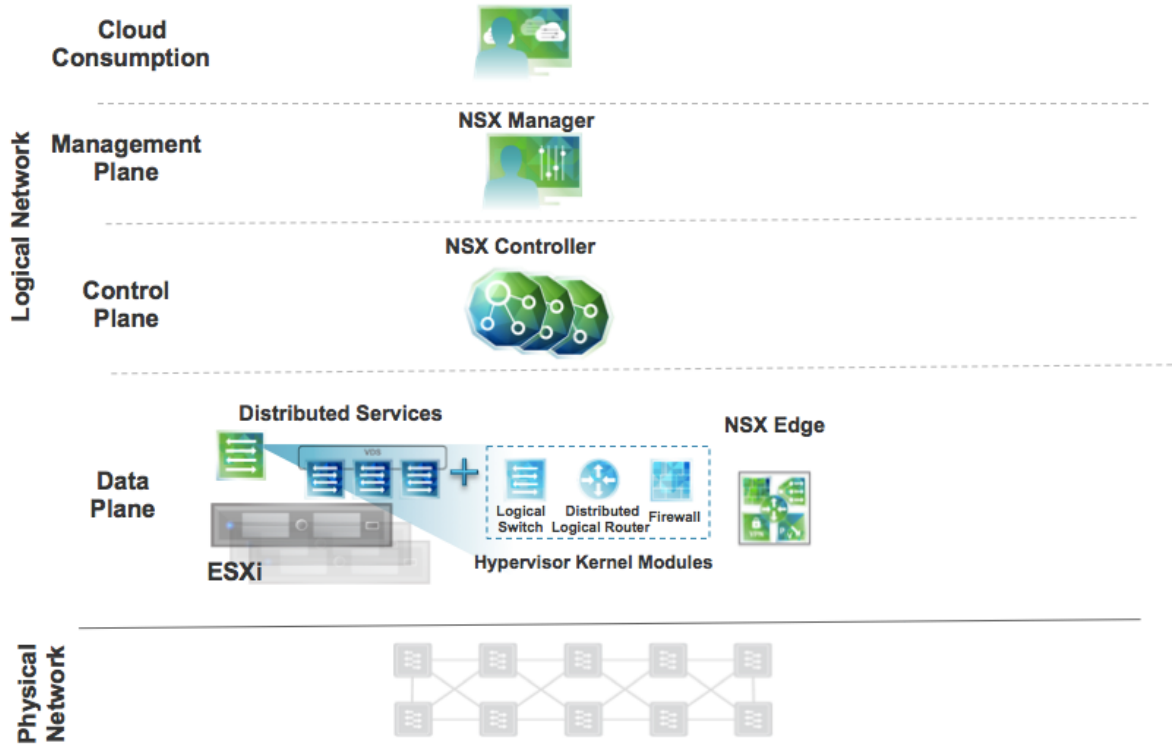


Figure 2 - NSX Components

Control Plane

The NSX control plane runs in the NSX controller. In a vSphere-optimized environment with VDS the controller enables multicast free VXLAN and control plane programming of elements such as Distributed Logical Routing (DLR).

In all cases the controller is purely a part of the control plane and does not have any data plane traffic passing through it. The controller nodes are also deployed in a cluster of odd members in order to enable high-availability and scale.

Data Plane

The NSX Data plane consists of the NSX vSwitch. The vSwitch in NSX for vSphere is based on the vSphere Virtual Distributed Switch (VDS) with additional components to enable rich services. The add-on NSX components include kernel modules (VIBs) which run within the hypervisor kernel providing services such as distributed routing, distributed firewall and enable VXLAN bridging capabilities.

The NSX VDS abstracts the physical network and provides access-level switching in the hypervisor. It is central to network virtualization because it enables logical networks that are independent of physical constructs such as VLANs. Some of the benefits of the NSX vSwitch are:

- Support for overlay networking with the use of the VXLAN protocol and centralized network configuration. Overlay networking enables the following capabilities:
    - Creation of a flexible logical layer 2 (L2) overlay over existing IP networks on existing physical infrastructure without the need to re-architect any of the data center networks.
    - Agile provision of communication (east–west and north–south) while maintaining isolation between tenants.
    - Application workloads and virtual machines that are agnostic of the overlay network and operate as if they were connected to a physical L2 network.
- NSX vSwitch facilitates massive scale of hypervisors.
- Multiple features—such as Port Mirroring, NetFlow/IPFIX, Configuration Backup and Restore, Network Health Check, QoS, and LACP—provide a comprehensive toolkit for traffic management, monitoring and troubleshooting within a virtual network.

Additionally, the data plane also consists of gateway devices that can provide communication from the logical networking space (VXLAN) to the physical network (VLAN). This functionality can happen at Layer 2 (NSX bridging) or at L3 (NSX routing).

Management Plane and Consumption Platforms

The NSX management plane is built by the NSX manager. The NSX manager provides the single point of configuration and the REST API entry-points in a vSphere environment for NSX.

The consumption of NSX can be driven directly via the NSX manager UI. In a vSphere environment this is available via the vSphere Web UI itself. Typically end-users tie in network virtualization to their cloud management platform for deploying applications. NSX provides a rich set of integration into virtually any CMP via the REST API. Out of the box integration is also currently available through VMware vCloud Automation Center (vCAC) and vCloud Director (vCD).

NSX Functional Services

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

Figure 3 - NSX Logical Network Services

- **Switching** – Enabling extension of a L2 segment / IP Subnet anywhere in the fabric irrespective of the physical network design
- **Routing** – Routing between IP subnets can be done in the logical space without traffic going out to the physical router. This routing is performed in the hypervisor kernel with a minimal CPU / memory overhead. This functionality

provides an optimal data-path for routing traffic within the virtual infrastructure (east-west communication). Similarly the NSX Edge provides an ideal centralized point to enable seamless integration with the physical network infrastructure to handle communication with the external network (North-South communication).

- **Distributed Firewall** – Security enforcement is done at the kernel and VNIC level itself. This enables firewall rule enforcement in a highly scalable manner without creating bottlenecks onto physical appliances. The firewall is distributed in kernel and hence has minimal CPU overhead and can perform at line-rate.
- **Logical Load-balancing** – Support for L4-L7 load balancing with ability to do SSL termination.
- **VPN**: SSL VPN services to enable L2 and L3 VPN services.
- **Connectivity to physical networks**: L2 and L3 Gateway functions are supported within NSX-v to provide communication between workloads deployed in logical and physical spaces.

In the next two sections of this document we'll describe more in detail the interaction of the various NSX components and each of the logical service mentioned above.

NSX-v Functional Components

The logical networks created by the NSX platforms 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 (virtual machines), whereas the Gateway Access Layer provides L2 and L3 connectivity into the logical space to physical devices and endpoints deployed in the traditional physical network infrastructure.

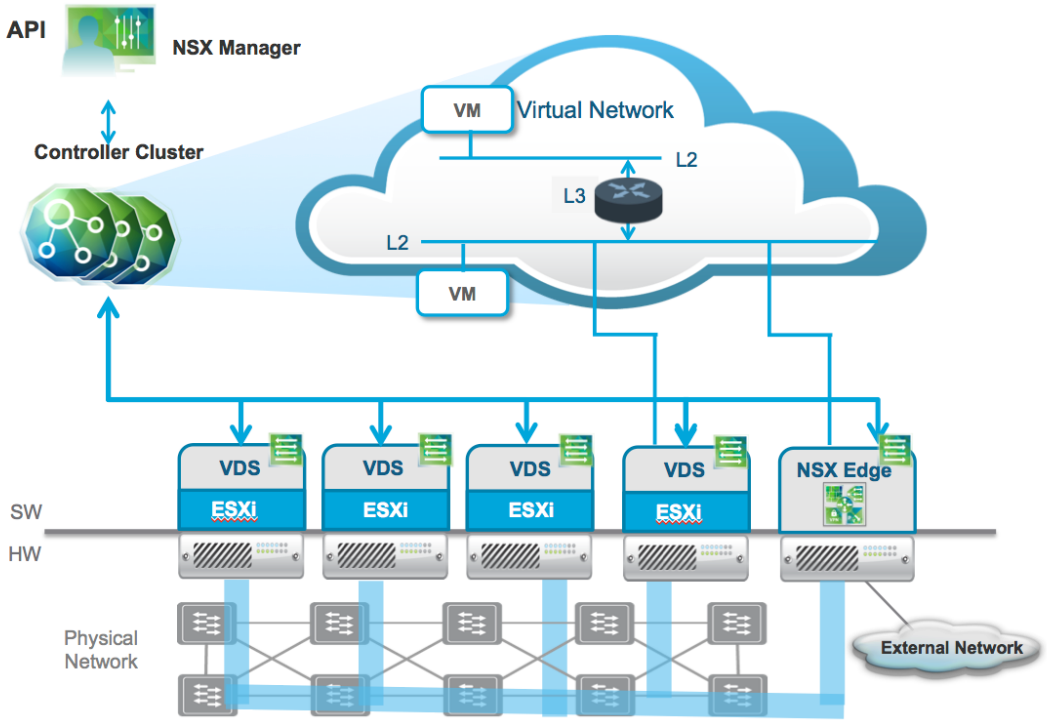


Figure 4 – NSX Functional Components

One of the main advantages of network virtualization is the capability of decoupling logical network connectivity from the underlying physical network fabric and providing logical network abstraction and functions. As shown in Figure 4, there are several functional components that enable that and form the NSX for vSphere architecture. As discussed in the rest of this document, the use of an overlay protocol (VXLAN) to encapsulate virtual machine traffic between the different functional components is a key function to achieve logical/physical network decoupling. Let's now take a more detailed look at the functions of each component of the NSX-v platform.

NSX Manager

The NSX manager is the management plane virtual appliance that helps configure logical switches and connect virtual machines to these logical switches. It also provides the management UI and entry point for API for NSX, which helps automate deployment and management of the logical networks through a Cloud management platform.

In the NSX for vSphere architecture, the NSX Manager is tightly connected to the vCenter server managing the compute infrastructure. In fact, there is a 1:1 relationship between the NSX Manager and vCenter and upon installation the NSX Manager registers with vCenter and injects a plugin into the vSphere Web Client for consumption within the Web management platform.

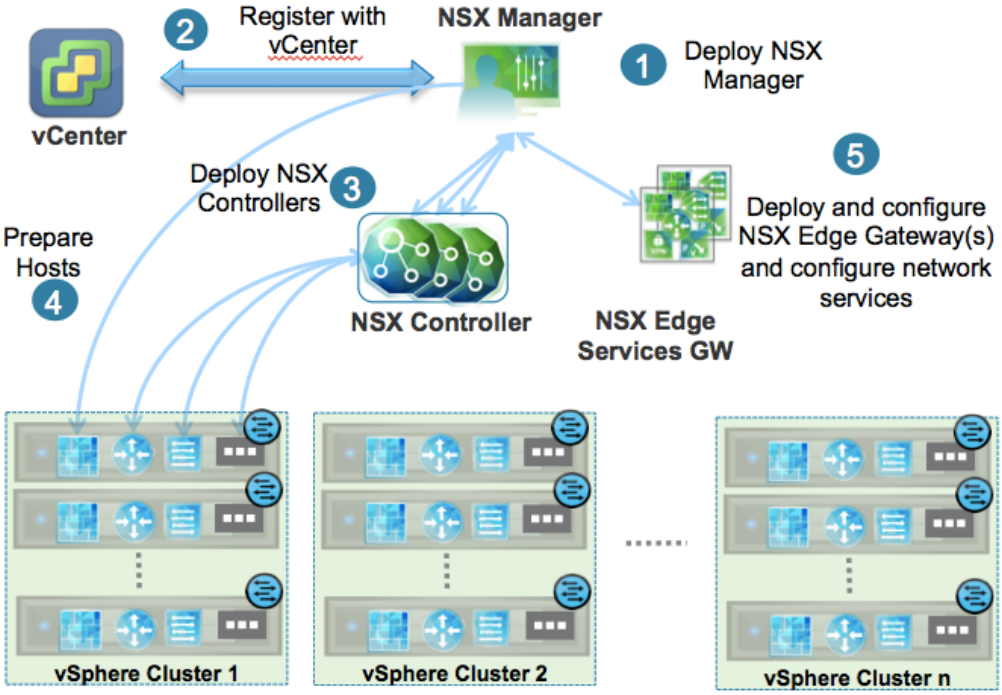


Figure 5 - NSX Manager Plugin inside vSphere Web Client

As highlighted in Figure 5, the NSX Manager is responsible for the deployment of the controller clusters and for the ESXi host preparation, installing on those the various vSphere Installation Bundles (VIBs) to enable VXLAN, Distributed Routing, Distributed Firewall and a user world agent used to communicate at the control plane level. 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). Those functionalities will be described in much more detail in following sections of this paper.

The NSX Manager also ensures security of the control plane communication of the NSX architecture by creating self-signed certificates for the nodes of the controller cluster and for each ESXi hosts that should be allowed to join the NSX domain. The NSX Manager installs those certificates to the ESXi hosts and the NSX Controller(s) over a secure channel; after that, mutual authentication of NSX entities occurs by verifying the certificates. Once this mutual authentication is completed, control plane communication is encrypted.

Note: in NSX-v software release 6.0, SSL is disabled by default. In order to ensure confidentiality of the control-plane communication, it is recommended to enable SSL via an API call. From 6.1 release the default value is changed and SSL is enabled.

In terms of resiliency, since the NSX Manager is a virtual machine, the recommendation is to leverage the usual vSphere functionalities as vSphere HA to ensure that the NSX Manager can be dynamically moved should the ESXi hosts where it is running occur a failure. It is worth noticing that such failure scenario would temporarily impact only the NSX management plane, while the already deployed logical networks would continue to operate seamlessly.

Note: the NSX Manager outage may affect specific functionalities (when enabled) as for example Identity Based Firewall (since it won't be possible to resolve usernames to AD groups), and flow monitoring collection.

Finally, NSX Manager data, including system configuration, events and audit log tables (stored in the internal data base), can be backed up at any time by performing an on-demand backup from the NSX Manager GUI and saved to a remote location that must be accessible by the NSX Manager. It is also possible to schedule periodic backups to be performed (hourly, daily or weekly). Notice that restoring a backup is only possible on a freshly deployed NSX Manager appliance that can access one of the previously backed up instances.

Controller Cluster

The Controller cluster in the NSX platform is the control plane component that is responsible in managing the switching and routing modules in the hypervisors. 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

support from the physical network infrastructure. Customers now don't have to provision multicast group IP addresses and also don't need to enable PIM routing or IGMP snooping features on physical switches or routers.

Additionally, the NSX Controller supports an ARP suppression mechanism that reduces the need to flood ARP broadcast requests across the 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 must deploy a Controller Cluster with multiple nodes. 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.

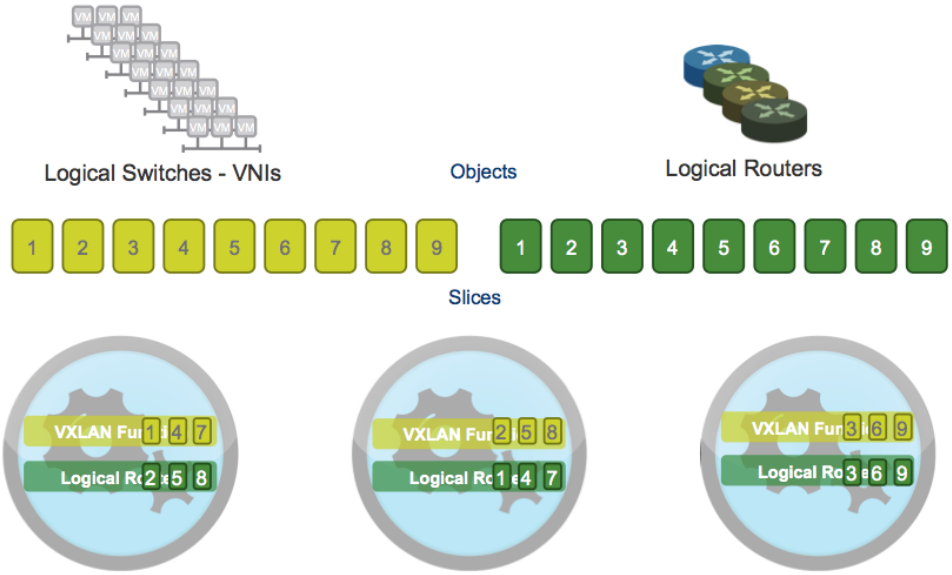


Figure 6 – Slicing Controller Cluster Node Roles

The Figure 6 above illustrates how the roles and responsibilities are fully distributed between the different cluster nodes. This means, for example, that different logical networks (or logical routers) may be managed by different Controller nodes: each node in the Controller Cluster is identified by a unique IP address and 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, so to be able to establish communication channels with all the members of the Controller Cluster. This allows the ESXi host to know at any given time what specific node is responsible for a given logical network

In the case of failure of a Controller Node, the slices for a given role that were owned by the failed 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 and determining when a node has failed, so to be able to reallocate the slices to the other nodes using a specific algorithm. The master also informs the ESXi hosts about the failure of the cluster node, so that they can update their internal information specifying what node owns the various logical network slices.

The election of the Master for each role requires a majority vote of all active and inactive nodes in the cluster. This is the main reason why a Controller Cluster must always be deployed leveraging an odd number of nodes.

Number of NSX Nodes in Cluster	2	3	4	5
Majority Number	2	2	3	3
Number of Nodes that can fail	0	1	1	2

Figure 7 - Controller Nodes Majority Numbers

Figure 7 highlights the different majority number scenarios depending on the number of Controller Cluster nodes. It is evident how deploying 2 nodes (traditionally considered an example of a redundant system) would increase the scalability of the Controller Cluster (since at steady state two nodes would work in parallel) without providing any additional resiliency. This is because with 2 nodes, the majority number is 2 and that means that if one of the two nodes were to fail, or they lost communication with each other (dual-active scenario), neither of them would be able to keep functioning (accepting API calls, etc.). The same considerations apply to a deployment with 4 nodes that cannot provide more resiliency than a cluster with 3 elements (even if providing better performance).

Note: NSX currently (as of software release 6.1) supports only clusters with 3 nodes. The various examples above with different numbers of nodes were given just to illustrate how the majority vote mechanism works.

NSX controller nodes are deployed as virtual appliances from the NSX Manager UI. Each appliance is characterized by an IP address used for all control-plane interactions and by specific settings (4 vCPUs, 4GB of RAM) that cannot currently be modified.

In order to ensure reliability to the Controller cluster, it is good practice to spread the deployment of the cluster nodes across separate ESXi hosts, to ensure that the failure of a single host would not cause the loss of majority number in the cluster. NSX does not currently provide any embedded capability to ensure this, so the recommendation is to 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://pubs.vmware.com/vsphere-55/index.jsp#com.vmware.vsphere.resmgmt.doc/GUID-7297C302-378F-4AF2-9BD6-6EDB1E0A850A.html

VXLAN Primer

The deployment of overlay technologies has become very popular because of their capabilities of 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 3, independently from how the underline physical infrastructure is configured. The physical network effectively becomes a backplane used to transport overlay traffic.

This decoupling effect helps in solving many of the challenges traditional data center deployments are facing nowadays, as for example:

- Agile and rapid application deployment: traditional networking design represents a bottleneck slowing down the rollout of new application at the pace that businesses are demanding. The time 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 to extend the L2 domains (VLANs) across the entire data center network infrastructure, affecting the overall scalability and potentially jeopardizing the overall resiliency of the design.
- Large-scale multi-tenancy: the use of VLANs as a mean of creating isolated networks limits to 4094 the maximum number of tenants that can be supported. This number, while it may seem large for typical enterprise deployments, it is becoming a serious bottleneck for most of the Cloud Providers.

Virtual Extensible LAN (VXLAN) has become the “de-facto” standard overlay technology, embraced by multiple vendors (it was developed by VMware in conjunction with Arista, Broadcom, Cisco, Citrix, Red Hat, and others). Deploying VXLAN is key to be able to build logical networks that provide L2 adjacency between workloads, without the issue and scalability concerns found with traditional layer 2 technologies.

As shown in Figure 8, 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).

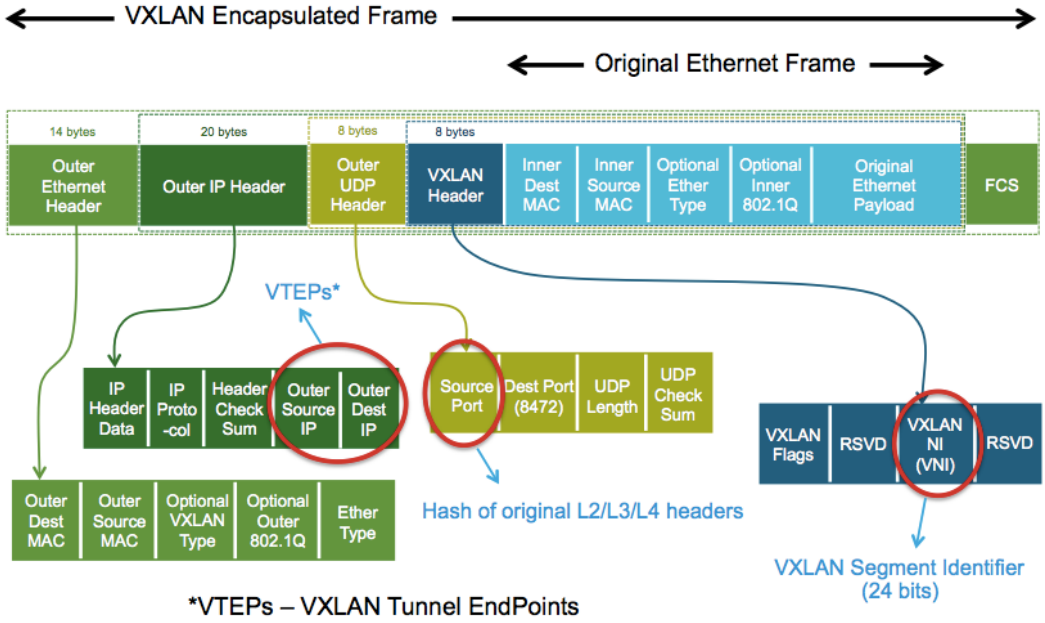


Figure 8 - VXLAN Encapsulation

Few considerations about the VXLAN packet format shown above:

- 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 (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" 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 potentially available inside the transport network infrastructure.
- Up to the current release 6.1, 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 at the link below:

  http://tools.ietf.org/html/rfc7348#page-19

  It is however possible to modify this value in an NSX deployment via a REST API call. However, before doing so and to avoid data-plane outages it is essential to ensure that all the ESXi hosts are running an NSX-v release (no backward compatibility with hosts running vCNS with vSphere 5.1) and that the configuration in the physical network (access-list, firewall policies, etc.), 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 obviously increases the size of the IP packet. For this reason, the recommendation is to increase to a minimum of 1600 Bytes the overall MTU size for all the 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), as will be discussed in more detail later on in this paper.

Figure 9 below describes at high level the steps required to establish L2 communication between virtual machines connected to different ESXi hosts leveraging the VXLAN overlay functionalities.

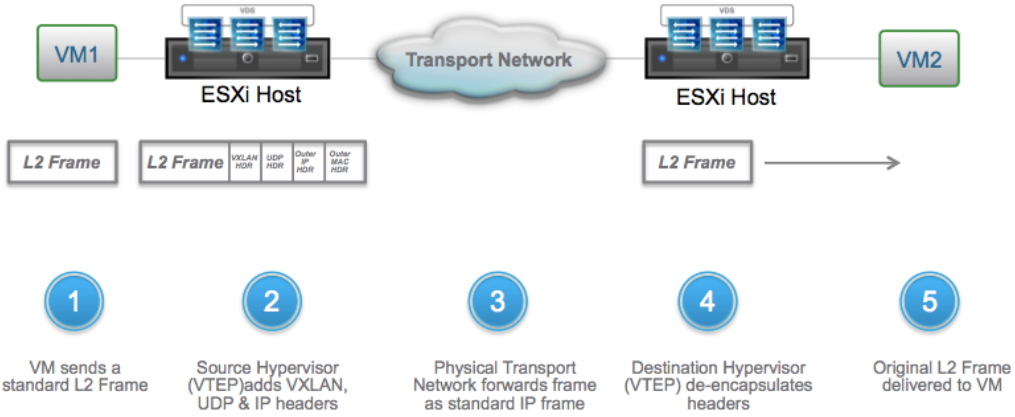


Figure 9 - L2 Communication Leveraging VXLAN Encapsulation

1. VM1 originates a frame destined to the VM2 part of the same L2 logical segment (IP subnet).
2. 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.
3. The transport network is only required to enable IP communication between the source and destination VTEPs.
4. 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).
5. The frame is delivered to the 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 the specific VXLAN control and data plane enhancements provided by NSX-v.

ESXi Hypervisors with VDS

As already mentioned in the introductory section of this paper, the Virtual Distributed Switch (VDS) is a building block for the overall NSX-v architecture. VDS is available on all VMware ESXi hypervisors, so it is now important to clarify their control and data plane interactions inside the NSX architecture.

Note: 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

User Space and Kernel Space

As shown in Figure 10, each ESXi host has a User space and a Kernel space.

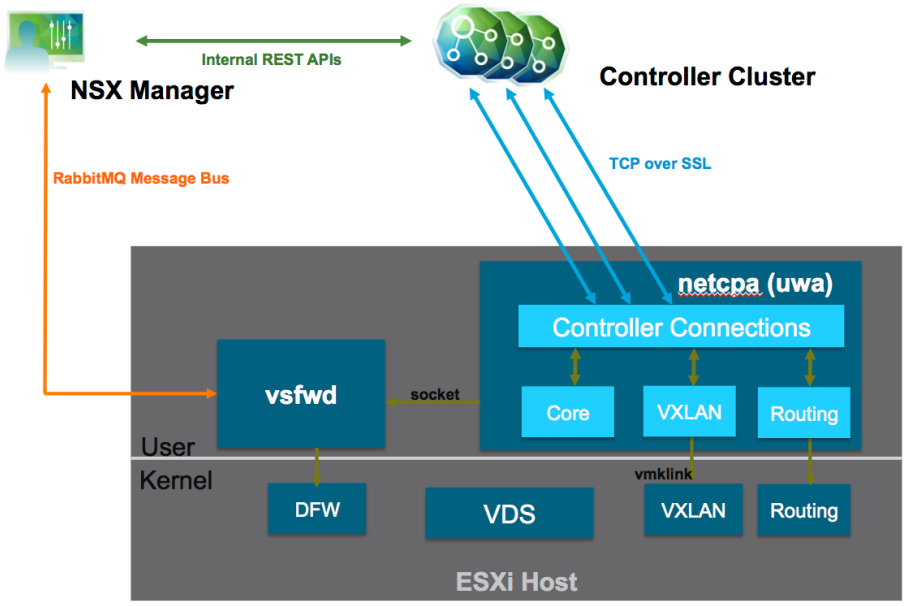


Figure 10 - ESXi Host User and Kernel Space Diagram

VMware Installation Bundles (VIBs) are collection of files that get installed in the Kernel space of the ESXi hypervisor to enable the functionalities requested in the NSX-v architecture: distributed switching and routing, distributed firewalling and VXLAN encapsulation/decapsulation.

In the User space are the 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 an 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: policy rules that need to be programmed on the distributed firewall in the kernel, controller nodes IP addresses, private key and host certificate to authenticate the communication between host and controller 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. Leveraging this control-plane channel with the ESXi hypervisors, the controllers nodes populate local tables (MAC address table, ARP table and VTEP table) to keep track of were the workloads are connected in the deployed logical networks.

NSX Edge Services Gateway

The NSX Edge can really be considered a sort of “swiss army knife”, because of the multiple services it can provide to the NSX-v architecture.

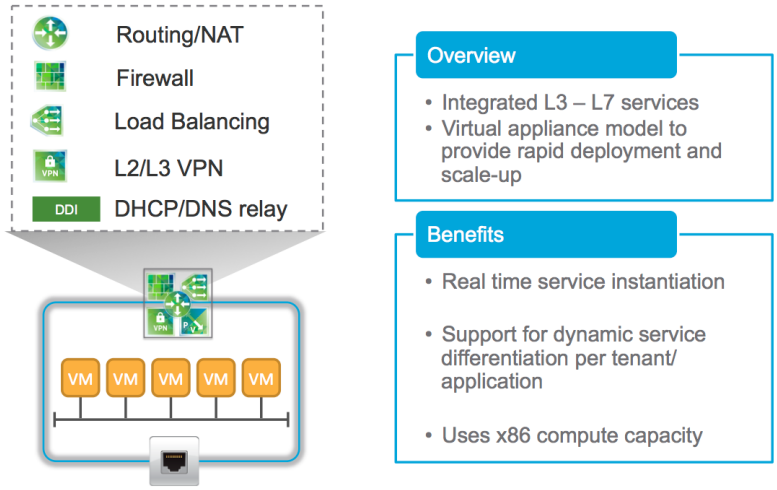


Figure 11 - Services Provided by NSX Edge

Figure 11 highlights the logical services provided by the NSX Edge:

- Routing and Network Address Translation (NAT): 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 routing protocols (OSPF, iBGP and eBGP) and can also communicate leveraging static routing. NAT can be performed for traffic flowing through the Edge and both source and destination NAT is supported.
- Firewall: the NSX Edge supports stateful firewalling capabilities, which complement the Distributed Firewall (DFW) enabled in the Kernel of the ESXi hosts. While the DFW is mostly utilized to enforce security policies for communication between workloads connected to logical networks (the so called east-west communication), the firewall on the NSX Edge is usually filtering communications between the logical space and the external physical network (north-south traffic flows).
- Load Balancing: the NSX Edge can perform load-balancing services for server farms of workloads deployed in 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 also provides VPN capabilities, both at L2 and L3. L2 VPN is usually positioned to extend L2 domains between separate DC sites that are geographically dispersed in support of use cases like compute resources bursting or hybrid cloud deployments. L3 VPNs can either be deployed for allowing IPSec site-to-site connectivity between two NSX Edges or other vendors VPN terminators, or to permit remote users to access the private networks behind the NSX Edge leveraging an SSL VPN connection.
- DHCP, DNS and IP address management (DDI): finally, the NSX Edge can also support DNS relay functionalities and act as a DHCP server providing IP addresses, default gateway, DNS servers and search domains information to workloads connected to the logical networks.

  Note: NSX release 6.1 introduces support for DHCP Relay on the NSX Edge, so that a centralized and remotely located DHCP server can be used to provide IP addresses to the workloads connected to the logical networks.

From a deployment perspective, the NSX Manager offers different form factors to be used depending on the functionalities that need to be enabled, as shown in Figure 12.

X-Large
6 vCPU
8192 MB vRAM

Suitable for high performance Firewall + Load Balancer + Routing

Quad-Large
4 vCPU
1024 MB vRAM

Suitable for high performance Firewall

Large
2 vCPU
1024 MB vRAM

Compact
1 vCPU
512 MB vRAM

Figure 12 - NSX Edge Form Factors

Notice that the form factor can be changed from NSX Manager (via UI or API calls) even after its initial deployments. However, a small service outage is expected in that scenario.

Finally, for what concerns resiliency and high-availability, the NSX Edge Services Gateway can be deployed as a pair of Active/Standby units.

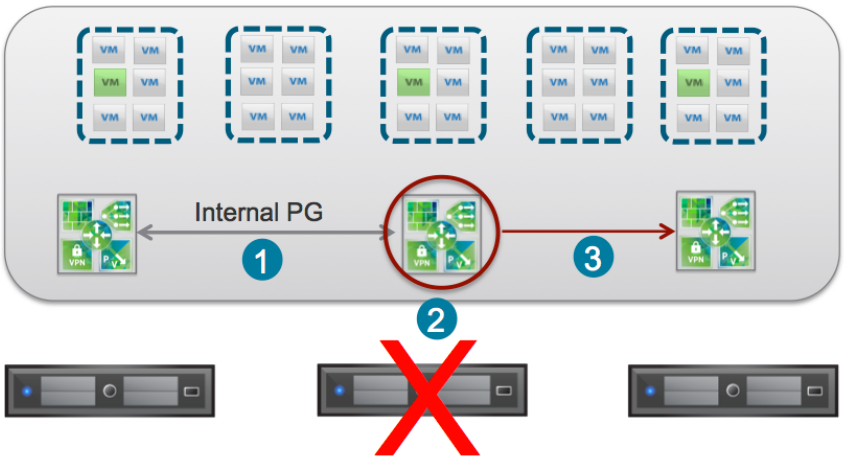


Figure 13 – NSX Edge Active/Standby Deployment

NSX Edge HA is based on the behavior highlighted in Figure 13:

1. NSX Manager deploys the pair of NSX Edges on different hosts (anti-affinity rule). 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 (leveraging the first vNIC deployed on the NSX Edge), unless the user explicitly select a different interface to be used. Notice that a VXLAN L2 segment can be used to exchange the keepalives, so that it may happen over a routed physical network infrastructure.
2. If the ESXi server hosting the active NSX Edge fails, 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, but it can be tuned down (via UI or API calls) to 6 seconds.
3. The NSX Manager also monitors the state of health of the deployed NSX Edges, so it ensures to restart the failed unit on another ESXi host.

Note: 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 (routing, NAT, firewalling, etc.), providing this way services statefulness.

More deployment considerations for NSX Edge can be found in the “NSX-v Design Considerations” section.

Transport Zone

In the simplest sense, a Transport Zone defines a collection of ESXi hosts that can communicate with each other across a physical network infrastructure. As previously mentioned, this communication happens leveraging one (or more) specific interface defined on each ESXi host and named VXLAN Tunnel EndPoints (VTEPs).

A Transport Zone extends across one or more ESXi clusters and in a loose sense defines the span of logical switches. To understand this better, it is important to clarify the relationship existing between Logical Switch, VDS and Transport Zone.

A VDS can span across a certain number of ESXi hosts, since it is possible to add/remove single ESXi hosts from a specific VDS. In a real life NSX deployment, it is very likely that multiple VDS are defined in a given NSX Domain. Figure 14 shows a scenario where a “Compute-VDS” spans across all the ESXi hosts part of compute clusters, and a separate “Edge-VDS” extends across ESXi hosts in the edge clusters.

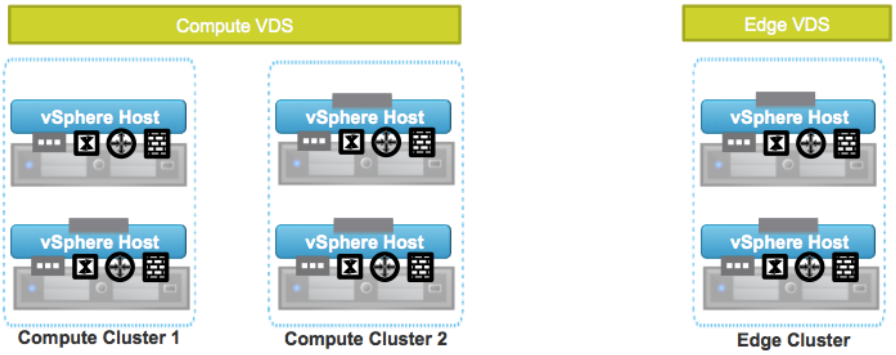


Figure 14 - Separate VDS spanning ESXi Clusters

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 15.

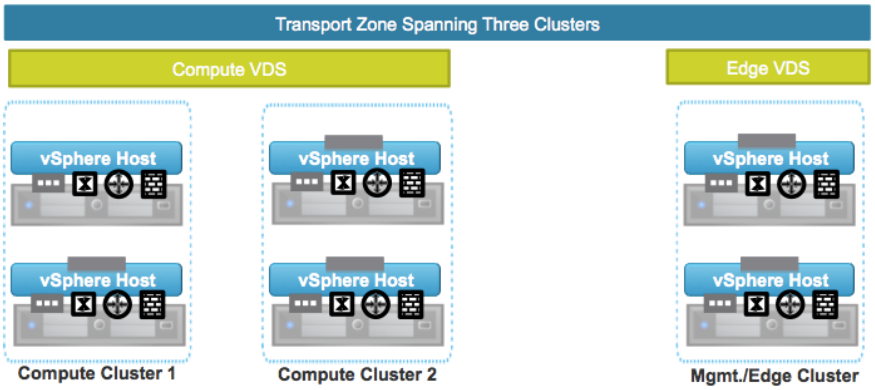


Figure 15 - Transport Zone Spanning VDS and ESXi Clusters

Note: more considerations and recommendations on VDS design in an NSX-v domain can be found in the “VDS Design in an NSX-v Domain” section.

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 as such performs near line rate network traffic protection. DFW performance and throughput scale linearly by adding new ESXi hosts, up to the maximum limit supported by NSX.

DFW is activated as soon as the host preparation process is completed. If a VM needs no DFW service at all, 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 instance, 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 completely different based on "apply to" application: when a DFW rule is created, the user can select a Point of Enforcement (PEP) for this rule: options vary from Logical Switch to vNIC. By default "apply to" is not selected meaning the DFW rule is propagated down to all DFW instances.

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

- L2 rules are mapped to L2 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 (like ARP for instance).
- L3/L4 rules are mapped to L3/L4 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. As a concrete example, if the L2 default policy rule is modified to 'block', then all L3/L4 traffic will be blocked as well by DFW (and ping would stop working for instance).

DFW is an NSX component designed to protect workload-to-workload network traffic (Virtual-to-Virtual or Virtual-to-Physical). In other words, DFW main goal is to protect east-west traffic; that said, since DFW policy enforcement is applied to the vNICs of the VMs, it can 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 (Virtual-to-Physical) and as such is the first entry point to the Software Defined Data Center.

DFW and ESG distinct positioning roles are depicted in the diagram below:

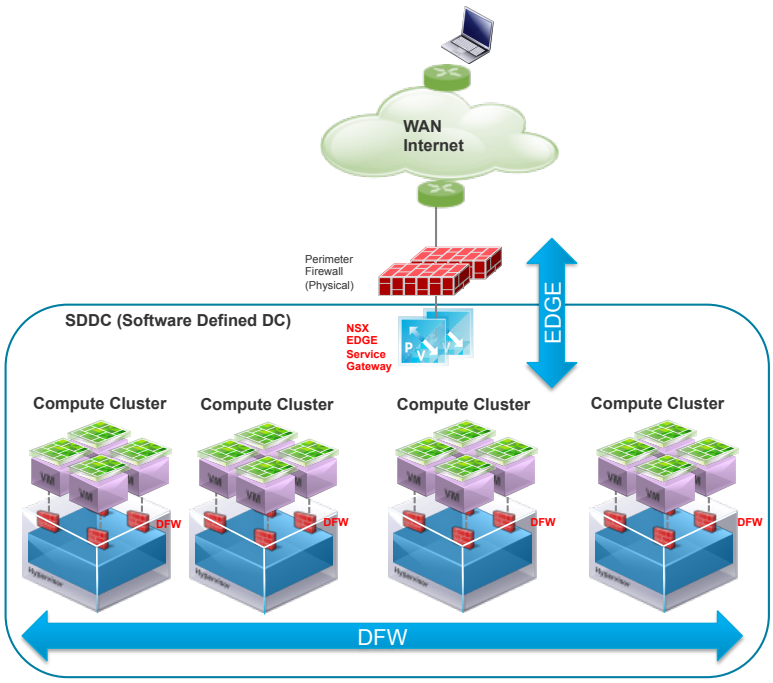


Figure 16 – DFW and Edge Service Gateway Traffic Protection

NSX DFW operates at the VM vNIC level, meaning that a VM is always protected irrespective of the way it is connected to the logical network: VM can be connected to a VDS VLAN-backed port-group or to a Logical Switch (VXLAN-backed port-group). All these connectivity modes are fully supported. ESG Firewall can also be used to protect workloads sitting on physical servers and appliances, such as NAS.

The DFW system architecture is based on 3 distinct entities – each of them playing a clearly defined role:

- **vCenter server** is the management plane of the solution. DFW policy rules are created through 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** is the control plane of the solution. It receives rules from the vCenter server and stores them in the 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 case a CMP is used for security automation.
- **ESXi host** is the data plane of the solution. DFW rules are received from the NSX Manager and then translated into kernel space for real-time execution. VM network traffic is inspected and enforced per ESXi host. Let's take this scenario: 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 (when packets reach to VM2).

  Note: the SSH client depicted in Figure 17 can access the ESXi host CLI to perform specific DFW related troubleshooting.

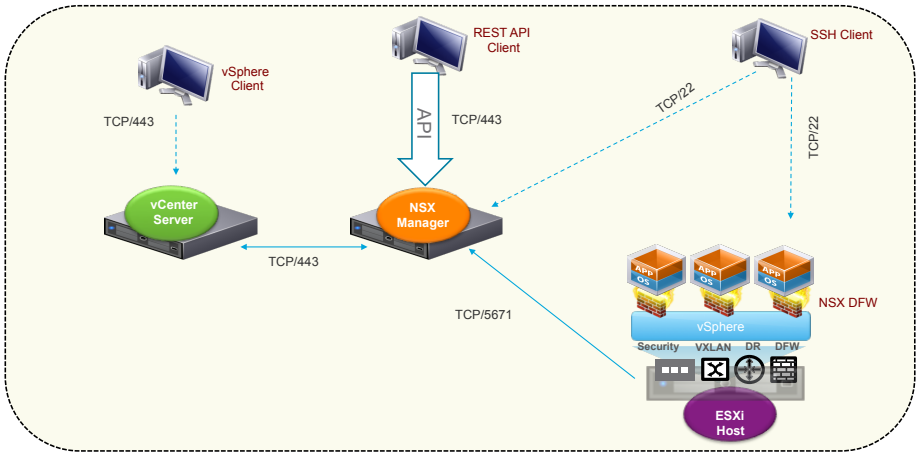


Figure 17 – NSX DFW System Architecture and Communications Channels

It's important to keep in mind that when vCenter containers are used in the DFW policy rules, VMtools must be installed on the guest VMs. VMtools knows about the IP address of the VM (dynamic IP address coming from DHCP or static IP address manually configured on the VM by the administrator) and then provides this information down to the DFW engine that operates based on MAC, IP and TCP/UDP port fields.

Note: if the DFW policy rules use only IP information like host IP or subnet IP or IP sets, then obviously the presence of VMtools on guest VM is not required anymore.

As previously mentioned, the DFW function is activated when a host is prepared. During this operation, a kernel VIB is loaded into the hypervisor: VMware internetworking Service Insertion Platform or **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 and this instance is located between the VM and the Virtual Switch (VDS port-group VLAN-backed or Logical Switch): DVfilter slot 2 is allocated for the DFW instance. All ingress and egress packets to/from this VM have to pass through the DFW and there is absolutely no way to circumvent it.

A set of daemons called vsfwd run permanently on the ESXi host and perform multiple 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 (using the vpxa process on the ESXi host) is only used for vSphere related purposes like VM creation or storage modification and to program host with the IP address of the NSX Manager. This communication is not used at all for any DFW operation.

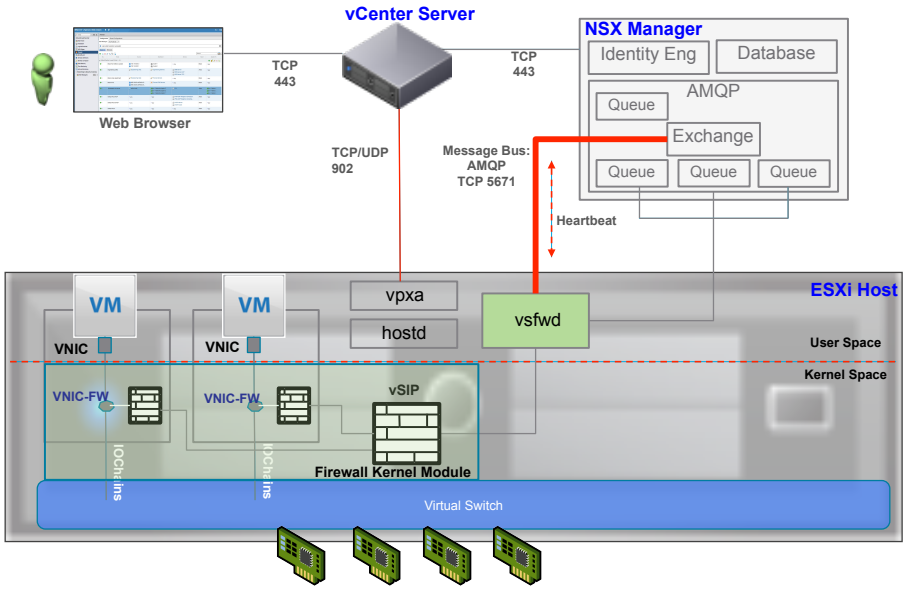


Figure 18 – NSX DFW Components Details

The VSIP kernel module brings in reality more than DFW functionalities. This Service Insertion Platform adds complementary services like Spoofguard and traffic redirection to third party partner services (Palo Alto Networks or Trend or Symantec to name a few).

Spoofguard protects against IP spoofing by maintaining a reference table containing VM name and IP address (this information is retrieved from VMtools when the VM boots for the first time). Spoofguard is inactive by default and must be explicitly enabled per Logical Switch or VDS port-group to become operational. 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.

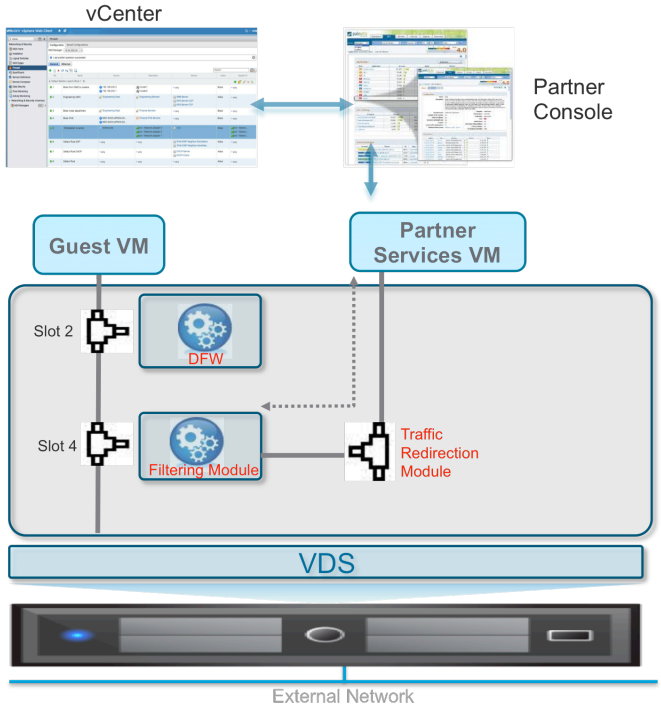


Figure 19 – Traffic Redirection to Third-Party Partner Services

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 Internet to a set of APACHE or TOMCAT servers can be redirected to a L4-L7 deep packet inspection firewall for advanced protection.

Traffic redirection is defined under Service Composer/Security Policy (NSX version 6.0) or under DFW menu, Partner Security Services tab (NSX version 6.1).

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 (i.e. same options for source field, destination field and services field) and 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 and that has been successfully deployed on the platform). Finally, log options can be enabled for this traffic redirection rule.

The DFW instance on an ESXi host (1 instance per VM vNIC) contains 2 separate tables:

- **Rule table**: used to store all policy rules.
- **Connection tracker table**: cache flow entries for rules with permit action.

  Note: a specific flow is identified by the 5-tuple information Source IP address/Destination IP address/protocols/L4 source port/L4 destination port. Notice that 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.

Before exploring the use case for these 2 tables, let's first understand how DFW rules are enforced:

1. DFW rules are enforced in top-to-bottom ordering.
2. Each packet is checked against the top rule in the rule table before moving down the subsequent rules in the table.
3. The first rule in the table that matches the traffic parameters is enforced

Because of this behavior, when writing DFW rules, it is always recommended to put the most granular policies at the top of the rule table. This is the best way to ensure they will be enforced before any other rule.

DFW default policy rule (the one at the bottom of the rule table) is a "catch-all" rule: packet not matching any rule above the default rule will be enforced by the default rule. After the host preparation operation, the DFW default rule is set to 'allow' action. The main reason is because VMware does not want to break any VM to VM communication during staging or migration phases. However, it is a best practice to change the default rule to 'block' action and enforce access controls through a positive control model (only traffic defined in the firewall policy is allowed onto the network).

Let's now have a look at policy rule lookup and packet flow:

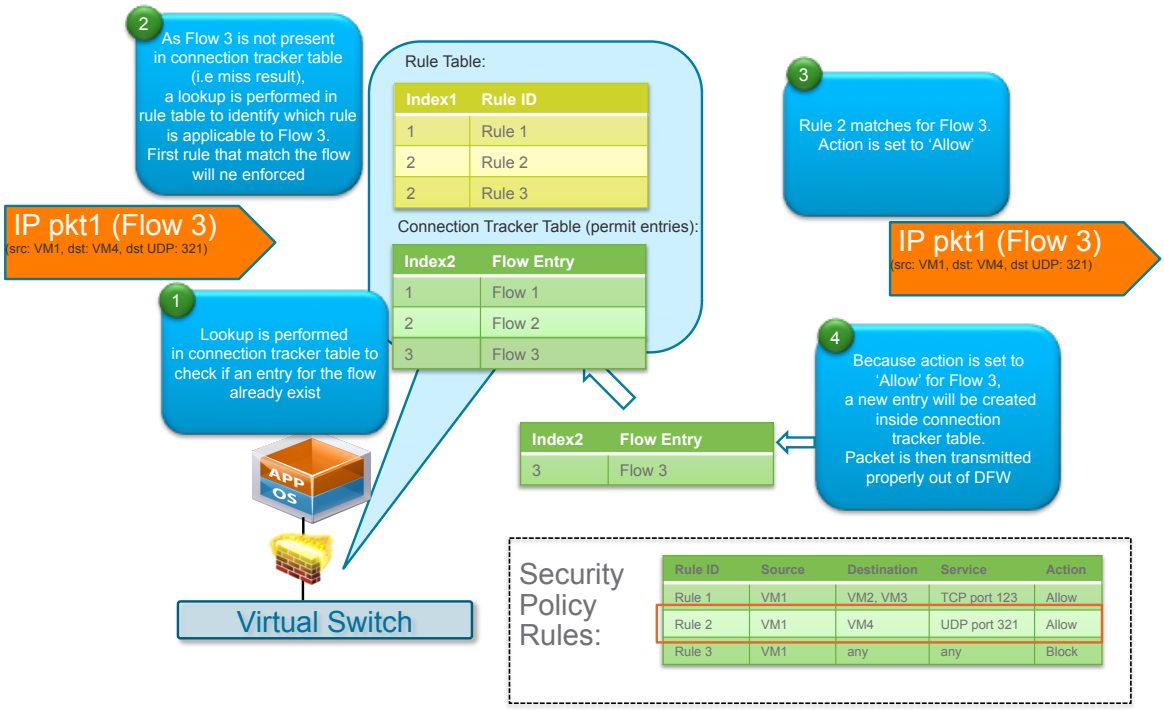


Figure 20 – DFW Policy Rule Lookup and Packet – First Packet

An IP packet (first packet - pkt1) that matches Rule number 2 is sent by the VM. The order of operation is the following:

1. Lookup is performed in the connection tracker table to check if an entry for the flow already exists.
2. As Flow 3 is not present in the connection tracker table (i.e miss result), a lookup is performed in the rule table to identify which rule is applicable to Flow 3. The first rule that match the flow will be enforced.
3. Rule 2 matches for Flow 3. Action is set to 'Allow'.
4. Because action is set to 'Allow' for Flow 3, a new entry will be created inside the connection tracker table. The packet is then transmitted properly out of DFW.

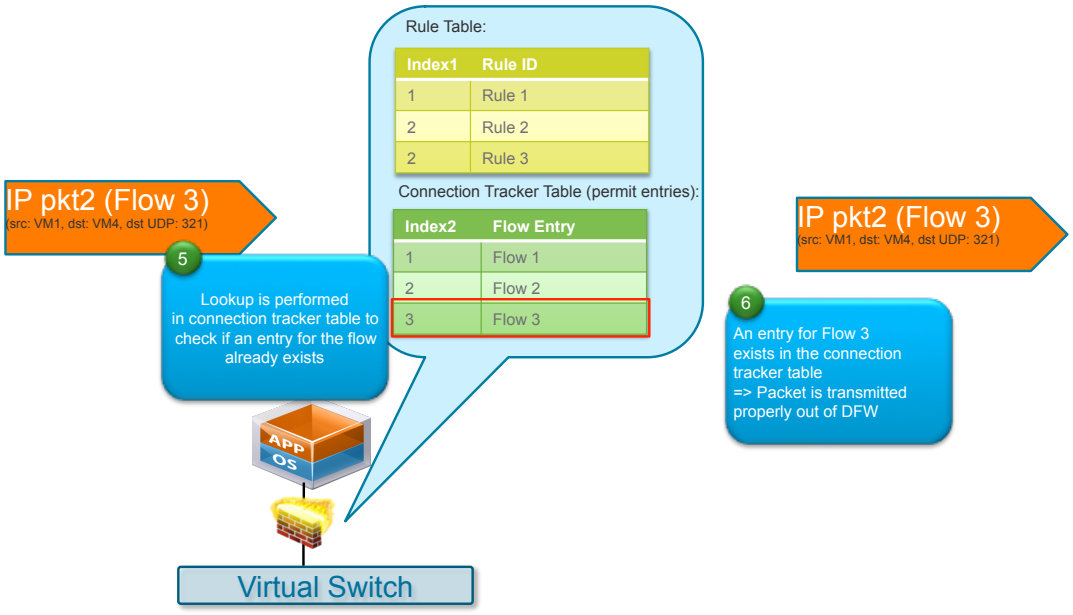


Figure 21 – DFW policy rule lookup and packet – subsequent packets.

Subsequent packets are processed in this order:

1. Lookup is performed in the connection tracker table to check if an entry for the flow already exists.
2. An entry for Flow 3 exists in the connection tracker table => Packet is transmitted properly out of DFW

One important aspect to emphasize is that DFW fully supports vMotion (automatic vMotion with DRS or manual vMotion). The rule table and the connection tracker table always follow the VM during vMotion operation. The positive result is there is no traffic disruption during workload moves and connections initiated before vMotion remain intact after the vMotion is completed. DFW brings VM movement freedom while ensuring continuous network traffic protection.

Note: this functionality is not dependent of Controllers or NSX Manager being up and available.

NSX DFW brings a paradigm shift that was not possible before: security services are no longer dependent on the network topology. With DFW, security is completely decoupled from logical network topology.

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

With NSX DFW, this is no longer needed 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 (workload to workload) can be enforced if VMs are located on same Logical Switch (or VDS VLAN-backed port-group) or on different Logical switches.

This key concept is the foundation for the **microsegmentation** use cases that will be described in the next chapter of this paper.

NSX-v Functional Services

In the context of the NSX architecture, the logical networking functionalities (switching, routing, security, etc.) can be thought of 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-v does not rely on a single transport view 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 hence 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.

Multi-Tier Application Deployment Example

The deployment of the NSX-v Components described in the previous section is paramount to the agile and flexible creation of applications with their required network connectivity and services. A typical example is the creation of a multi-tier application, highlighted in Figure 22.

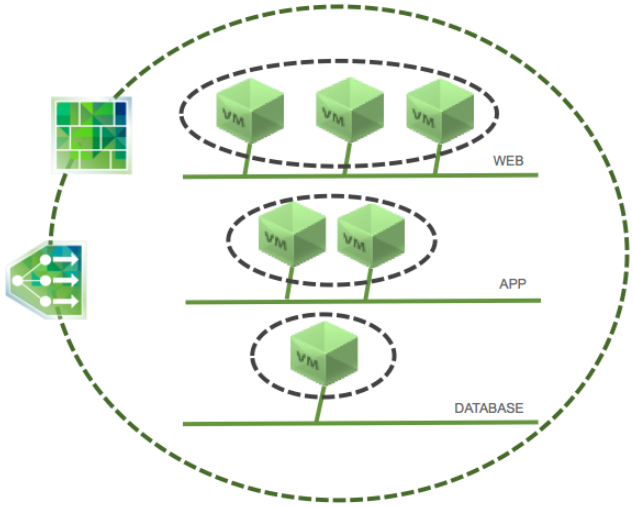


Figure 22 - Deployment of a Multi-Tier Application

The following sections will discuss the functionalities (logical switching, routing and other services) required for the dynamic creation of a multi-tier application that are enabled by the deployment of the NSX-v components.

Logical Switching

The Logical Switching capability in the NSX-v platform provides customers the ability to spin up isolated logical L2 networks with the same flexibility and agility, as it is to spin up virtual machines. Endpoints, both virtual and physical, can then connect to those logical segments and establish connectivity independently from the specific location where they are deployed in the data center network. This is possible because of the decoupling between network infrastructure and logical networks (i.e. underlay and overlay networks) provided by NSX network virtualization.

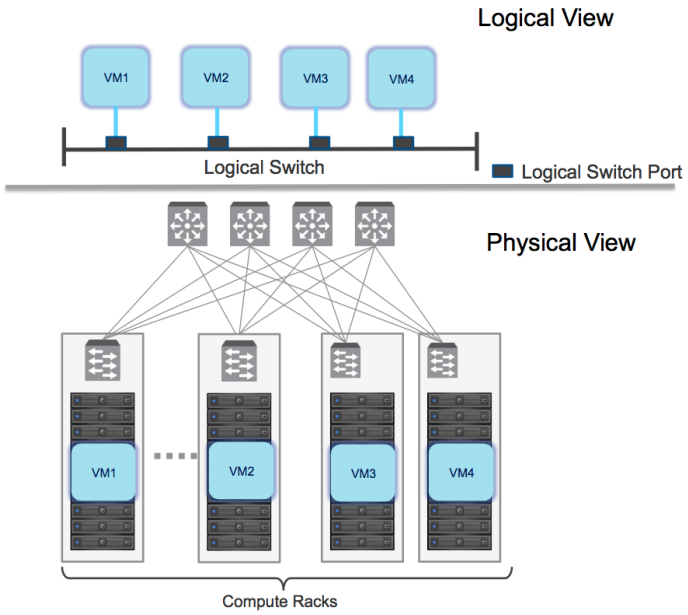


Figure 23 - Logical Switching (Logical and Physical Network Views)

Figure 23 displays the logical and physical network views when logical switching is deployed leveraging the VXLAN overlay technology that allows stretching a L2 domain (logical switch) across multiple server racks, independently from the underlay inter-rack connectivity (L2 or L3).

With reference to the example of the deployment of the multi-tier application previously discussed, the logical switching function allows to create the different L2 segments mapped to the different tiers where the various workloads (virtual machines or physical hosts) are connected.

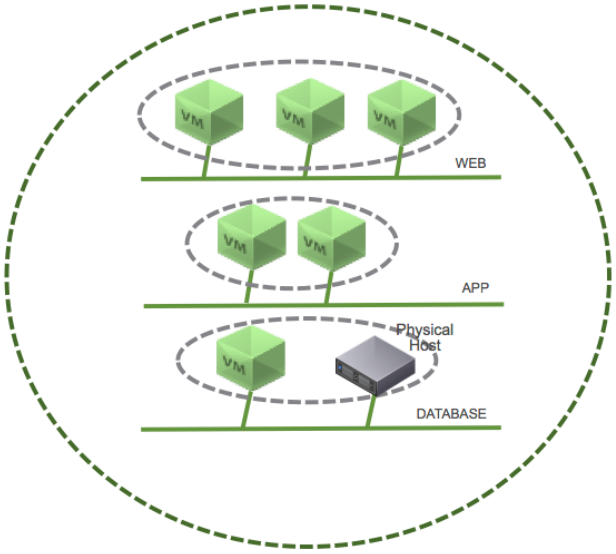


Figure 24 - Creation of Logical Switches for the App Tiers

It is worth noticing that logical switching functionality must enable both virtual-to-virtual and virtual-to-physical communication in each segment and that the use of NSX VXLAN-to-VLAN bridging is also required to allow connectivity to the logical space to physical nodes, as it is often the case for the DB tier.

Replication Modes for Multi-Destination Traffic

When two VMs connected to different ESXi hosts need to communicate directly between them, unicast VXLAN encapsulated traffic is exchanged between the VTEP IP addresses associated to the two hypervisors. Sometimes,

traffic originated by a VM needs to be sent to all the other VMs belonging to the same logical switch. This is specifically the case for three types of L2 traffic:

- Broadcast
- Unknown Unicast
- Multicast

Note: usually we refer to these types of multi-destination traffic types using the acronym BUM (Broadcast, Unknown Unicast, Multicast).

In any of the three scenarios mentioned above, traffic originated by a given ESXi host needs to be replicated to multiple remote hosts (hosting other VMs part of the same logical network). NSX supports three different replications 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. However this can be overridden on a given Logical Switch basis.

Before discussing more in detail those 3 replication modes, it is important to introduce the concept of "VTEP Segment".

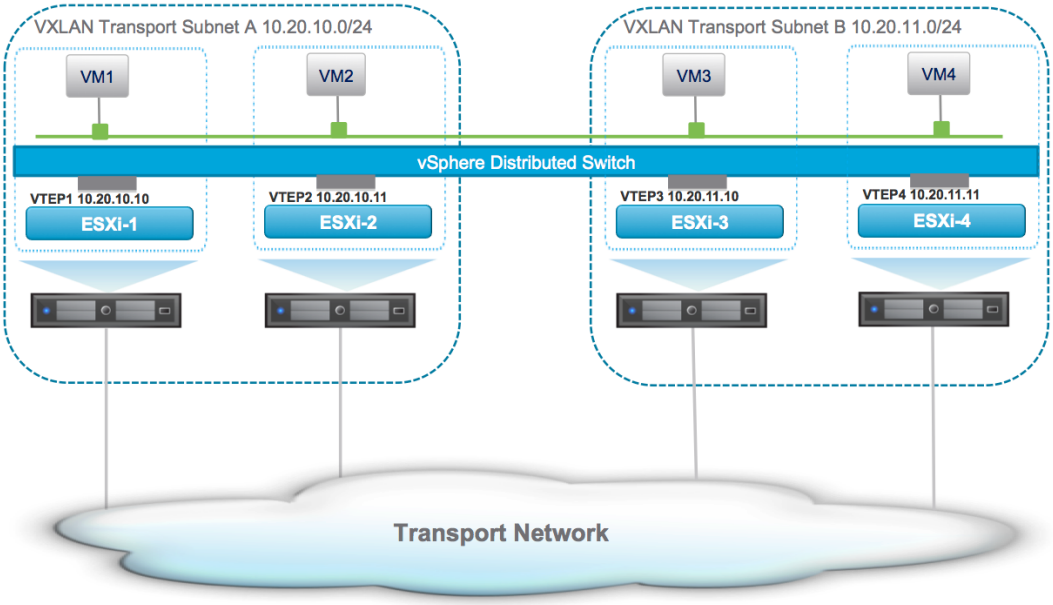


Figure 25 - VTEP Segments

In the example shown in Figure 25 there are four ESXi hosts belonging to two separate "VTEP Segments". 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). The definition of VTEP Segment is paramount to be able to clearly understand the replication modes that are discussed in the sections below.

Multicast Mode

When Multicast replication mode is chosen for a given Logical Switch, NSX relies on the Layer 2 and Layer 3 Multicast capability of the data center physical network to ensure VXLAN encapsulated multi-destination traffic is sent to all the VTEPs. Multicast Mode is the way of 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 (as a matter of fact in this mode the controller is not involved at all). This behavior does not allow to leverage the decoupling effect between logical and physical networking, since 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 (Logical Switch). L2 multicast capability is used to replicate traffic to all VTEPs in the local segment (VTEP IP addresses are part of the same IP subnet) and 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 must enable L3 multicast routing.

In the example in Figure 26 below, the VXLAN segment 5001 has been associated to the multicast group 239.1.1.1. As a consequence, as soon as the first VM gets connected to that logical switch, the ESXi hypervisor hosting the VM

generates an IGMP Join message to notify the physical infrastructure its interest in receiving multicast traffic sent to that specific group.

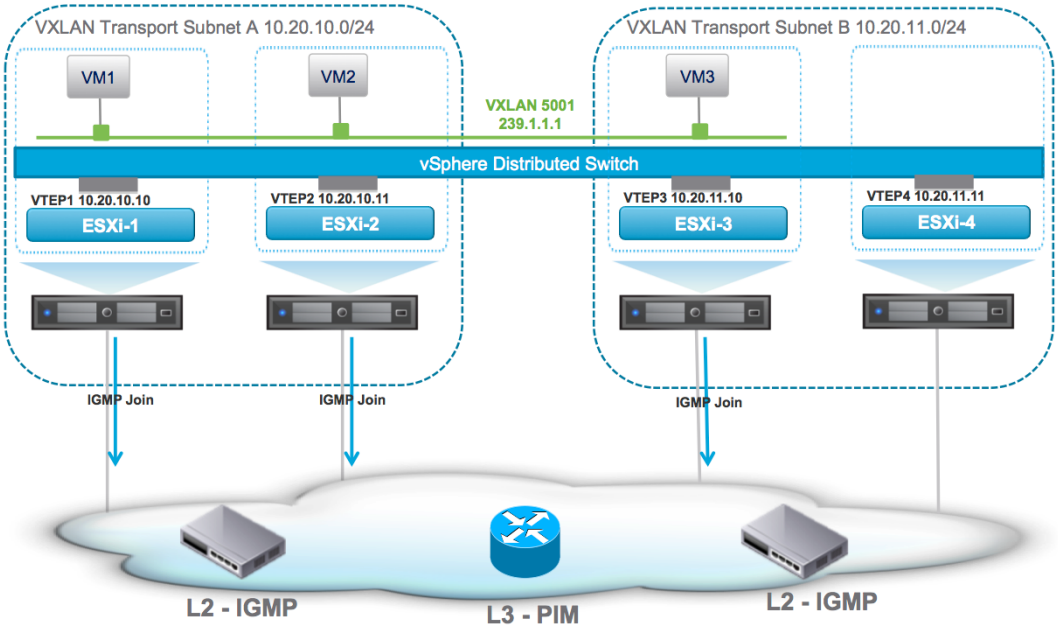


Figure 26 – IGMP Joins

As a result of the IGMP Joins sent by ESXi1-ESXi-3, multicast state is built in the physical network to ensure delivery of multicast frames sent to the 239.1.1.1 destination. Notice that ESXi-4 does not send the IGMP Join since it does not host any active receivers (VMs connected to the VXLAN 5001 segment).

The sequence of events required to deliver a BUM frame in multicast mode is depicted in Figure 27.

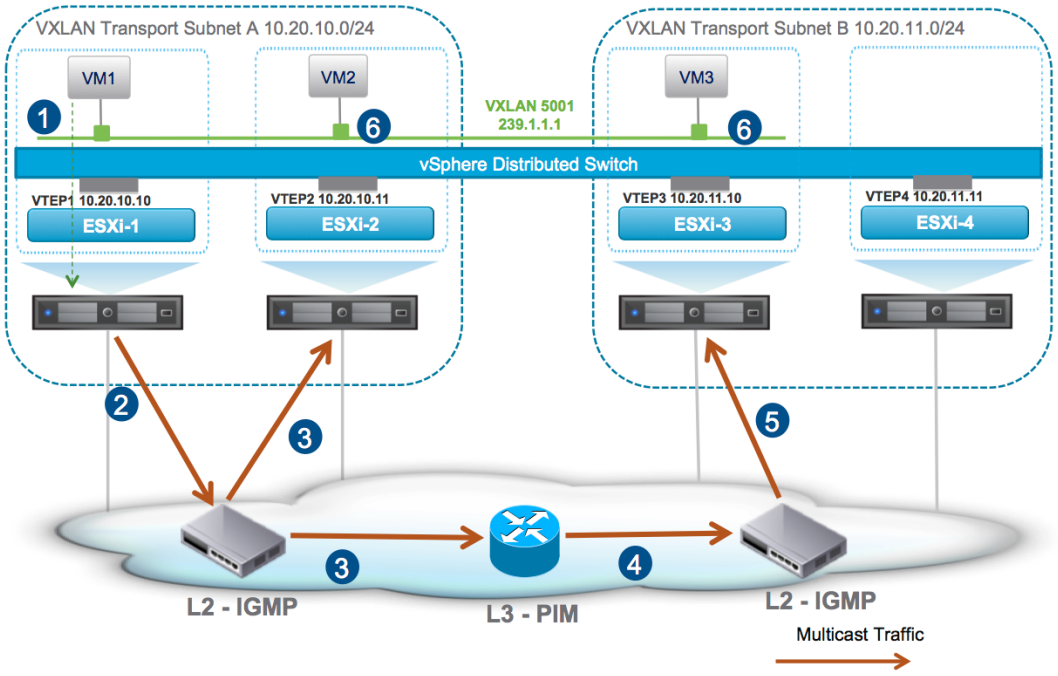


Figure 27 - Replication in Multicast Mode

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 other words, ESXi-1 acts as a source for the multicast stream 239.1.1.1.
3. The L2 switch receiving the multicast frame performs replication: assuming IGMP Snooping is configured on the switch, it will be able to replicate the frame only to the relevant interfaces connecting to ESXi-2 and the L3 router. If IGMP Snooping is not enabled or not supported, the L2 switch treats the frame as a L2 broadcast packet and replicates it to all the 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 delivered to VM2 and VM3.

One important choice to make when configuring multicast mode is 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, as 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. On the downside, this option may increase significantly the amount of multicast state that needs to be built in the physical network devices, and it is critical to determine what is the maximum number of groups those platforms can support.
- On the other side of the spectrum is the option of leveraging a single multicast group for all the defined VXLAN segments. This reduces drastically the amount of multicast state in the transport infrastructure, but may cause unnecessary traffic to be received by the ESXi hosts. Still referring to the example in Figure 27, ESXi-4 would receive traffic belonging to VXLAN 5001 as soon as a local VM gets connected to any logical segment (even different from VXLAN 5001).
- In most cases, the decision is to deploy a m:n mapping ratio as a trade-off scenario between the two options described above. 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. This means that VXLAN segment “1” and “n+1” will be using the same group, and so far and so on for the others.

In any case, multicast mode is presented in this paper just to build the baseline knowledge required to better understand and appreciate the enhancements made possible by NSX-v with the other two options (unicast and hybrid modes).

Unicast Mode

Unicast Mode represents a completely opposite approach than multicast mode, where the decoupling between logical and physical networks is fully achieved. In Unicast Mode, the ESXi hosts in the NSX domain are divided in separate groups (VTEP segments) based on the IP subnet their VTEP interfaces belong to. 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 the ESXi hypervisor hosting the VM sourcing the traffic and belonging to a different VTEP segment to all the ESXi hosts part of its segment (i.e. 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 the local segment that have at least one VM actively connected to the logical network where multi-destination traffic is sent to. 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 28 below illustrates the Unicast Mode replication mechanism.

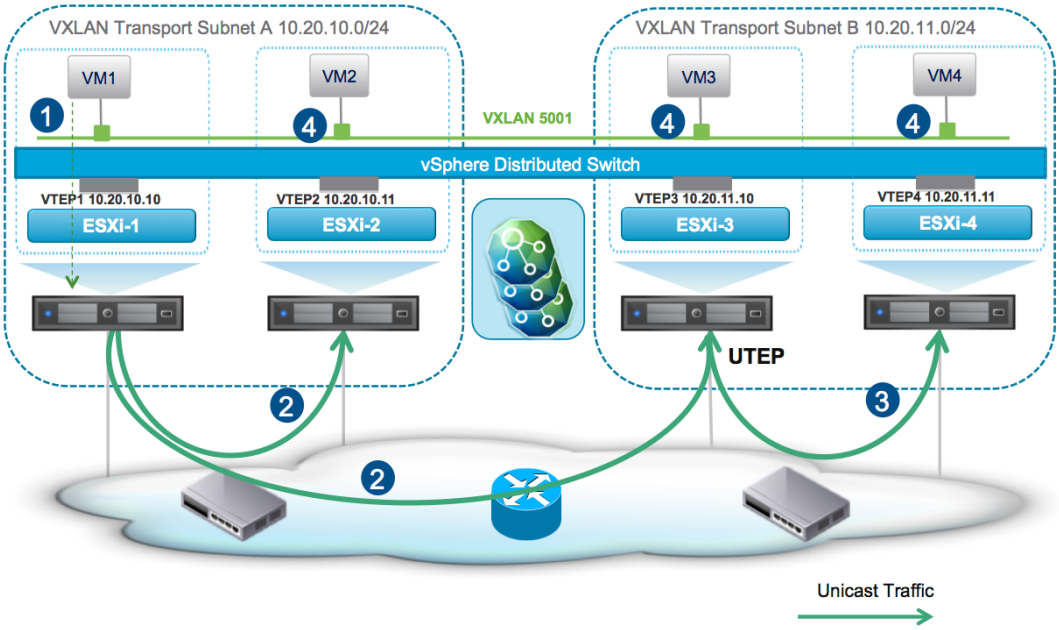


Figure 28 – Unicast Mode

1. In this example we have 4 VMs connected to Logical Switch 5001 and VM1 generates a BUM frame to be sent to all those VMs. Notice how in this scenario there is no need to specify a multicast group associated to this VXLAN segment.
2. ESXi1 looks at its local VTEP table filled with the information received via control plane communication with the controller nodes and determines the need to replicate the packet only to the other VTEP belonging to the local segment (ESXi2) and to the UTEP part of remote segments (ESXi3 since only one remote VTEP segment is shown in this example). The unicast copy sent to the UTEP is characterized by having 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, look at its 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 (ESXi-4 only in this example).

Still referring to the example above, if it was VM2 to generate a BUM frame, the unicast replication would be performed by the ESXi-2, following the same steps described above. ESXi-2 may elect a different host (for example ESXi-4) as remote UTEP, since 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 need any explicit configuration on the physical network to enable distribution of multi-destination VXLAN traffic. This mode is very well suited for smaller deployments with fewer VTEPs per segment and few physical segments. However this mode is not recommended for large scaled environments, specifically where VMs generate a good amount of BUM traffic, as the overhead of replication increases with the number of segments.

Hybrid Mode

Hybrid Mode offers operational simplicity similar to Unicast Mode (no IP Multicast Routing configuration required in the physical network) while leveraging the Layer 2 Multicast capability of physical switches.

This is illustrated in the example in Figure 29, where it can also be noticed how the specific VTEP responsible for performing local replication to the other VTEPs part of the same subnet is now named "MTEP". The reason is that in Hybrid Mode the [M]TEP uses L2 [M]ulticast to replicate BUM frames locally, whereas the [U]TEP does that leveraging [U]nicast frames.

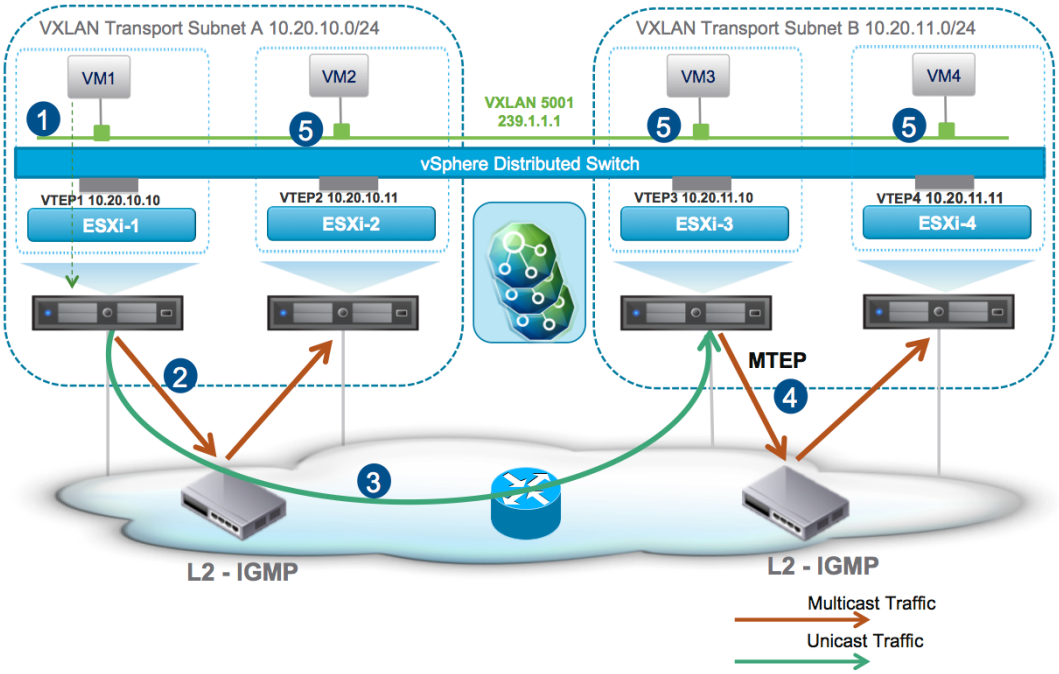


Figure 29 - Hybrid Mode Logical Switch

1. VM1 generates a BUM frame that needs to be replicated to all the other VMs 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 (similarly to what previously shown in Figure 26). Since PIM is not required (hence usually not configured), it is strongly recommended to define an "IGMP Querier" per VLAN to ensure successful L2 multicast delivery and avoid not deterministic behavior.

   Note: when IGMP snooping is enabled, but there's no "IGMP Querier" definition, some types of Ethernet switches will resort to flooding the multicast traffic in the VLAN, 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 (ESXi3 since only one remote segment is shown in this example). The unicast copy sent to the MTEP with the 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.

Note: similarly to what already discussed for Unicast Mode, if it were VM2 to generate a BUM frame, the ESXi-2 host would perform the unicast replication to remote MTEPs. Once again, each ESXi host locally determine what ESXi hosts belonging to remote VTEP segments are acting as MTEP: ESXi-1 may use ESXi-3 as remote MTEP (as shown in figure above), whereas ESXi-4 may be used as remote MTEP by ESXi-2.

Populating the Controller Tables

After discussing how multi-destination traffic is handled in an NSX-v deployment, let's now concentrate on layer 2 unicast communications. The focus here is on NSX-v deployments fully leveraging the NSX controller nodes; hence, the first thing to discuss is the control plane communication between the ESXi hosts and the controller cluster to populate the VTEP, MAC and ARP tables on the controller nodes.

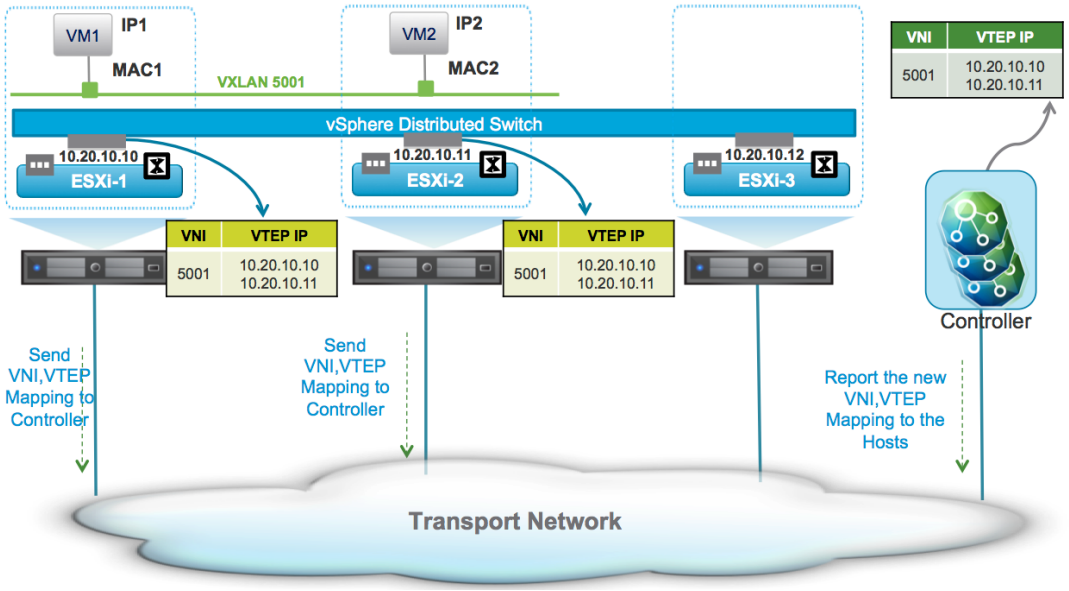


Figure 30 - VNI-VTEP Report

As shown in Figure 30, when the first VM connects to a VXLAN segment (VM1 on ESXi1 and VM2 on ESXi-2 in this example), the ESXi hosts generate a control plane message to the controller (or better to the specific controller node in charge of that specific logical switch slice) with the VNI/VTEP mapping information. The controller node populates its local VTEP table with this information and it sends a report message to all the ESXi hypervisors hosting VMs actively connected to that same VXLAN segment (that is the reason why the message is not sent to ESXi-3 in our example). The ESXi hosts can then populate their local VTEP tables; as previously mentioned, this info can for instance be leveraged to determine the list of VTEPs to which replicate multi-destination traffic.

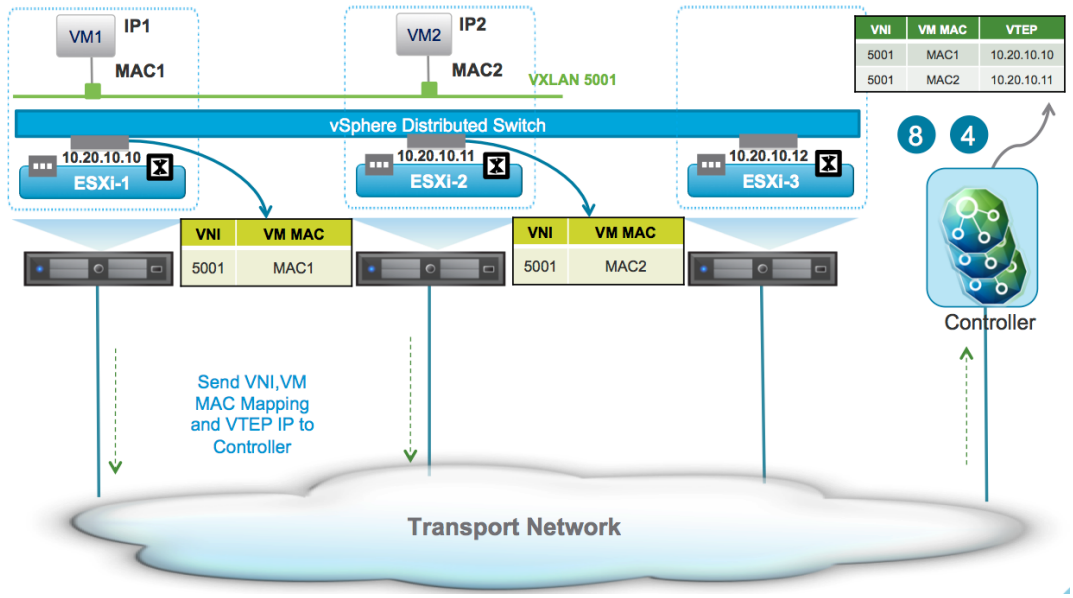


Figure 31 – VNI-MAC Address Report

The second piece of information reported by the ESXi hosts to the controller is the MAC address for VMs locally connected to a specific VNI. The controller uses those reports to populate its local MAC table, but dissimilar to the VNI-VTEP report, it does not send this information to all the ESXi hosts. That is the reason why the ESXi hosts at this point are only aware of the locally connected MAC addresses, as highlighted in Figure 31 above.

Finally, the last piece of information needed by the controller is the IP address of the VMs, so to be able to populate its local ARP table that will be used to perform the ARP suppression functionality that will be presented in the next section while discussing unicast communication.

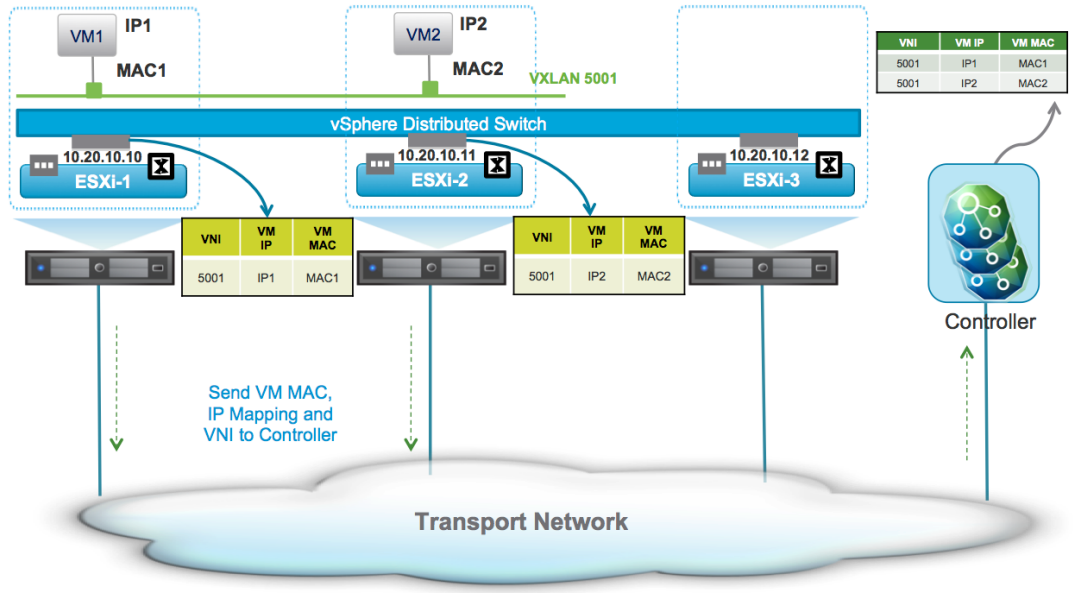


Figure 32 - VNI-IP Address Report

The ESXi hosts learn the IP address of locally connected VMs in two main ways:

1. For VMs getting an IP address using DHCP, the ESXi host will be able to glean the IP address by snooping the DHCP response sent by the DHCP server and destined to the local VMs.
2. 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 that use the info to populate its local ARP table.

Unicast Traffic (Virtual to Virtual Communication)

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

Figure 33 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 (VXLAN segment).

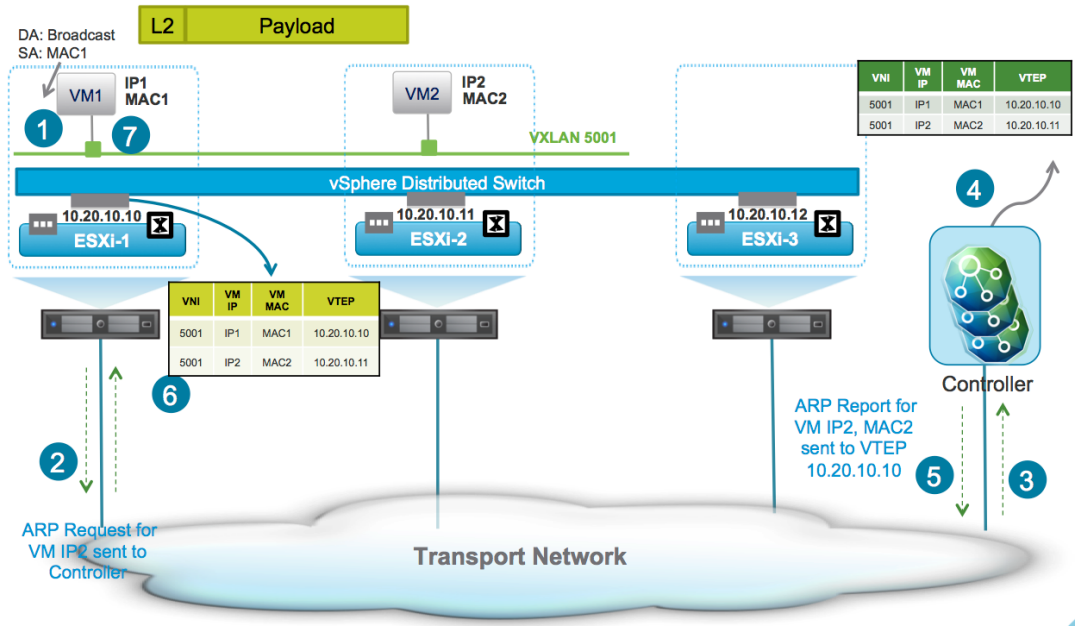


Figure 33 – ARP Resolution via NSX Controller

1. VM1 generates an ARP request (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, so 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 is essential because VM1 does not have to know how the ARP reply was handled, from its point of view it came directly from VM2.

Note: if the controller does not have the mapping information, it will notify ESXi-1 that will then resort to flooding 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 34.

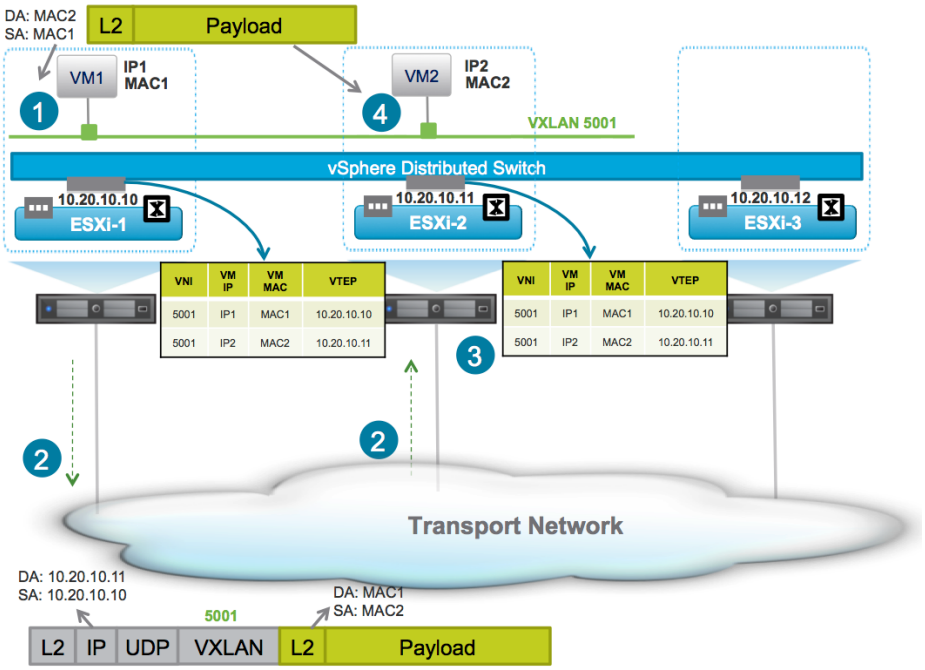


Figure 34 - Intra Logical Switch Unicast Communication

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 (control plane learning), so 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 while decapsulating it, it 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). This is an example of data plane learning for a VM specific location.
4. The frame is delivered to VM2.

Note: now that both ESXi-1 and ESXi-2 have populated their local tables, traffic can flow in both directions.

Unicast Traffic (Virtual to Physical Communication)

There are several circumstances where it may be required to establish L2 communication between virtual and physical workloads. Some typical scenarios are (not exhaustive list):

- Deployment of multi-tier applications: in some cases, the Web, Application and Database tiers can be deployed as part of the same IP subnet. Web and Application tiers are typically leveraging virtual workloads, but that is not the case for the Database tier where bare-metal servers are commonly deployed. As a consequence, it may then be required to establish intra-subnet (intra-L2 domain) communication between the Application and the Database tiers.
- Physical to virtual (P-to-V) migration: many customers are virtualizing applications running on bare metal servers and during this P-to-V migration it is required to support a mix of virtual and physical nodes on the same IP subnet.
- Leveraging external physical devices as default gateway: in such scenarios, a physical network device may be deployed to function as default gateway for the virtual workloads connected to a logical switch and a L2 gateway function is required to establish connectivity to that gateway.
- Deployment of physical appliances (firewalls, load balancers, etc.).

To fulfill the specific requirements listed above, it is possible to deploy devices performing a “bridging” functionality that enables communication between the “virtual world” (logical switches) and the “physical world” (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 (VXLAN to VLAN ID mapping), even if the hypervisor running the VM is not physically connected to that L2 physical network.

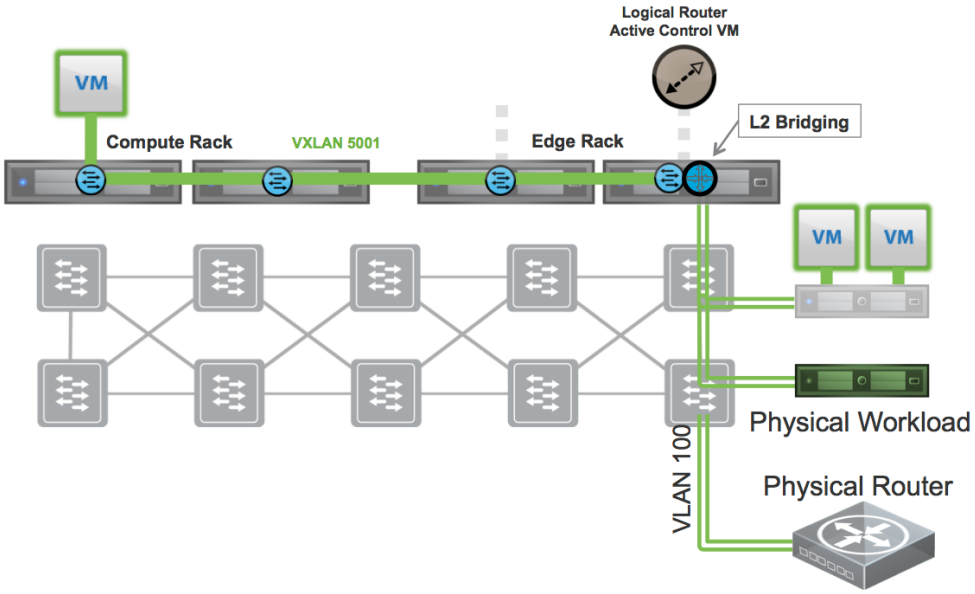


Figure 35 - NSX L2 Bridging

Figure 35 shows an example of L2 bridging, where 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 (in VLAN 100). In the current NSX-v implementation, the VXLAN-VLAN bridging configuration is part of the distributed router configuration; the specific ESXi hosts performing the L2 bridging functionality is hence the one where the control VM for that distributed router is running. In case of failure of that ESXi host, the ESXi hosting the standby Control VM (which gets activated once it detects the failure of the Active one) would take the L2 bridging function.

Note: for more information on distributed routing and the role of the Control VMs, please refer to the following "Logical Routing" section.

Independently from the specific implementation details, below are some important deployment considerations for the NSX L2 bridging functionality:

- The VXLAN-VLAN mapping is always performed in 1:1 fashion. This means 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 only on a specific ESXi host.
- However, 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, and not in user space. Once again, the Control VM is only used to determine the ESXi host where a given bridging instance is active, and not to perform the bridging function.

Figure 36 and Figure 37 show the ARP exchange between a virtual machine and a bare-metal server, which is the first step required to be able to provide virtual-to-physical unicast communication.

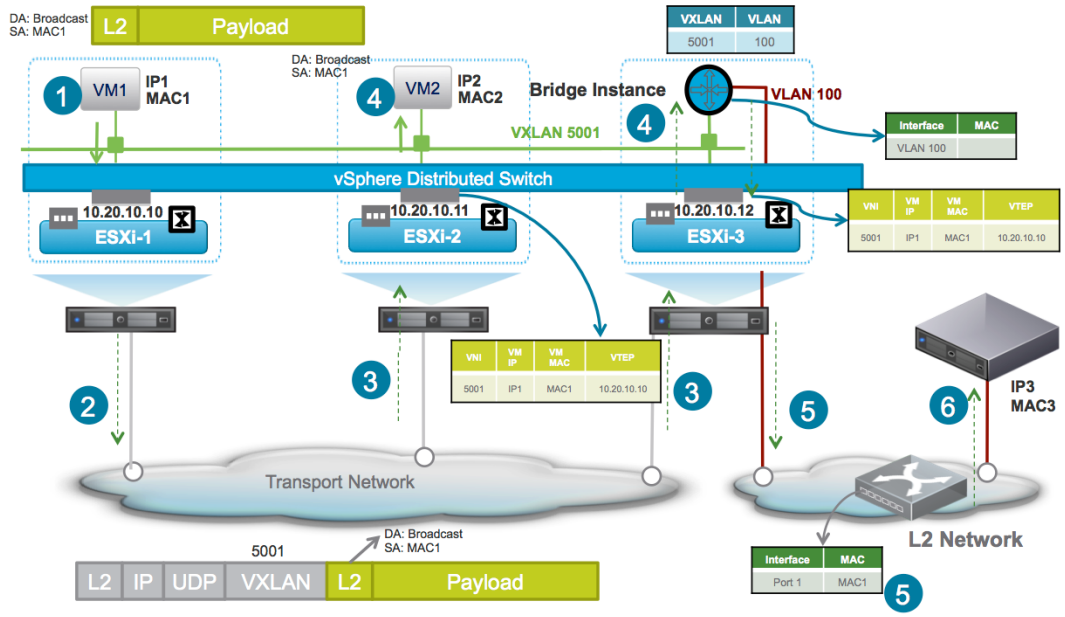


Figure 36 - ARP Request Originated from the Virtual Machine

1. VM1 generates an ARP request to retrieve the MAC/IP mapping for the bare-metal server.
2. As discussed in the previous section, the ESXi hosts intercepts the ARP request and generates a control-plane request directed to the controller to retrieve the mapping information. Let's assume in this scenario that the controller does not have this information (for example because the bare-metal server has just been connected to the network and it has not generated traffic yet). In this case, ESXi-1 must 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 instead 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.

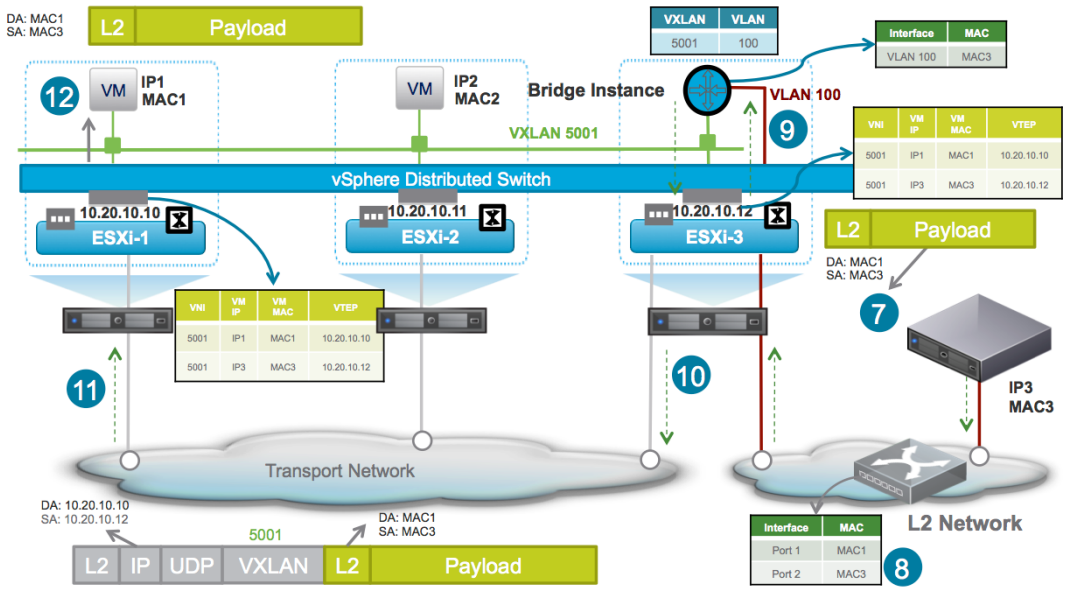


Figure 37 - ARP Response Originated from the Bare-Metal Server

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 as well.
10. ESXi-3 knows where the MAC2 is located because of the learning performed at previous step 3, so 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, 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 (the source MAC address in the frame is MAC3) and delivers it to VM1.

At this point, data plane communication between VM1 and the bare-metal host can commence, in similar fashion to what discussed for unicast virtual to virtual L2 communication.

Logical Routing

The Logical Routing capability in the NSX platform provides customers the ability to interconnect endpoints (virtual and physical) deployed in different logical L2 networks. Once again, this is possible due to the decoupling between network infrastructure and logical networks provided by the deployment of network virtualization.

Figure 38 shows the logical view of the routed topology interconnecting two logical switches and the corresponding physical view.

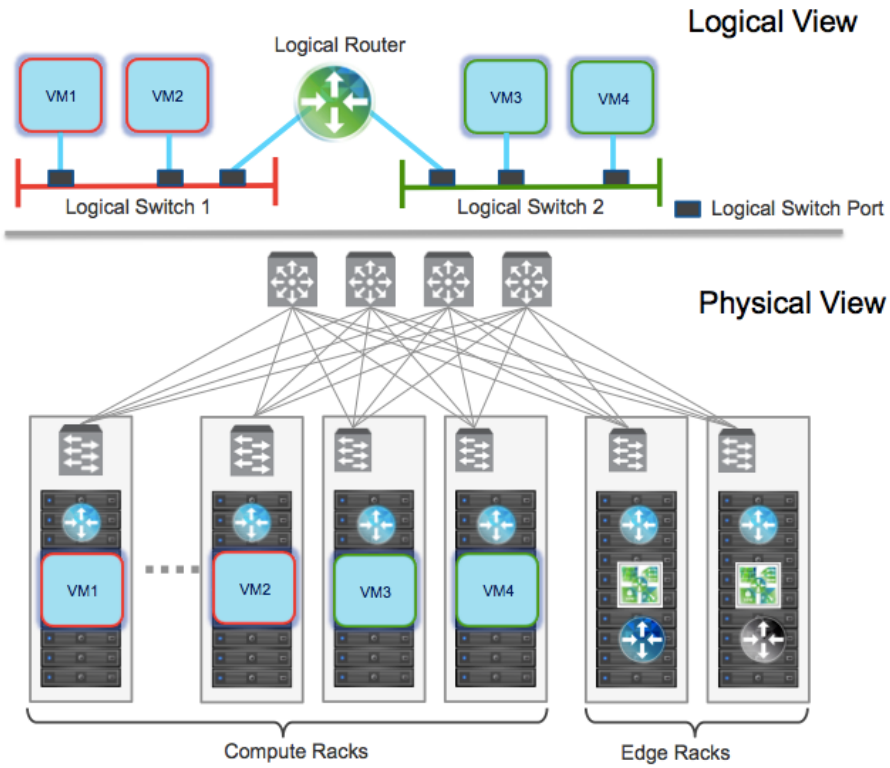


Figure 38 - Logical Routing (Logical and Physical Network Views)

The deployment of Logical Routing can serve two purposes: interconnecting endpoints (logical or physical) belonging to separate logical L2 domains (as shown in figure above) or interconnecting endpoints belonging to logical L2 domains with devices deployed in the external L3 physical infrastructure. The first type of communication, usually confined inside the data center, is referred to as east-west communication; the second one is named north-south communication and provides connectivity into the data center from the external physical world (WAN, Internet or Intranet).

Going back to the recurring example of the deployment of a multi-tier application, logical routing is the functionality required to interconnect between them the different application tiers (east-west communication) and also to provide access to the Web tier from the external L3 domain (north-south communication), as shown in Figure 39 below.

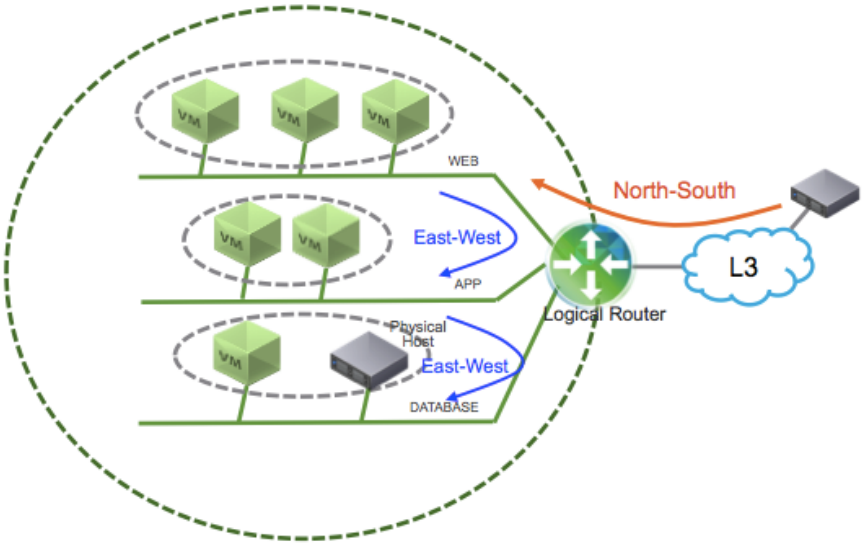


Figure 39 - Logical Routing for a Multi-Tier Application

Logical Routing Components

The two types of logical routing communications discussed above are usually achieved leveraging two different functionalities: centralized routing and distributed routing.

Centralized routing represents the on-ramp/off-ramp functionality that allows communication between the logical network space and the external layer 3 physical infrastructure.

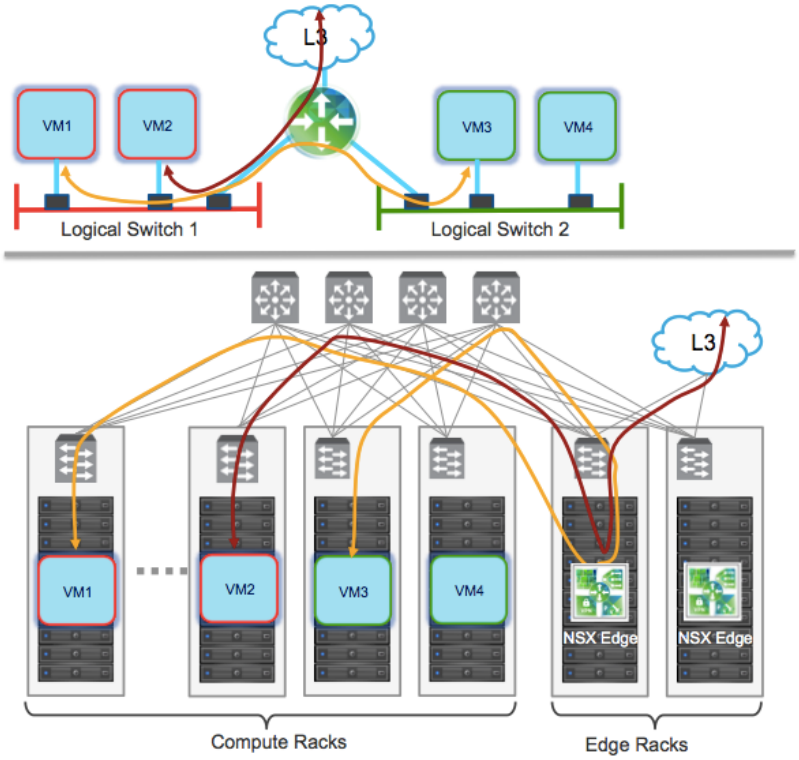


Figure 40 - Centralized Routing

Figure 40 highlights how the NSX Edge Services Gateway provides the traditional centralized routing support in the NSX-v platform. Along with the routing services, the NSX Edge also supports other network services like DHCP, NAT, Firewall, Load Balancing, and VPN.

Notice how centralized routing deployment can be utilized both for east-west and north-south routed communications. However, east-west routed flows are not optimized in a centralized routing deployment, since traffic is always hair-pinned from the compute racks toward the edge rack where the active NSX Edge is deployed. This is the case even when two virtual machines belonging to separate Logical Switches are deployed inside the same hypervisor.

As highlighted in Figure 41, the deployment of distributed routing prevents the “hair-pinning” for VM-to-VM routed communication by providing hypervisor level routing functionality. Each hypervisor installs in the kernel specific flow information (received from the NSX controller) ensuring a direct communication path even when the endpoints belong to separate Logical Switches (IP subnets).

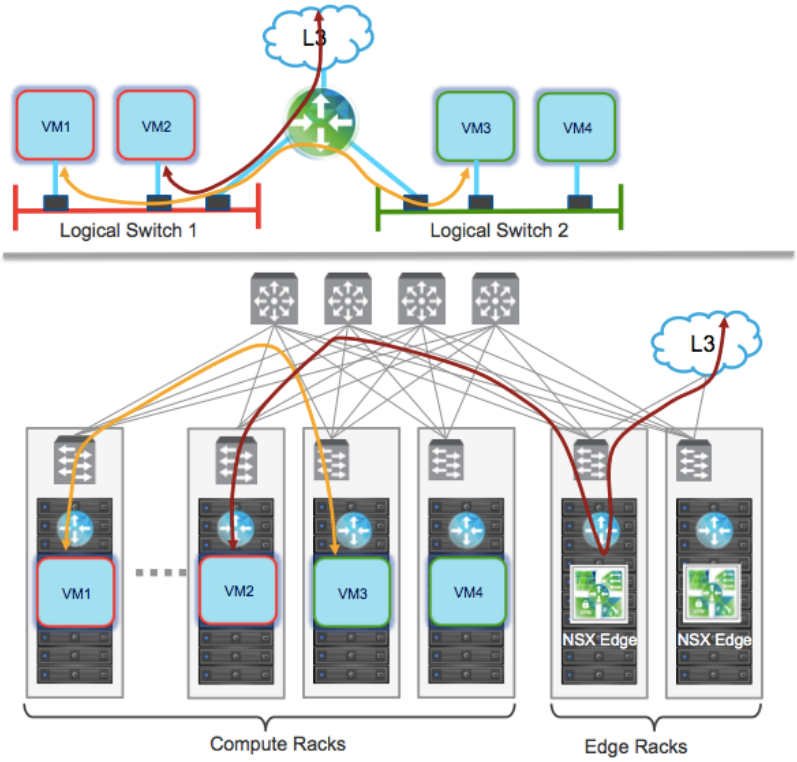
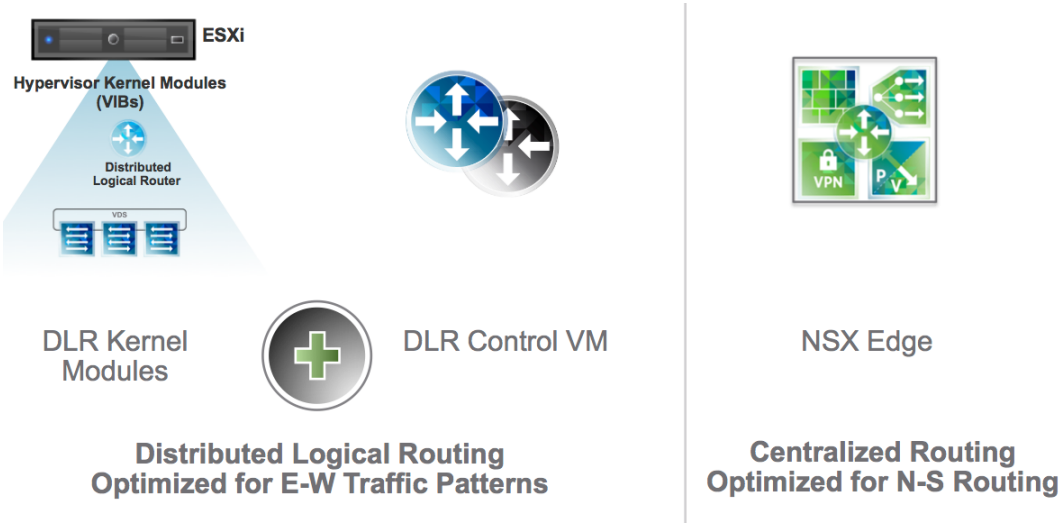


Figure 41 - Distributed Routing

Figure 42 shows the NSX components involved in distributed and centralized routing.



We already discussed the role of the NSX Edge in handling centralized routing and other logical network services. Distributed routing is provided by a logical element called Distributed Logical Router (DLR), which has two main components:

- The DLR control plane is provided by the DLR Control VM: this VM supports dynamic routing protocols (BGP, OSPF), exchanges routing updates with the next layer 3 hop device (usually the NSX Edge) and communicates with the NSX Manager and the Controller Cluster. High-availability for the DLR Control VM is supported through Active-Standby configuration: a pair of virtual machines functioning in active/standby modes are provided, similarly to what previously discussed for the NSX Edge.

- At the data-plane level there are DLR kernel modules (VIBs) that 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. The kernel modules have routing information base (RIB) that is pushed through the controller cluster. All the data plane function of route lookup, ARP entry lookup is performed by the kernel modules. The kernel modules are equipped with logical interfaces (called LIFs) connecting to the different logical switches (and/or VLAN-backed port-groups). Each LIF has assigned an IP address representing the default IP gateway for the logical L2 segment it connects to and a vMAC address. The IP address obviously is unique per LIF, whereas the same vMAC is assigned to all the defined LIFs.

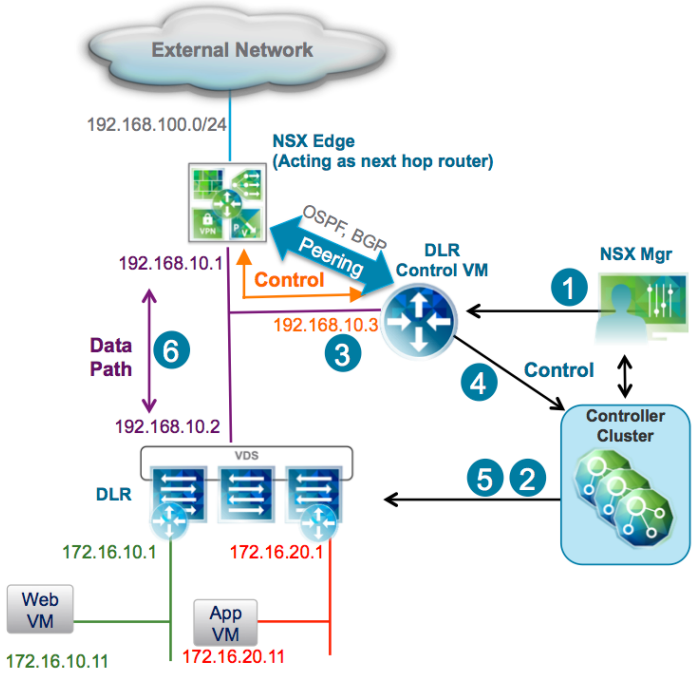


Figure 42 - Logical Routing Components

Figure 42 shows the interaction between all the logical routing components to enable distributed routing.

1. A DLR instance is created from the NSX Manager UI (or via API calls) and routing is enabled, leveraging the protocol of choice (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. Assuming a routing protocol is also enabled on the next hop layer device (an NSX Edge in this example), OSPF/BGP peering is established between the NSX Edge and the DLR control VM. The NSX Edge and the DLR can then exchange routing information:
   - 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). As a consequence, it then pushes those routes advertisements to the NSX Edge. Notice that the next-hop for those prefixes is not the IP address assigned to the Control VM (192.168.10.3) but 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”.
   - The NSX Edge pushes to the Control VM the prefixes to reach IP networks in the external network; in most scenarios a single default route is likely to be sent by the NSX Edge, since it represents the single point of exit toward the physical network infrastructure.
4. The DLR Control VM pushes the IP routes learnt from the NSX Edge to the Controller cluster.
5. The Controller cluster is responsible for distributing routes learned from the DLR Control VM across the hypervisors. Each controller node in the cluster takes responsibility of distributing the information for a particular logical router

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

6. 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 43.

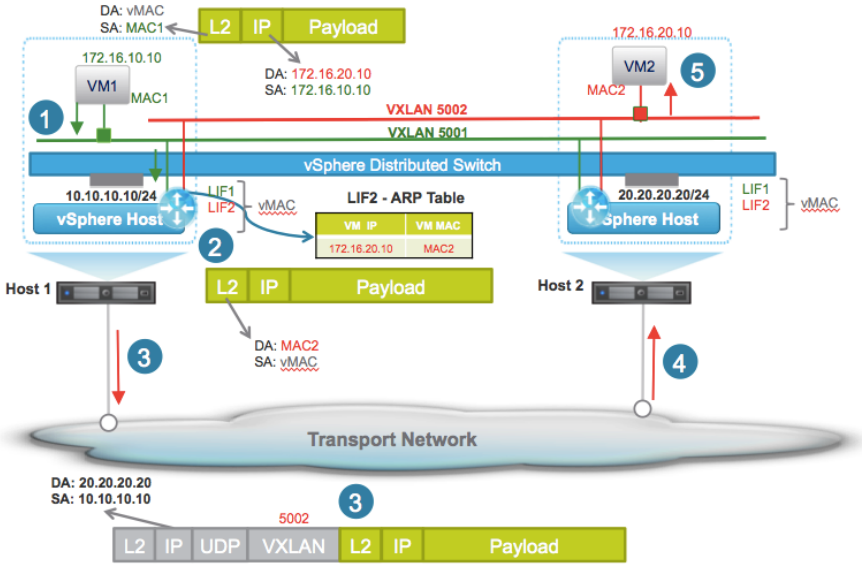


Figure 43 - Routed Communication between Virtual Machines

1. VM1 wants to send a packet to VM2 connected to a different VXLAN segment, so the packet is sent to VM1 default gateway interfaces located on the local DLR (172.16.10.1).
2. A routing lookup is performed at the local DLR, which determines that destination subnet is directly connected to DLR LIF2. A lookup is performed in LIF2 ARP table to determine the MAC address associated to VM2 IP address.

   Note: if the ARP information were not available, the DLR would 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 and the original packet is then VXLAN encapsulated and sent to the VTEP of ESXi2 (20.20.20.20).
4. ESXi-2 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.

Notice that 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 44 shows instead the sequence of steps required for establishing ingress communication from external networks to logical segments connected to the Distributed Logical Router.

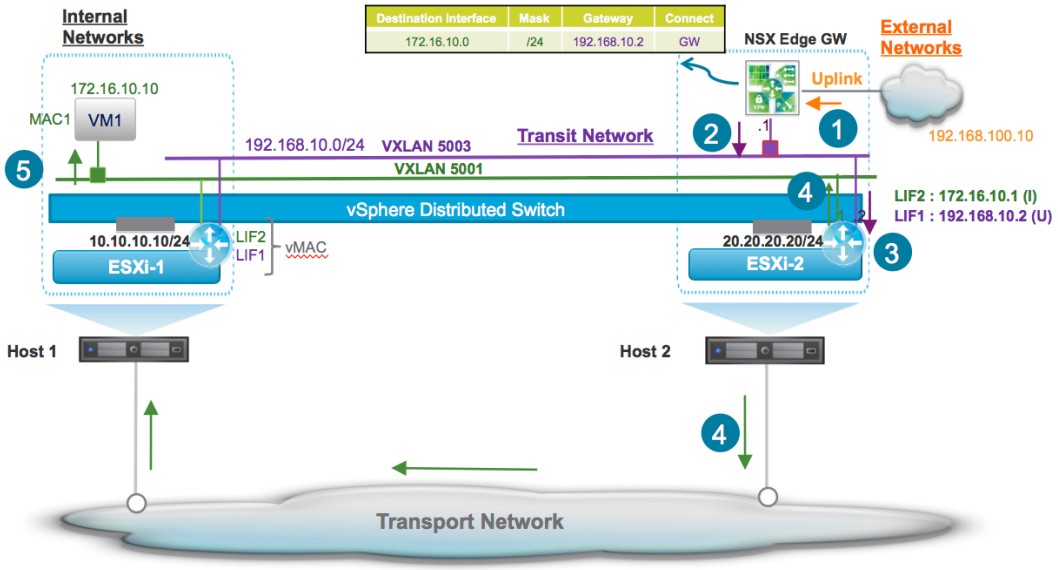


Figure 44 - DLR: Ingress Traffic from External Networks

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 learnt via a routing exchange with the DLR Control VM and the next-hop (192.168.10.2) represents the IP address of the DLR on the data-plane (Forwarding Address).
3. The DLR is also installed in the kernel of ESXi-2, where the Active NSX Edge is deployed. This means both routing pipelines, at the NSX Edge level and at the DLR level, are performed locally on ESXi-2.
4. The destination IP subnet (172.16.10.0/24) is directly connected to the DLR, so packet is routed from the Transit Network into the VXLAN 5001 segment. After a 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.

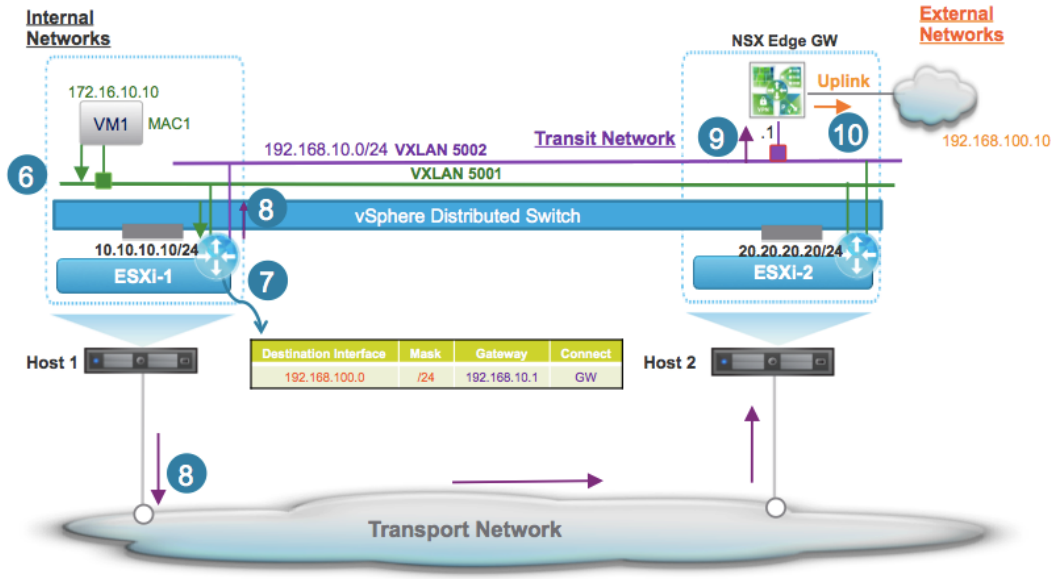


Figure 45 - DLR: Egress Traffic to External Networks

6. VM1 wants to reply to the external destination 192.168.100.10, so 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 toward the destination is the NSX Edge interface on the Transit Network (192.168.10.1). This information was received on the DLR Control VM from the NSX Edge itself 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 and the packet is then 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 and sends the packet into the physical network to the next L3 hop on the path toward the destination. The packet will then be delivered by the physical infrastructure to the final destination 192.168.100.10.

Logical Routing Deployment Options

Depending on the customer requirements various topologies can be built using logical switching and logical routing features of the NSX platform. In this section, we will cover the following two routing topologies that utilizes both distributed and centralized logical routing capabilities:

1) Physical Router as Next Hop
2) Edge Services Gateway as Next Hop

Physical Router as Next Hop

As shown in the diagram below, an organization is hosting multiple applications and wants to provide connectivity among the different tiers of the application as well as connectivity to the external network.

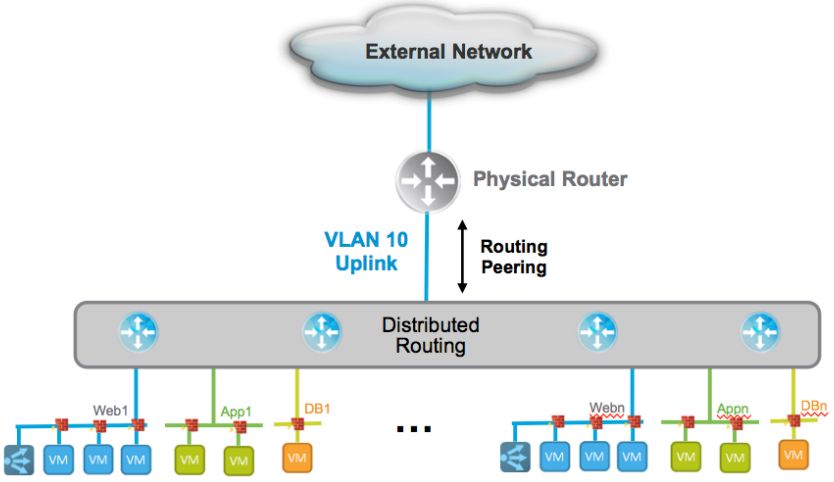


Figure 46 - Physical Router as Next-Hop

In this topology separate logical switches provide layer 2 network connectivity for the VMs in the particular tier and the east-west and north-south routing decisions happen at the hypervisor level in a distributed fashion. The distributed logical routing configuration allows the VMs on two different tiers to communicate with each other. Similarly, the dynamic routing protocol support on the logical router enables the exchange of routes with the physical next-hop router. This in turn allows external users to access the applications connected to the logical switches in the data center.

The downside of this design is that the VLAN used by the DLR to send traffic to the physical router must be made available to all the ESXi hosting compute resources, which may or may not be possible depending on the specific data center fabric design of choice. More considerations about this will be found in the “NSX-v Deployment Considerations” section of this paper.

Edge Services Gateway as Next Hop

The introduction of the NSX Edge Services Gateway between the DLR and the physical router, shown below in Figure 47, allows removing the concern described above.

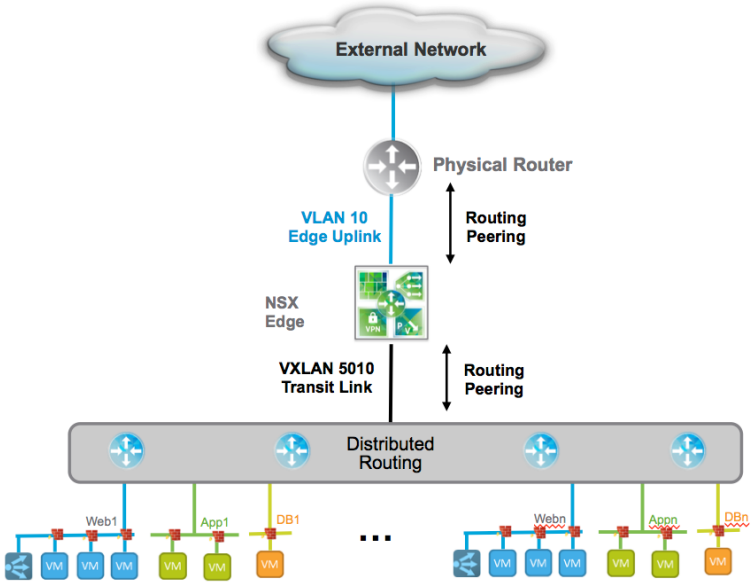


Figure 47 - NSX Edge between DLR and Physical Router

In this deployment model, the DLR Control VM peers with an NSX Edge on a VXLAN Transit Link, whereas 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 allows to remove any dependency from the physical network for what concerns connectivity between the Compute ESXi hosts (where DLR routing happens for outbound traffic) and the NSX Edge, given that the traffic will be VXLAN encapsulated in the data-plane.
- The establishment of routing 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 would not require any further modification to the physical network nor would change the number of routing adjacencies established with the physical routers, fully realizing the decoupling between logical and physical networks promised by NSX.

One concern with this option may be the fact that all the north-south traffic requires to transit through the NSX Edge, and this may represent a constraint in terms of bandwidth. The use of Active/Active ECMP capabilities in NSX Edge (introduced from SW release 6.1) allows scaling-out the available bandwidth for the on-ramp/off-ramp function provided by the NSX-Edge.

Note: for more information about the deployment of Active/Active ECMP NSX Edge, please refer to the “Edge Racks Design” section.

In a service provider environment there are multiple tenants, and each tenant can have different requirements in terms of number of isolated logical networks and other network services such as LB, Firewall, and VPN etc. In such deployments, the NSX Edge Services Gateway provides network services capabilities along with dynamic routing protocol support.

As shown in Figure 48 below, the 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. Also, 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 east-west traffic routing is handled by the distributed router in the hypervisor and the north-south

traffic flows through the NSX Edge Services Gateway.

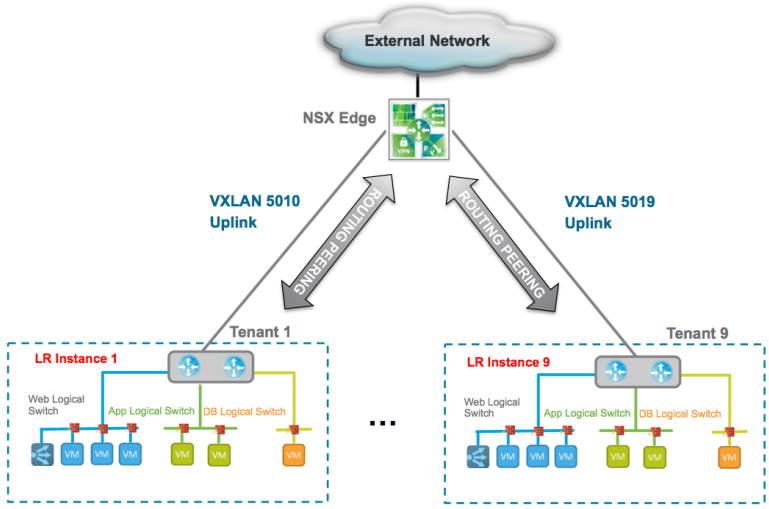


Figure 48 - NSX Edge Services Gateway as Next Hop (also providing network services)

As shown above, this deployment model limits to 9 the number of tenants that can be connected to the same NSX Edge. This limitation derives from the fact that the NSX Edge, as any other VM, is equipped with 10 virtual interfaces (vNICs) and one of them is used as Uplink toward the physical network. Also, in this scenario it is imperative not to have overlapping IP addresses between the 9 tenants that connect to the same NSX Edge, unless the overlapping prefixes are not advertised between the DLR and the NSX Edge and used only in the context of each single tenant.

Scalable Topology

The service provider topology described in the earlier section can be scaled out as shown in the Figure 49. The diagram shows nine tenants served by NSX Edge on the left and the other nine by the Edge on the right. Service provider can easily provision another NSX edge to serve additional tenants.

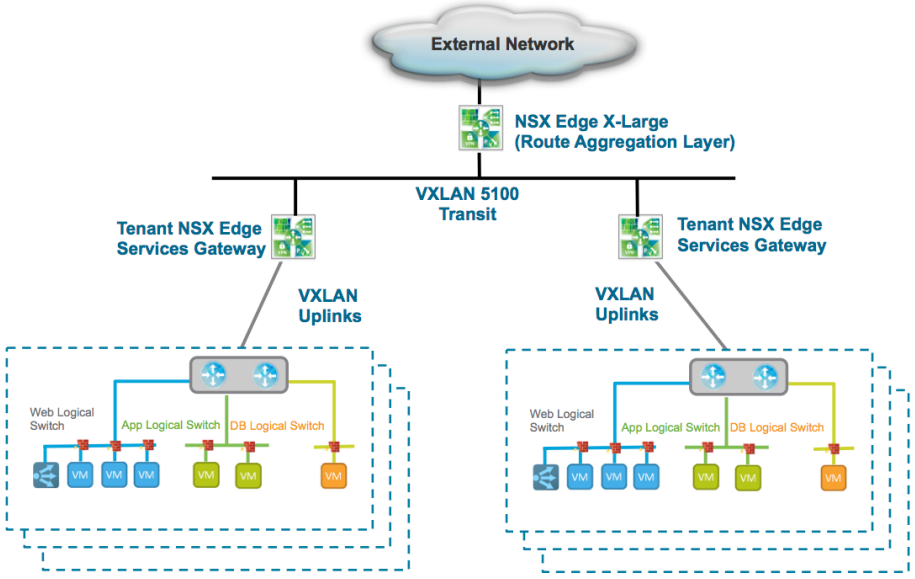


Figure 49 - Scalable Topology

In addition to provide scalability, this model provides the advantage that tenants with overlapping IP addresses can be deployed in separate groups connected to different NSX Edges. NAT must be performed on the first tier of NSX Edges

to support this scenario, in order to ensure that tenants using overlapping IP addresses can be distinguished once their traffic flows toward the aggregation NSX Edge.

Logical Firewalling

The logical firewalling is another functional element to be added for our example of deployment of a multi-tier, as represented in Figure 50 below.

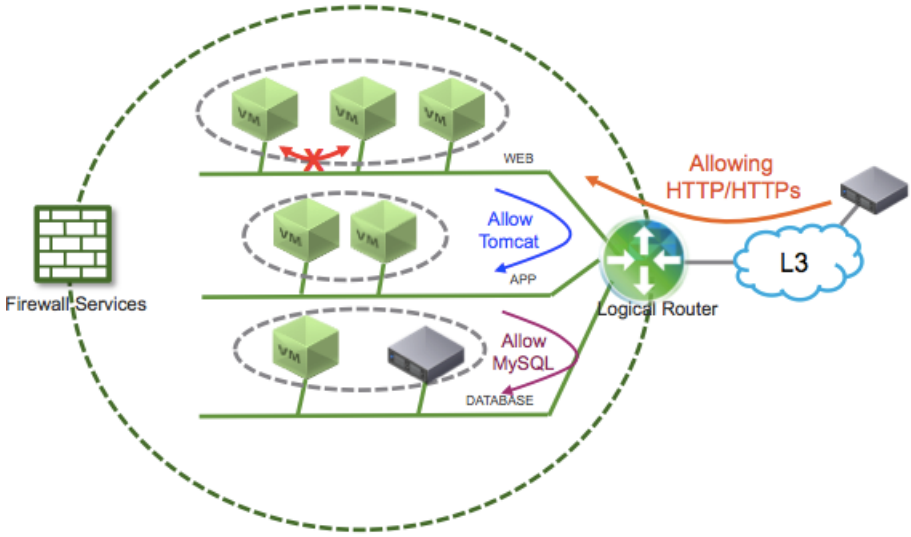


Figure 50 - Controlling Communication in a Multi-Tier Application

The VMware NSX platform includes two firewall functionalities: a centralized firewall service offered by the NSX Edge Services Gateway and a distributed kernel enabled as a VIB package (similarly to the distributed routing function previously described) on all the ESXI hosts part of a given NSX domain. The Distributed firewall (DFW), discussed in the first chapter, provides firewalling with near line rate performance, virtualization and identity aware with activity monitoring, among other network security features native to network virtualization.

Network Isolation

Isolation is the foundation of most network security, whether for compliance, containment or simply keeping development, test and production environments from interacting. While manually configured and maintained routing, ACLs and/or firewall rules on physical devices have traditionally been used to establish and enforce isolation and multi-tenancy are inherent to network virtualization.

Virtual networks (leveraging VXLAN technology) are isolated from any other virtual network and from the underlying physical network by default, delivering the security principle of least privilege. Virtual networks are created in isolation and remain isolated unless specifically connected together. No physical subnets, no VLANs, no ACLs, no firewall rules are required to enable this isolation.

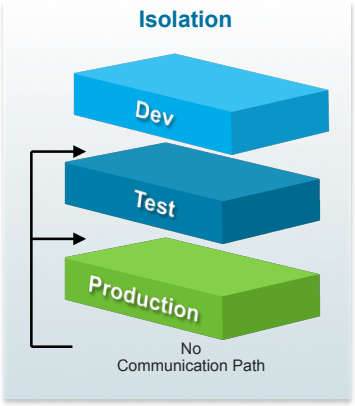


Figure 51 – Network Isolation

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. Case in point 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, all 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 then the workloads connected to the virtual networks. For example, 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. Again, independent from any VLANs, ACLs, or firewall rules that would traditionally be required to create this isolation.

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. For example, segmenting traffic between a Web tier, Application tier and Database tier. Traditional processes for defining and configuring segmentation 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, application ports and protocols.

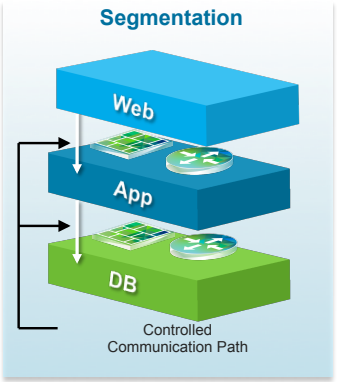


Figure 52 – Network Segmentation

Network segmentation, like isolation, is a core capability of VMware NSX network virtualization. A virtual network can support a multi-tier network environment, meaning 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).

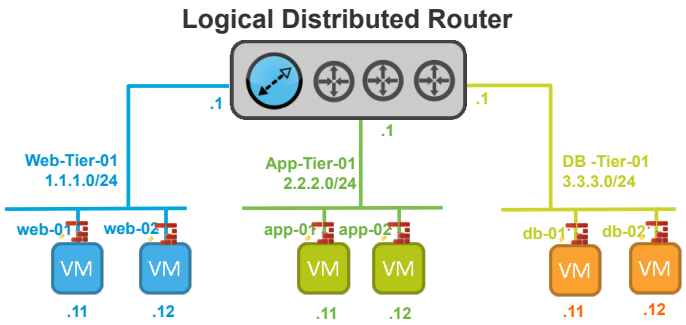


Figure 53 - Micro-segmentation with NSX DFW – Multi-Tier Network Design

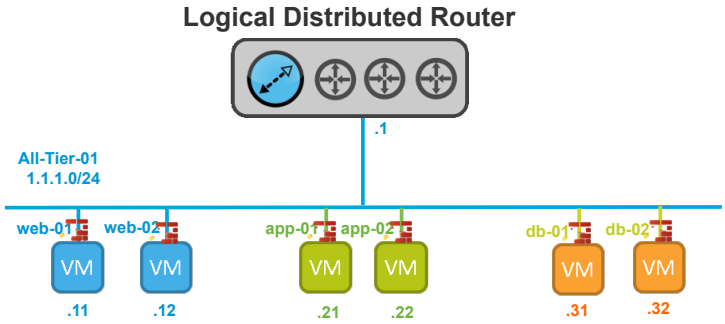


Figure 54 - Micro-segmentation with NSX DFW – Single-Tier Network Design

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, however, has been shaken, as more and more attacks, breaches and downtime are attributed to human error in too antiquated, manual network security provisioning and change management processes.

In a virtual network, network services (L2, L3, Firewall, etc.) 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.

Taking Advantage of Abstraction

Traditionally, network security required the security team to have a deep understanding of network addressing, application ports, protocols, all bound to 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 these physical constraints and apply network security based on user, application and business context.

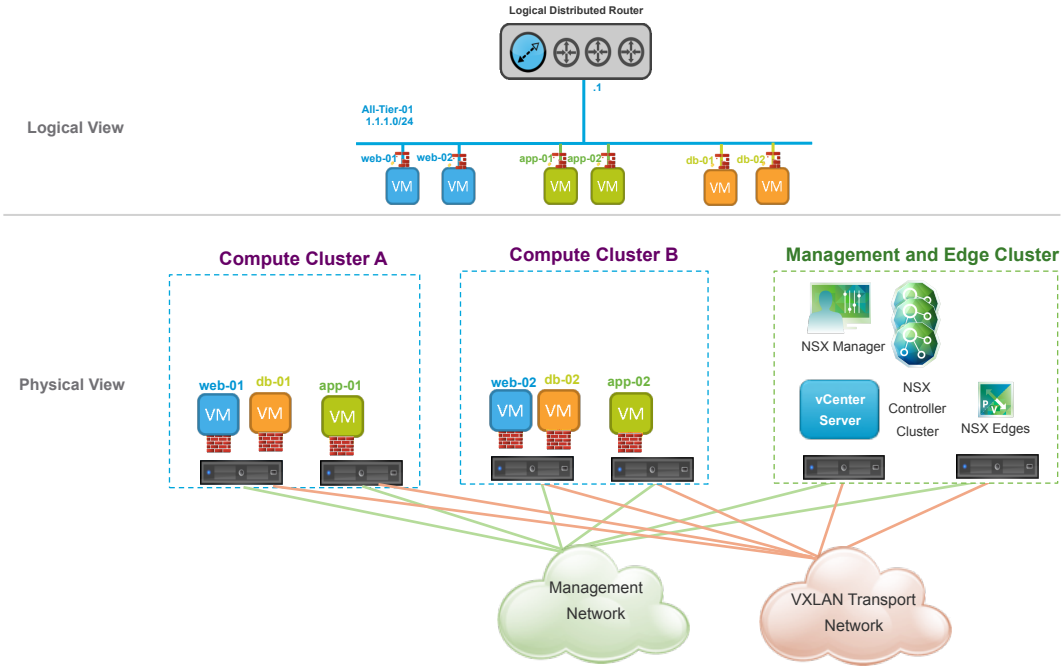


Figure 55 – Security Policy is No More tight to Physical Network Topology!

Advanced Security Service Insertion, Chaining and Steering

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 consumed in the logical pipeline.

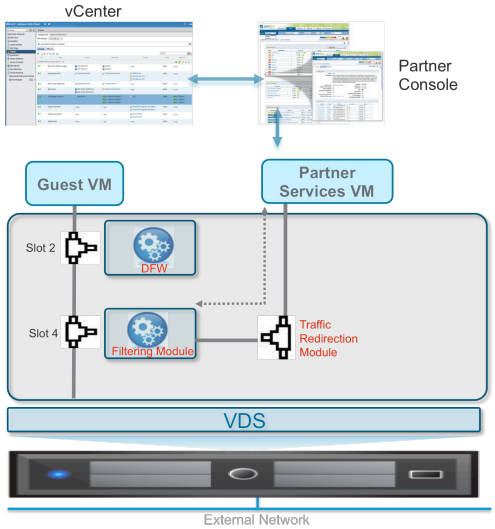


Figure 56 – Service Insertion, Chaining and Steering

Between guest VM and logical network (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 in the above Figure 56, slot 2 is allocated to DFW, slot 4 to the Palo Alto Networks VM-series FW. Another set of slots is available to plug more third-party services.

Traffic exiting the guest VM always follows the path with increasing slot-ID number (i.e. packet is redirected to slot 2 and then slot 4). Traffic reaching the guest VM follows the path in the reverse slot-ID order (slot 4 and then slot 2).

Every security team uses a unique combination of network security products to meet the needs of their environment. The VMware NSX platform is being leveraged by VMware's entire ecosystem of security solution providers. Network security teams are often challenged to coordinate network security services from multiple vendors in relationship to each other. Another powerful benefit of the NSX approach is its ability to build policies that leverage NSX service insertion, chaining and steering to drive service execution in the logical services pipeline, based on the result of other services, making it possible to coordinate otherwise completely unrelated network security services from multiple vendors.

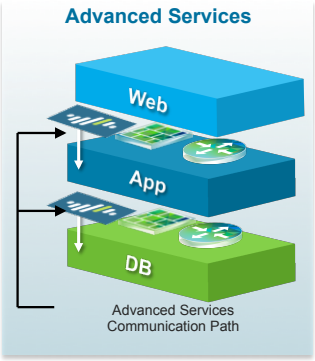


Figure 57 – Network Segmentation with Advanced Services provided by third party vendor.

For example, our integration with Palo Alto Networks will leverage the VMware NSX platform to distribute the Palo Alto Networks VM-Series next generation firewall, making the advanced features locally available on each hypervisor. Network security policies, defined for applications workloads provisioned or moved to that hypervisor, are inserted into the virtual network's logical pipeline. At runtime, the service insertion leverages the locally available Palo Alto Networks next-generation firewall feature set to deliver and enforce application, user, context-based control policies at the workloads virtual interface.

Consistent Visibility and Security Model Across both Physical and Virtual Infrastructures

VMware NSX allows automated provisioning and context-sharing across virtual and physical security platforms. Combined with traffic steering and policy enforcement at the virtual interface, partner services, traditionally deployed in a physical network environment, are 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.

1. **Existing tools and processes.** Dramatically increase provisioning speed, operational efficiency and service quality while maintaining separation of duties between server, network and security teams.
2. **Control closer to the application, without downside.** Traditionally, this level of network security would 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.
3. **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.

Micro-segmentation with NSX DFW Use Case and Implementation

Let's now discuss how DFW can be leveraged to implement micro-segmentation for a 3-tier application (Web / App / DB) where multiple organizations (Finance and HR) share the same logical network topology.

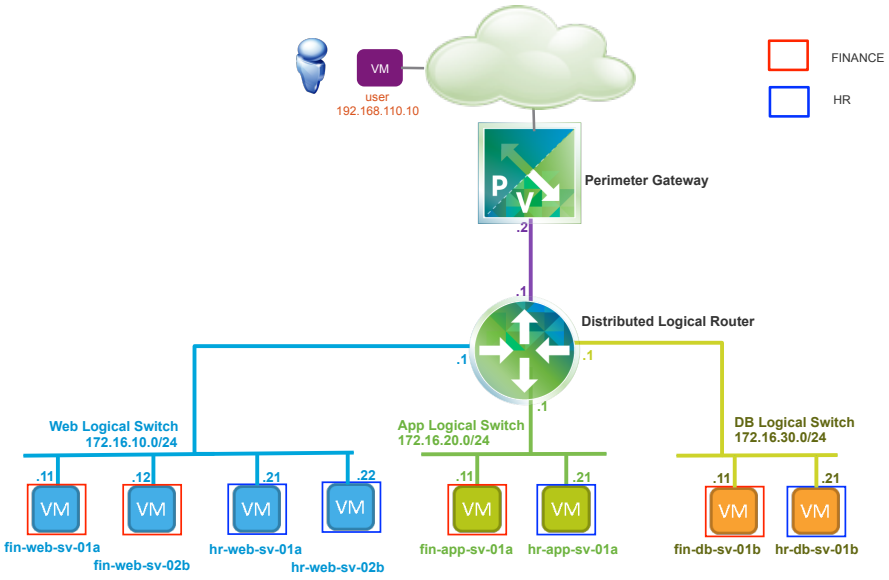


Figure 58 – Micro-Segmentation Use Case

In this network design, we have 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 Switch.

DLR is connected to an Edge Services Gateway that provides connectivity to physical world. An end user (connected to the external network) can reach the Web Logical Switch subnet (dynamic routing protocol turned on DLR and ESG to announce Web LS subnet for instance).

In this example, the Finance and HR organization each have 2 Web servers, 1 App server and 1 DB server.

Particularity in this micro-segmentation design is that servers of same role are connected to same Logical Switch, irrespective of the organization they belong to. HR and Finance are used as examples and 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 we would have if Finance and HR workloads were connected to different logical networks. Remember that the power of DFW resides in the fact that the network topology is no more a barrier to security enforcement: any type of topology is supported and for any of them, the same level of traffic access control can be achieved!

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

An interesting property within Service Composer is the Security Group (SG) construct. SG allows to dynamically or statically including objects into a container. And this container will be used as source or destination field of our DFW security policy rule.

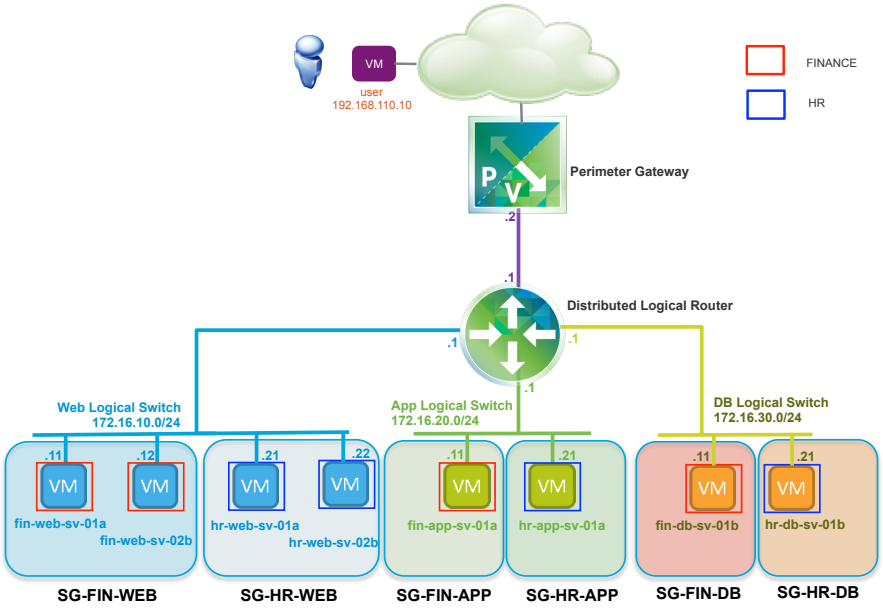


Figure 59 – Grouping VM into Service Composer/Security Groups

As shown in the above diagram, Web servers from the Finance organization are grouped into a Security Group called SG-FIN-WEB. To build this SG, we can use for instance dynamic inclusion based on VM name or Security Tag: VM name containing 'fin-web' syntax will be automatically included in SG-FIN-WEB – or VM assigned with Security Tag 'FIN-WEB-TAG' will be automatically added to SG-FIN-WEB. In case user wants to use static inclusion, he can manually pick fin-web-sv-01a and fin-web-sv-02b into SG-FIN-WEB.

In this design, we have 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 implement micro-segmentation. Once done, it is now easy to implement traffic policy as shown below:

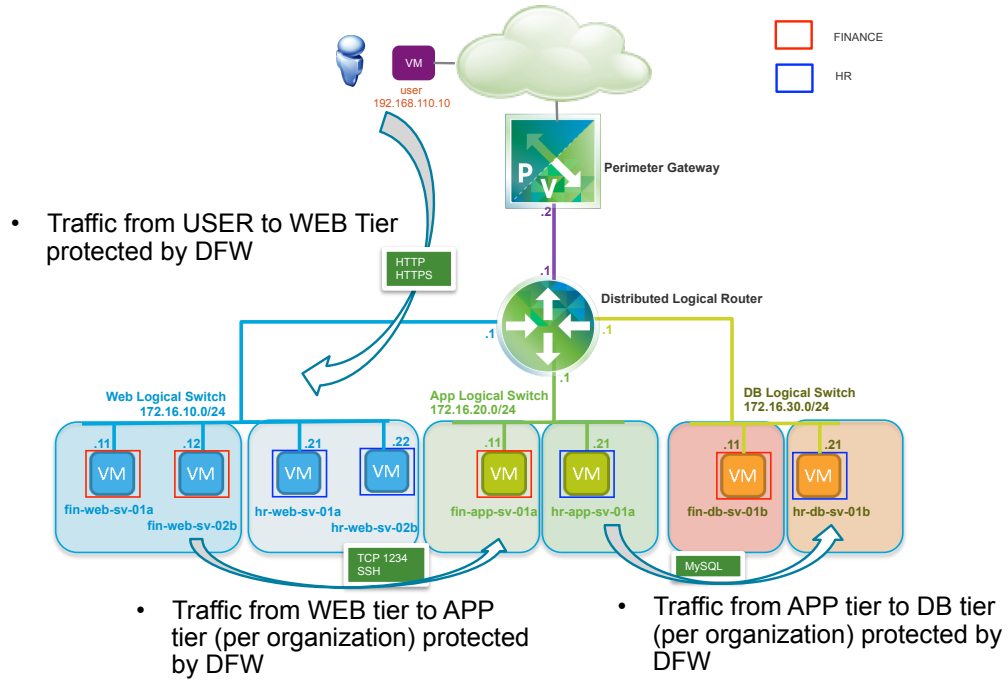


Figure 60 – Inter-Tier Network Traffic Security Policy

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

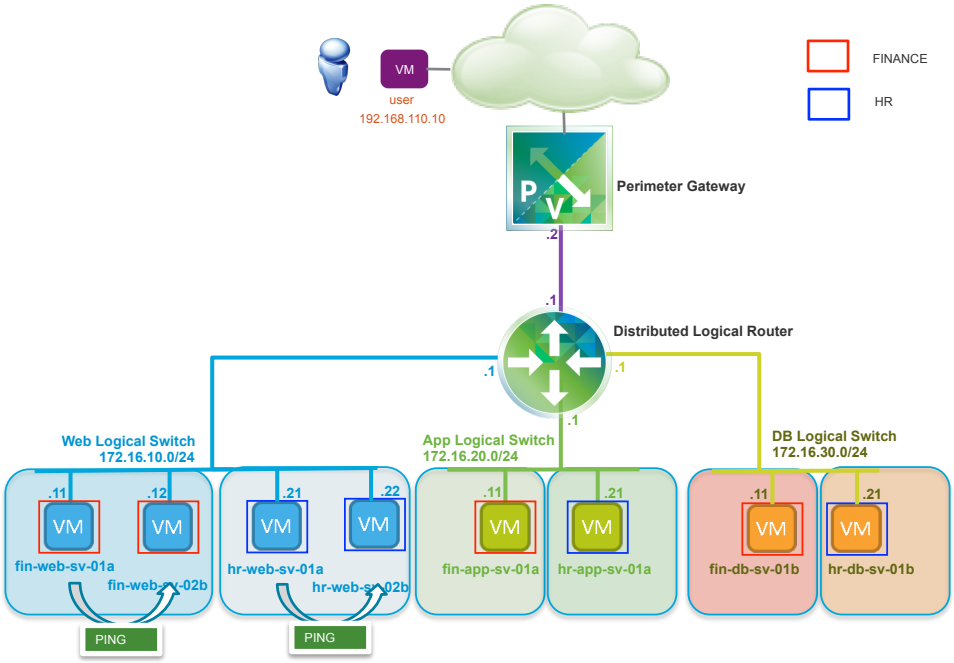
- From INTERNET to Web servers, allow HTTP and HTTPS. Other traffic is discarded.
- From Web tier to App tier, allow TCP 1234 and SSH (services used for simple example). Other traffic is discarded.
- From App tier to DB tier, allow MySLQ. Other traffic is discarded.

An important point to remember is 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 as following:

- Servers of same organization can ping each other when belonging to the same tier.

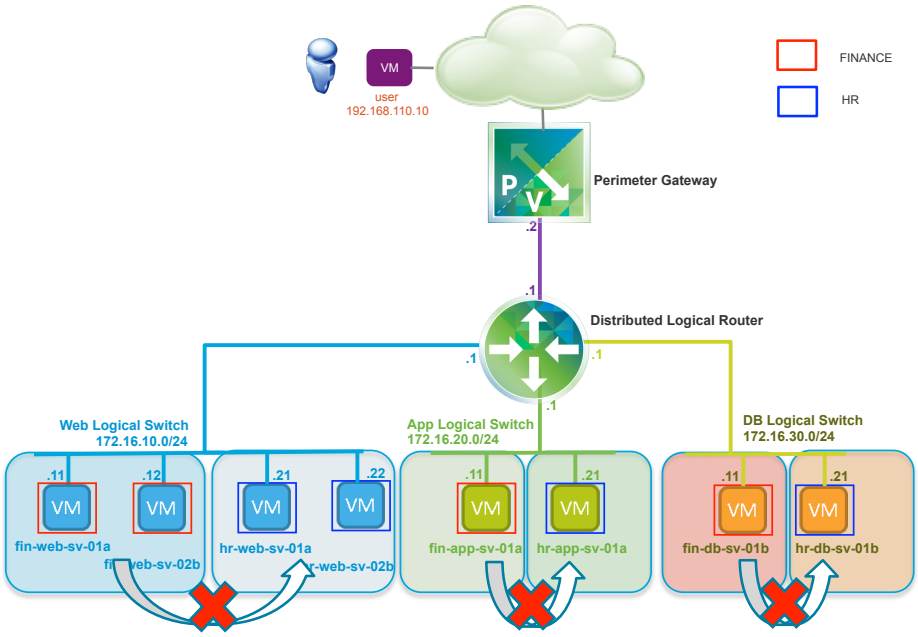
As an example, fin-web-sv-01a can ping fin-web-sv-02b. However, fin-web-sv-01a cannot ping hr-web-sv-01a or hr-web-sv-02b.



- Intra-Tier traffic permitted for PING within each organization

Figure 61 – Intra-Tier Network Traffic Security Policy (same organization)

Also, communication between workloads from different organizations within the same tier should simply be forbidden as shown in Figure 62 below.



- Intra-Tier traffic blocked across organization

Figure 62 – Intra-Tier Network Traffic Security Policy (across organization)

The first thing to do for implementing this micro-segmentation use case is to modify the DFW default policy rule from Action = Allow to Action = Block. This is because we are going to use a positive security model, where only the allowed traffic flows are defined in the security rules table and everything other communication is denied.

Name	Source	Destination	Service	Action
Default rule	ANY	ANY	Any	Block

Figure 63 - DFW Default Policy Rule

Note: 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 VC 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. Notice that the same considerations hold true for other management products (vCAC, vCOPS, etc.), as they will get 1 instance of DFW per vNIC (only NSX Manager and NSX Controllers are automatically excluded from DFW policy enforcement).

The security policy rules governing inter-tiers traffic can thus be defined as shown in Figure 64.

Name	Source	Destination	Service	Action
FIN – INTERNET to WEB Tier	ANY	SG-FIN-WEB	HTTP HTTPS	Allow
HR– INTERNET to WEB Tier	ANY	SG-HR-WEB	HTTP HTTPS	Allow
FIN – WEB Tier to APP Tier	SG-FIN-WEB	SG-FIN-APP	TCP-1234 SSH	Allow
HR– WEB Tier to APP Tier	SG-HR-WEB	SG-HR-APP	TCP-1234 SSH	Allow
FIN – APP Tier to DB Tier	SG-FIN-APP	SG-FIN-DB	MYSQL	Allow
HR – APP Tier to DB Tier	SG-HR-APP	SG-HR-DB	MYSQL	Allow

Figure 64 - DFW Inter-Tiers Policy Rules

Finally, for intra-tier communication, we are going to implement the security policy rules shown in Figure 65.

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

Figure 65 - DFW Intra-Tier Policy Rules

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), as well as improving its high-availability characteristics.

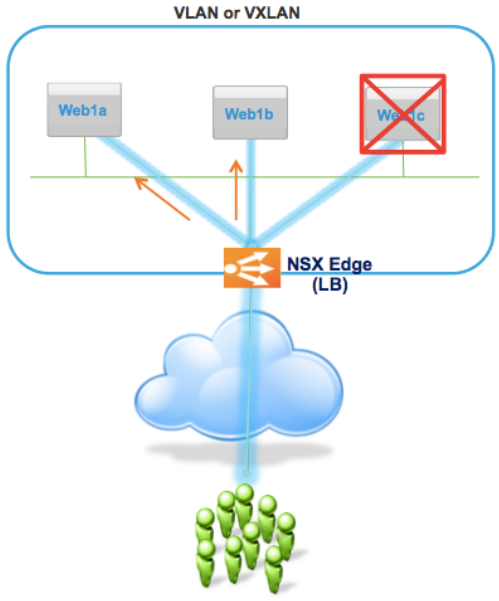


Figure 66 - NSX Load Balancing

The NSX load balancing service is specially designed for cloud with the following characteristics:

- Fully programmable via API
- Same single central point of management/monitoring as other NSX network services

The load balancing services natively offered by the NSX Edge satisfies the needs of the majority of the application deployments. This is because the NSX Edge provides a large set of functionalities:

- Support any TCP applications**,** including, but not limited to, LDAP, FTP, HTTP, HTTPS
- Support UDP application starting from NSX SW release 6.1.
- Multiple load balancing distribution algorithms available: round-robin, least connections, source IP hash, URI
- Multiple health checks**:** TCP, HTTP, HTTPS including content inspection
- Persistence**:** Source IP, MSRDP, cookie, ssl session-id
- Connection throttling**:** max connections and connections/sec
- L7 manipulation**,** including, but not limited to, URL block, URL rewrite, content rewrite
- Optimization through support of SSL offload

Note: the NSX platform can also integrate load-balancing services offered by 3rd party vendors. This integration is out of the scope for this paper.

In terms of deployment, the NSX Edge offers support for two types of models:

- One-arm mode (called proxy mode): this scenario is highlighted in Figure 67 and consists in deploying an NSX Edge directly connected to the logical network it provides load-balancing services for.

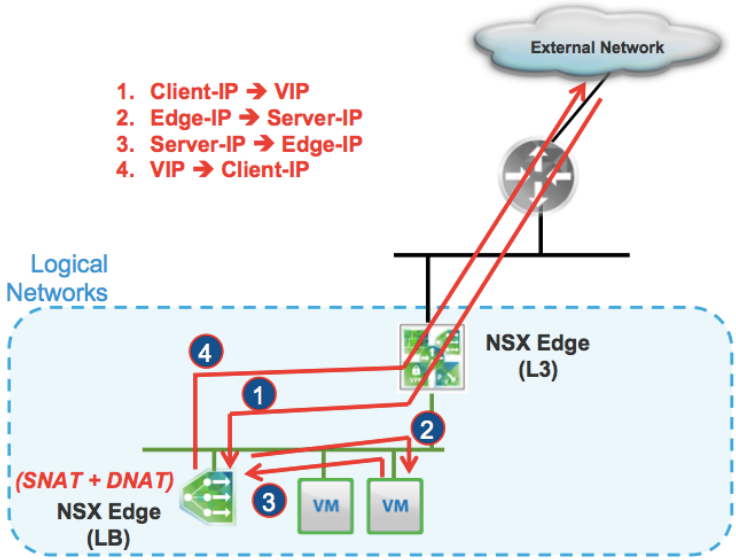


Figure 67 - One-Arm Mode Load Balancing Services

The one-armed load balancer functionality is shown above:

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 LB the return traffic from the server farm to the client.
3. The server in the server farm replies by sending the traffic to the LB (because of the S-NAT function previously discussed).
4. The LB performs again a Source and Destination NAT service to send traffic to the external client leveraging its VIP as source IP address.

The advantage of this model is that it is simpler to deploy and flexible as it allows deploying LB services (NSX Edge appliances) directly on the logical segments where they are needed 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 DC to have visibility into the original client IP address.

Note: the LB 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 but it is obviously limited to HTTP traffic.

- Inline mode (called transparent mode) requires instead deploying the NSX Edge inline to the traffic destined to the server farm. The way this works is shown in Figure 68.

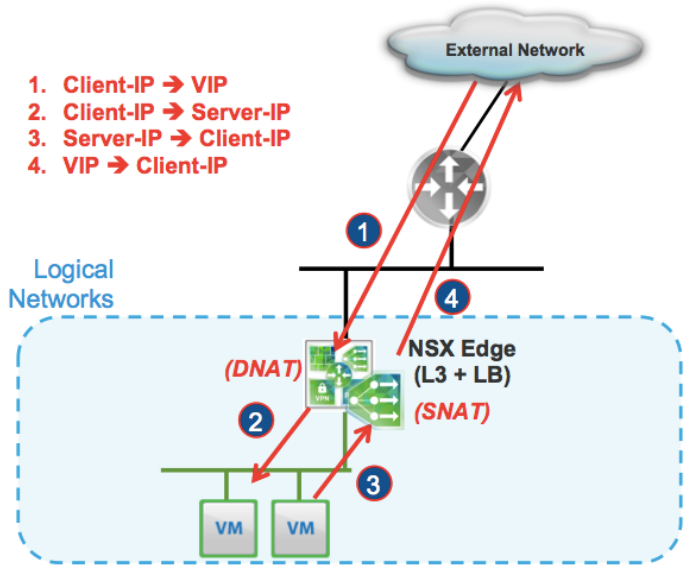


Figure 68 - Two-Arms Mode Load Balancing Services

1. The external client sends traffic to the Virtual IP address (VIP) exposed by the load balancer.
2. The load balancer (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 and the traffic is received again by the LB since it is deployed inline (and usually as the default gateway for the server farm).
4. The LB 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 allows the servers to have full visibility into the original client IP address. At the same time, it is less flexible from a design perspective as it usually forces using the LB as default gateway for the logical segments where the server farms are deployed and this implies that only centralized (and not distributed) routing must be adopted for those segments. It is also important to notice that in this case LB is another logical service added to the NSX Edge already providing routing services between the logical and the physical networks. As a consequence, it is recommended to increase the form factor of the NSX Edge to X-Large before enabling load-balancing services.

In terms of scalability and throughput figures, the NSX load balancing services offered by each single NSX Edge can scale up to (best case scenario):

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

In Figure 69 are some deployment examples of tenants with different applications and different load balancing needs. Notice how each of these applications is hosted on the same Cloud with the network services offered by NSX.

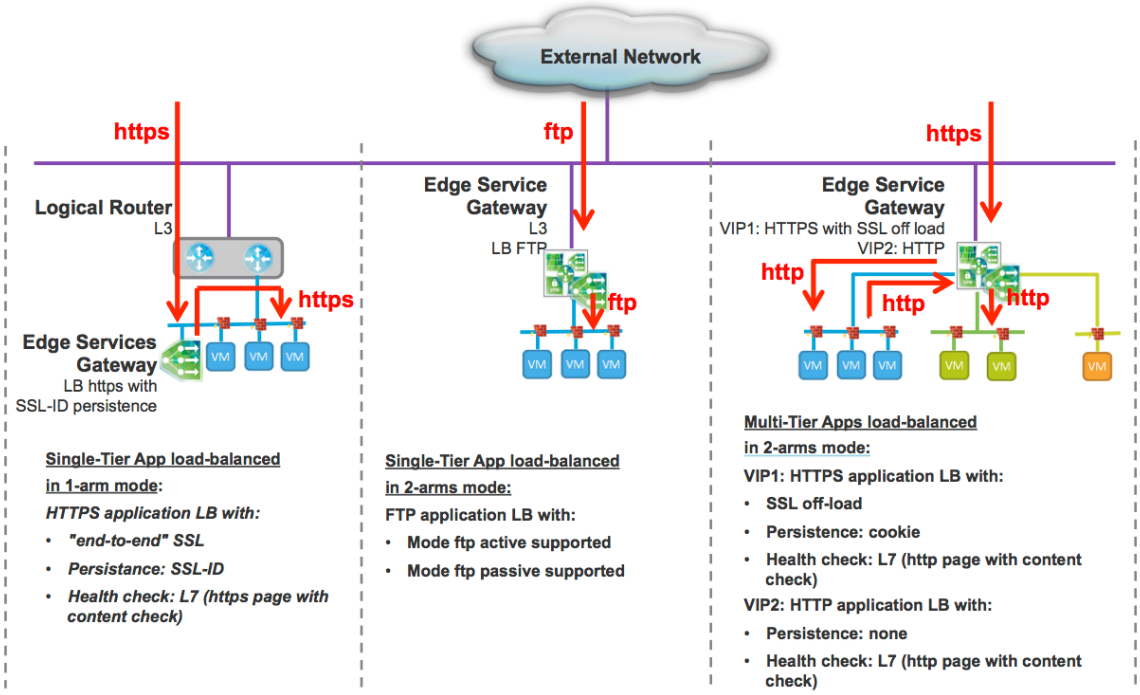


Figure 69 - Deployment Examples of NSX Load Balancing

Two final important points to highlight:

- The load balancing service can be fully distributed across tenants. This brings multiple benefits:
  - Each tenant has its own load balancer.
  - Each tenant configuration change does not impact other tenants.
  - Load increase on one tenant load-balancer does not impact other tenants load-balancers scale.
  - Each tenant load balancing service can scale up to the limits mentioned above.
- Other network services are still fully available
  - The same tenant can mix its load balancing service with other network services such as routing, firewalling, VPN.

Virtual Private Network (VPN) Services

The last two NSX logical network functionalities discussed in this paper are Virtual Private Network (VPN) services. Those are divided in two categories: L2 VPN and L3 VPN.

L2 VPN

The deployment of an L2 VPN service allows extending L2 connectivity across two separate data center locations.

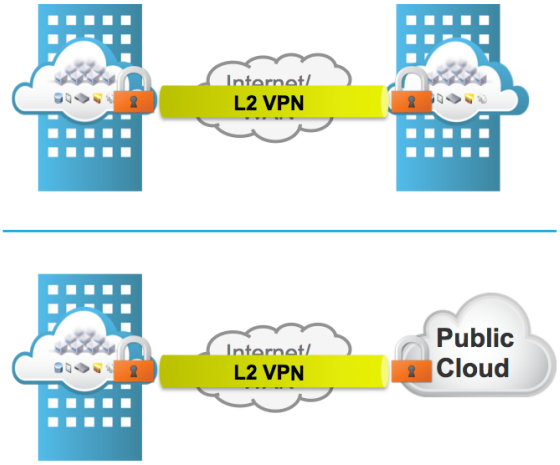


Figure 70 - Use Cases for NSX L2 VPN

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

- Enterprise Workload migration/DC Consolidation
- Service Provider Tenant On-boarding
- Cloud Bursting (Hybrid Cloud)
- Stretched Application Tiers (Hybrid Cloud)

Figure 71 highlights the creation of a L2 VPN tunnel between a pair of NSX Edge devices deployed in separate DC sites.

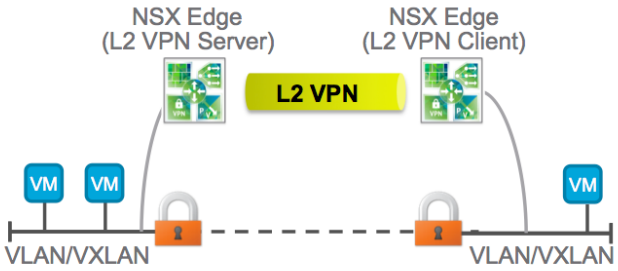


Figure 71 – NSX Edges for L2 VPN Services

Some considerations for this specific deployment are listed below:

- The L2 VPN connection is an SSL tunnel connecting separate networks in each location. The connected separate networks offer connectivity to the same address space (IP subnet), which is the characteristic that makes this a L2 VPN service.
- The local networks can be of any nature, VLAN or VXLAN and the L2 VPN service can also interconnect networks of different nature (VLAN on one site, VXLAN on the other site).
- Currently this is only a point-to-point service that can be established between two locations. The NSX Edge deployed in one DC site takes the role of the L2 VPN server, whereas the NSX Edge in 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. Independently from who owns and manages this network, no specific requirements are put on it in terms of latency and bandwidth, nor in terms of MTU. The NSX L2 VPN solution is built with much robustness to work pretty much across any available network connection.

The NSX 6.1 Software release brings many improvements to the L2 VPN solution. Some of the most relevant ones are:

- With 6.0 releases, it is required to deploy two independent NSX Domains in the two sites that need to be connected.

This implies the deployment of separate vCenter, NSX Manager and NSX Controller clusters in each location, and this may become an issue especially in service provider deployments (as for example for Hybrid Cloud use cases). 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 NSX at the remote site, basically allowing extending the solution to vSphere-only customers.

- NSX 6.1 release also introduces a third type of interface on the NSX Edge (in addition to the Uplink and Internal ones), named Trunk. Leveraging Trunks it is possible to extend L2 connectivity between multiple networks (VLAN or VXLAN backed port-groups) deployed on each site (in 6.0 the networks extended were 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 hence be deployed in each site.

L3 VPN

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

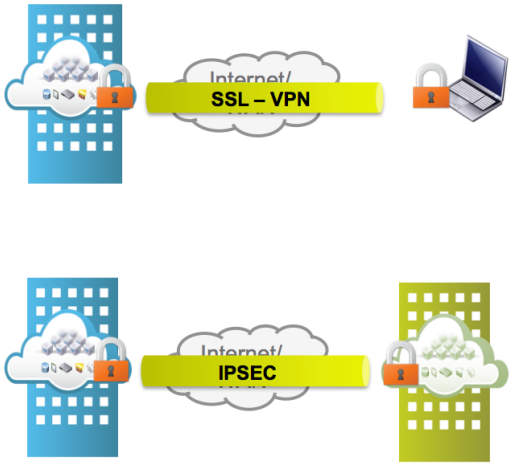


Figure 72 - NSX L3 VPN Services

As highlighted in Figure 72, 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 data center. This service is usually referred to as SSL VPN-Plus.

Alternatively, the NSX Edge can be deployed to use standardized IPSec protocol settings to interoperate with all major physical VPN vendors' equipment and establish site-so-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. It is also worth noticing that in the current implementation, unicast only communication is supported across the IPSec connection, and as consequence dynamic routing protocols cannot be used and static routing configuration is required.

NSX-v Design Considerations

In this section, we discuss how logical networks implemented by NSX-v using VXLAN overlay technology can be deployed over common data center network topologies. We first address requirements for the physical network and then look at the design considerations for deploying NSX-v network virtualization solution.

Evolution of Data Center Network Designs

Before discussing what attributes a physical network should have in order to provide a robust IP transport for deploying network virtualization, it may be worth it to quickly describe the current trends and design evolutions of DC networks. Until few years ago, most of the DC networks were exclusively built in a modular fashion, as shown in **Figure 73**.

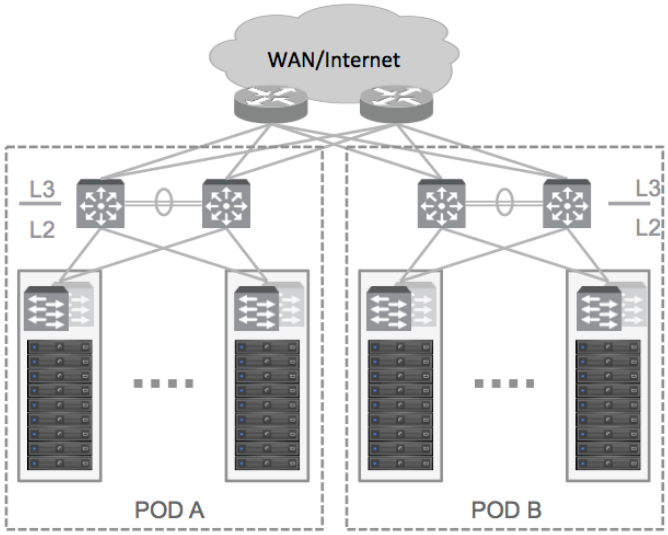


Figure 73 - Classical Access/Aggregation/Core DC Network

The modular design shown above represents a scalable approach, where multiple PODs can be interconnected via an L3 DC Core. Each POD represents an aggregation block that limits the extension of VLANs inside the DC fabric. The aggregation layer devices of each POD are in fact the demarcation line between L2 and L3 network domains and this is done to limit the extension of L2 broadcast and failure domains. The immediate consequence of this approach is that all the applications requiring L2 adjacency (Application clusters, Virtual Machines live migration, etc.) has to be deployed inside a given POD. Also, this classical design is mostly effective if the majority of traffic flows are exchanged in the north-south direction and only limited communication happens inter-PODs.

The classical modular approach just described has evolved in the last few years into a leaf-spine design (**Figure 74**).

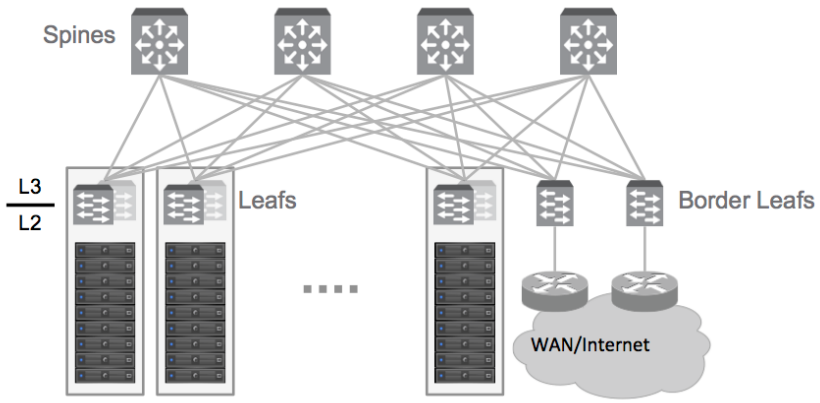


Figure 74 - Leaf-Spine Fabric Design

The network shown above typically takes the name “folded Clos” design, and is usually deployed by customers who require a DC fabric providing high scalability characteristics with a reduced number of functionalities. Every device in the Clos fabric takes the name of leaf or spine, depending on the tier where it is deployed:

Leaf Layer: the Leaf layer is what determines oversubscription ratios, and thus may influence the required size of the Spine. Devices in this layer usually employ simpler designs that have fewer “moving parts” to effectively forward packets while learning the network graph accurately.

Spine Layer: the Spine layer is responsible for interconnecting all Leafs. Individual nodes within a Spine are usually not connected to one another nor do they form any routing protocol adjacencies among themselves. Rather, Spine devices typically are characterized by a simple and minimal configuration, as they are responsible for performing fast traffic switching and essentially function as the fabric “backplane”.

Border Leaf Layer: in some scenarios the Spine devices may be used to directly connect to the outside world, but more often they forward externally destined traffic to a pair of specialized leaf nodes acting as Border Leafs. Border Leafs may inject into the fabric default routes to attract traffic intended for external destinations.

This design evolution is driven by two main requirements: the ever increasing amount of east-west communication and the desire to be able to deploy applications across the DC fabric without being restricted by the traditional limited extension of L2 domain inside each POD.

The first requirement is addressed by increasing the number of spine devices each leaf connects to: as shown in figure above, each leaf is offered multiple equal cost paths in order to augment the bandwidth available for east-west communication and make predictable and deterministic the latency experienced for those flows.

The second requirement can be tackled by creating a larger L2 Domain that extends across the entire DC Fabric. Different switch vendors have proposed alternatives solutions to build those large L2 Fabric networks by eliminating the use of Spanning-Tree as control plane protocol, like for example Cisco’s proprietary FabricPath technology (or the similar Transparent Interconnection of Lots of Links (TRILL) that is currently an IETF Draft) or Juniper’s QFabric solution.

However, more and more customers are steering toward the deployment of leaf-spine Routed Fabric networks, where the boundary between L2 and L3 is brought down to the leaf (or Top-of-Rack) device layer (see previous **Figure 74**).

The routed fabric design offers the same characteristics of low oversubscription, high east-west bandwidth availability, and predictability and configuration uniformity. In addition, it enhances device interoperability (since switches from multiple vendors can be interconnected together) and overall resiliency. The main challenge of a routed design is the incapability of supporting applications that require L2 adjacency between different nodes, given that the extension of an L2 domain is limited behind each leaf switch. This is where NSX and Network Virtualization come to the rescue: the decoupling between logical and physical networks provides any type of connectivity in the logical space, independently from how the underlay network is configured.

Since the expectation is that the deployment of routed fabric networks will become the preferred option for customers, the focus on the rest of this document is on data center routed access designs where the leaf/access nodes perform full layer 3 functionalities.

However, it is also expected that customers will have to slowly migrate from the original multi-pod deployment toward pure leaf/spine architecture. NSX can ease this migration process thanks to the decoupling of virtual networks and

physical network that allows it to seamlessly create a common pool of IT resources and consistent networking service model across multiple disparate types of networks (Figure 75).

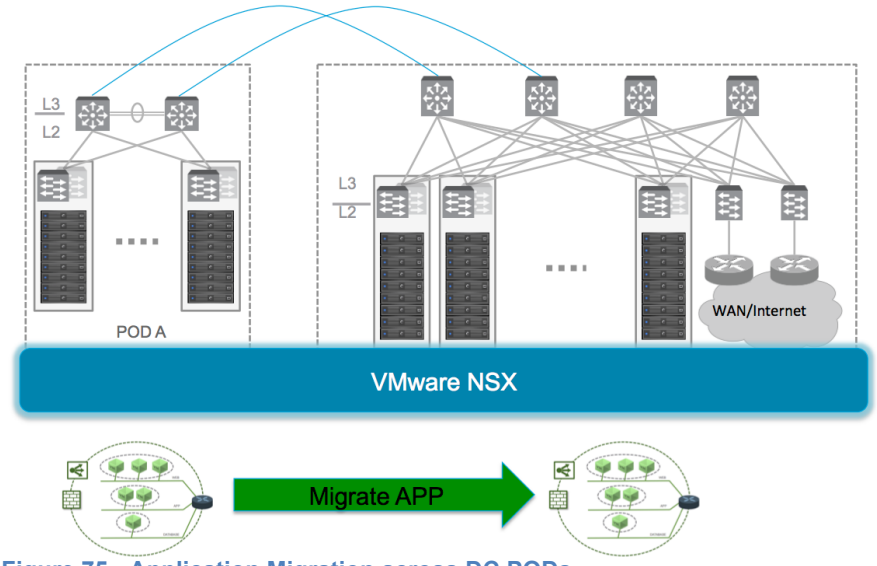


Figure 75 - Application Migration across DC PODs

Physical Network Requirements

One of the key goals of NSX and its network virtualization functionalities is to provide virtual-to-physical network abstraction and decoupling. However, in order to have a successful deployment it is important that the physical fabric provides a robust IP transport with the following attributes:

- Simplicity
- Scalability
- High-bandwidth
- Fault-tolerance
- Quality of Service (QoS)

The following sections offer some detail for each of these parameters.

Note: in the following discussion, the terms *access layer switch*, *top-of-rack (ToR) switch* and *leaf switch* are used interchangeably. A leaf switch (or better, a redundant 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.

Simplicity

Configuration of the switches that compose the overall fabric inside a data center must be simple. General or global configuration such as AAA, SNMP, SYSLOG, NTP, and so on, should be replicated almost line-by-line, independent of the position of the switches.

Leaf Switches

Ports facing the servers inside a rack should have a minimal configuration. The use of 801.Q trunks for carrying a small number of VLANs is becoming more popular with the adoption of 10GE interfaces on ESXi hosts; for example, VXLAN traffic, management, storage and VMware vSphere vMotion traffic. The leaf switch terminates and provides default gateway functionality, respectively, for each VLAN; that is, it has a switch virtual interface (SVI) for each VLAN.

A pair of leaf switches is typically deployed inside a rack to provide a redundant connection point for the servers.

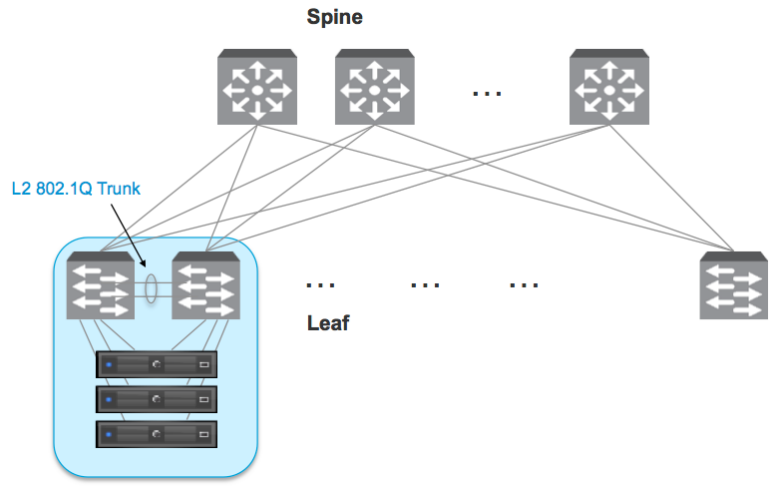


Figure 76 - Deploying a Redundant Pair of Leaf Nodes

As shown in Figure 76, a direct connection is required between this pair of leaf devices, usually configured as an 802.1Q L2 trunk carrying the different VLANs used by the servers in the rack. The need for this L2 trunk is independent from the specific server's uplinks configuration (discussed in the "VDS Uplinks Connectivity in an NSX-v Domain" section).

Uplinks from the leaf switch to the aggregation or spine layer are routed point-to-point links, as shown in Figure 77.

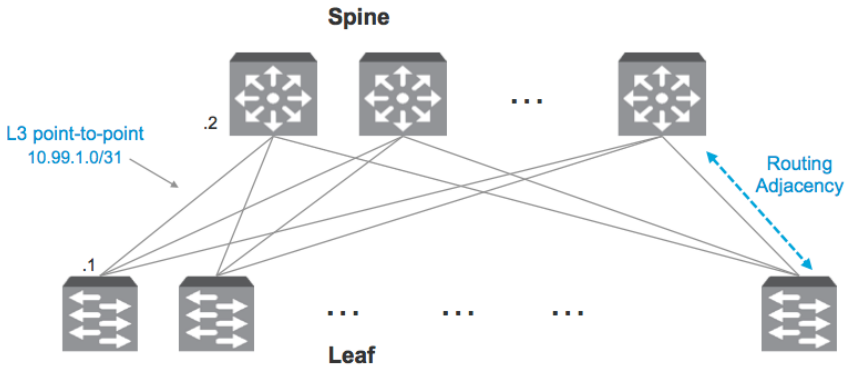


Figure 77 - L3 Connectivity between Leaf and Spine Switches

VLAN trunking on the link between leaf and spine is not allowed—not even for a single VLAN. A dynamic routing protocol—OSPF or eBGP, for example—is configured between the leaf and spine switches. Each ToR switch in the rack will advertise a small set of prefixes, typically one per VLAN or subnet it has present. In turn, it will calculate equal cost paths to the prefixes received from other ToR switches.

Note: when leveraging encapsulation technologies it is important to increase the MTU supported on the ESXi hosts and on all the interfaces of the devices deployed in the physical network. VXLAN adds 50 Bytes to each original frame; hence the typical recommendation is to increase the MTU size at least to a 1600 Bytes value. The MTU on the ESXi server (both for the internal logical switch and for the VDS uplinks) is automatically tuned (1600 Bytes by default) when provisioning VXLAN to the hosts itself from the NSX Manager UI.

Spine Switches

The spine switch has only interfaces that connect to leaf switches; all interfaces are configured as routed point-to-point links, effectively acting as the "other end" of the leaf switch's point-to-point uplinks.

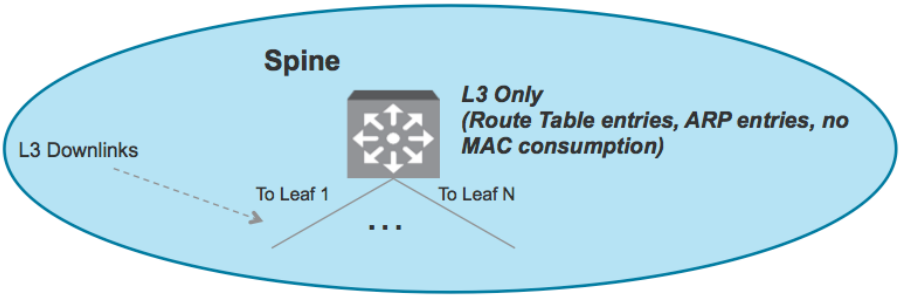


Figure 78 - Spine Switch Interfaces

Links between spine switches are not required. If there is a link failure between a spine switch and a leaf switch, the routing protocol will ensure that no traffic for the affected rack is attracted to the spine switch that has lost connectivity to that rack.

Scalability

Factors concerning scale include the number of racks supported in a fabric, the amount of bandwidth existing between any two racks in a data center, the number of paths a leaf switch can select from when communicating with another rack, and so on.

The total number of ports available across all the spine switches usually dictates the number of racks supported in a fabric and the oversubscription that is acceptable. More details are provided in the “High Bandwidth” section.

Different racks might be hosting different types of infrastructure. As it will be discussed in more detail in the “NSX-v Deployment Considerations” section, it is possible to identify three types of racks functionalities: compute racks, edge racks and infrastructure racks.

Keeping the design modular ensures flexibility in the network connectivity provided to the various racks. This is important because different type of racks might have different bandwidth requirements (storage filers for example drive more traffic in and out the infrastructure racks). Link speed as well as number of links can hence be varied to satisfy different bandwidth demands. This can be done for each rack without sacrificing any of the architecture of the spine switch or leaf switch.

The number of links between a given leaf node and the spine switches dictates the numbers of physical paths that traffic originated from this rack can take. Because the number of hops between any two racks is consistent, equal-cost multipathing (ECMP) can be utilized. Assuming traffic sourced by the servers carries a TCP or UDP header, traffic spray can occur on a per-flow basis. This also apply to VXLAN traffic since, as previously mentioned in the “VXLAN Primer”, the UDP Source Port value is calculated providing the hashing of the L2/L3/L4 headers eventually present in the original L2 frames getting encapsulated. This means that even if a couple of ESXi hypervisors were exchanging VXLAN traffic sourced and received from a single VTEP IP address, the traffic could be load-balanced across the fabric based on the original flows exchanged between workloads connected to the logical networks.

High Bandwidth

In spine–leaf switch topologies, oversubscription typically occurs—if at all—at one point: the leaf switch. The calculation is simple: total amount of bandwidth available to all servers connected to a given leaf switch divided by the aggregate amount of uplink bandwidth provides the oversubscription.

For example, 20 servers with one 10 Gigabit Ethernet (10GbE) port each create up to 200Gbps of bandwidth. Assuming eight 10GbE uplinks to the spine—a total of 80 Gbps— this results in a 2.5:1 oversubscription factor.

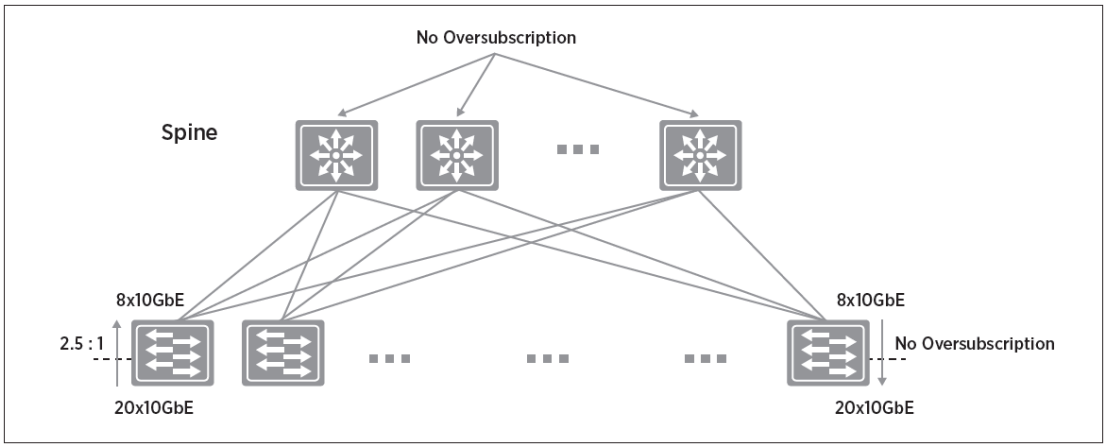


Figure 79 - Oversubscription Example for Leaf-Spine Topologies

As discussed in the previous section, depending on the rack's function, more or less bandwidth can be made available to a rack by virtue of provisioning more or fewer uplinks. In other words, the level of oversubscription can vary on a per-rack basis.

From an architecture standpoint, one rule must be obeyed: the number of uplinks from a leaf switch to each spine switch must be the same. This means that having two uplinks to spine switch A and only one uplink to spine switches B, C and D would be a poor design because "more" traffic would be sent to the leaf switch via spine switch A, potentially creating a hot spot.

Fault Tolerance

The larger the environment, the more switches that make up the overall fabric, the greater the possibility for one component of the data center switching fabric to fail. The reason for building a resilient fabric is that it can sustain individual link or box failures without a widespread impact.

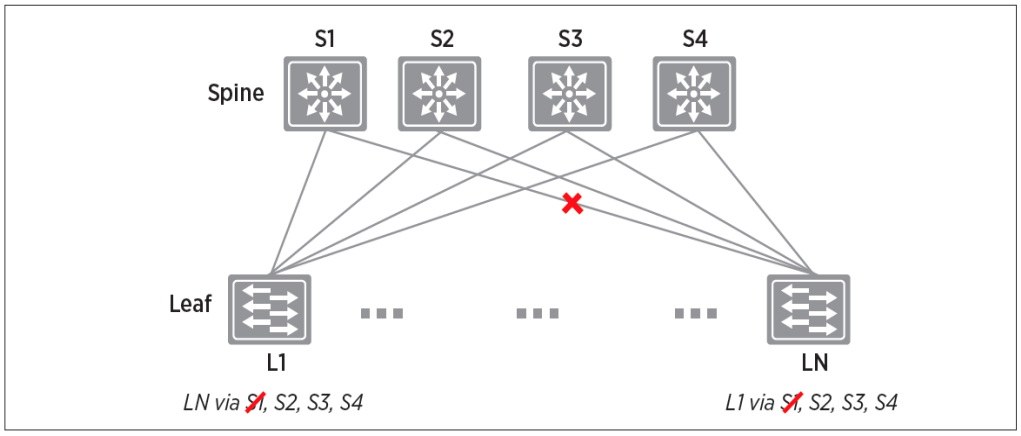


Figure 80 - Link Failure Scenario in Leaf-Spine Topologies

For example, if one link or one of the spine switches were to fail, traffic between racks would continue to be routed in an L3 fabric across the remaining spine switches. For L3 fabrics, the routing protocol ensures that only remaining paths would be chosen. And because more than two spine switches can be installed, the impact of a spine switch failure is reduced.

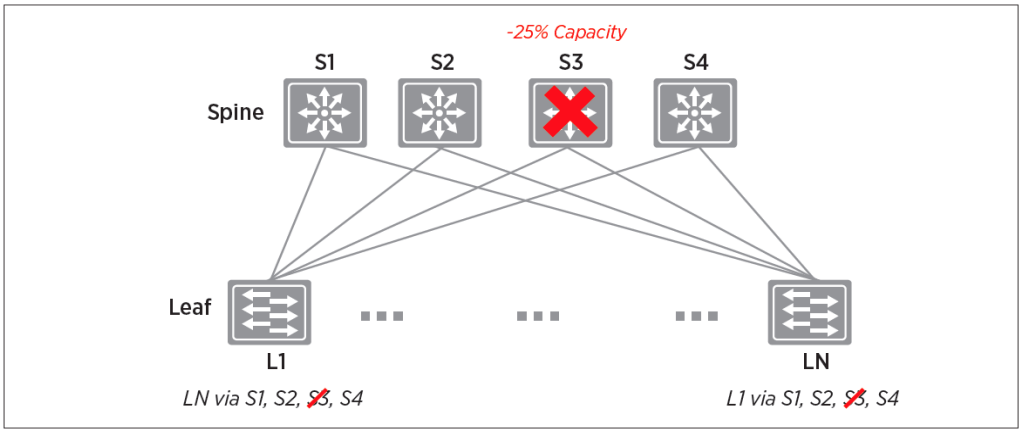


Figure 81 - Spine Switch Failure Scenario and Impact on Bandwidth

Multipathing-capable fabrics handle box or link failures, reducing the need for manual network maintenance and operations. If a software upgrade must be performed on a fabric switch, the node can be taken out of service gracefully by changing routing protocol metrics; quickly, the traffic through that switch will be drained, freeing it up for maintenance. Depending on the width of the spine—that is, how many switches are in the aggregation or spine layer—the additional load the remaining switches must carry is not as significant as if there were only two switches in the aggregation layer.

Differentiated Services – Quality of Service

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.

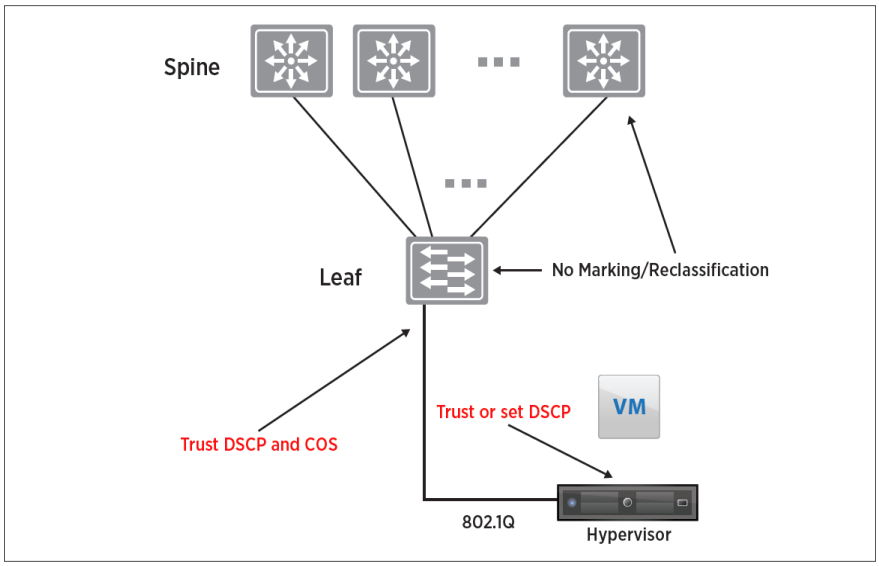


Figure 82 - Quality of Service (QoS) Tagging

For virtualized environments, the hypervisor presents the trusted boundary, meaning it sets the respective QoS values for the different traffic types. In this case, 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 looked at 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 L3 or IP layer. The L2 QoS is sometimes referred to as “Class of Service” (CoS) and the L3 QoS as “DSCP marking”.

NSX-v allows trusting the DSCP marking originally applied by a virtual machine or to explicitly modify and set the DSCP value at the logical switch level. In both cases, 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.

NSX-v Deployment Considerations

This section discusses how network virtualization offered by VMware NSX-v can be placed on top of the scalable L3 network fabric presented in the previous sections. 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.

Being able to provide L2 connectivity at the logical network level, independently from the characteristics of the underlying network infrastructure is the fundamental property that enables the decoupling effect. In the example of a L3 DC fabric where the L2/L3 boundary is positioned at the leaf layer, VLANs cannot span beyond a single rack inside the switching infrastructure but this does not prevent the provisioning of L2 adjacency between workloads connected to the logical networks.

When building a new environment, it is essential to choose an architecture that allows for future growth. The approach discussed here works for deployments that begin small but grow to large-scale ones while keeping the same overall architecture.

The guiding principle for such deployments is that the network virtualization solution does not imply any spanning of VLANs beyond a single rack. Although this appears to be a simple requirement, it has widespread impact on how a physical switching infrastructure can be built and on how it scales.

In an NSX enabled data center 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 DC architect:

- Flexibility of expanding and contracting resources for specific functions.
- Ability to isolate and develop span of control.
- Managing life-cycle of certain resources for specific functions (Better HW - CPU, memory and NIC, upgrade, migration etc.).
- High availability based on functional connectivity needs, DRS, FT, Edge P/V and HA etc
- Automation control over areas or functions that require frequent changes (app-tier, security tags, policies, load-balancer etc.)

Figure 83 highlights the aggregation of ESXi hosts in different types of racks: compute racks, infrastructure racks and edge racks.

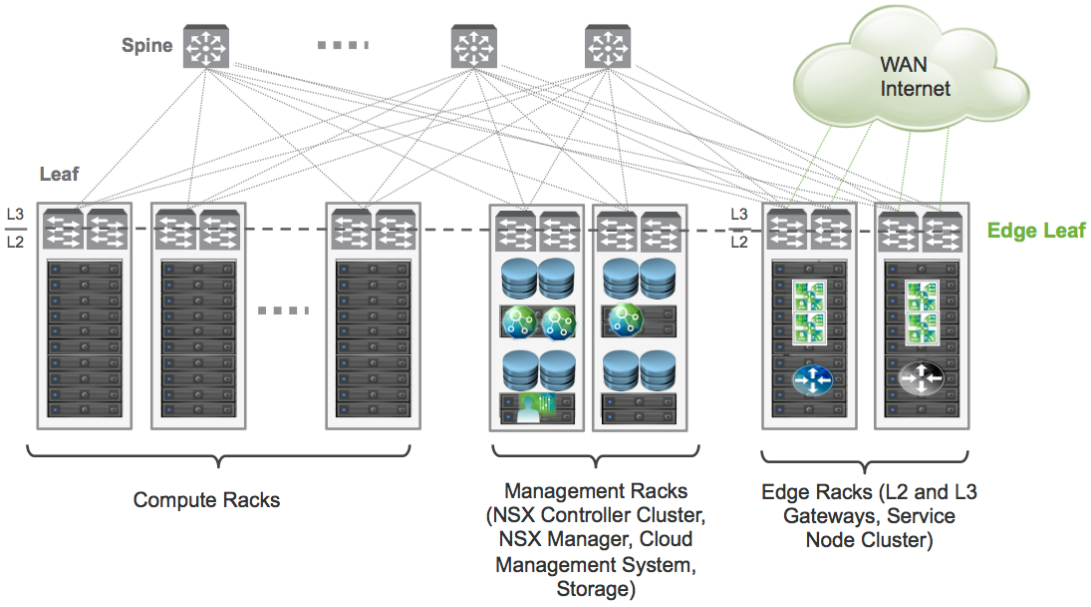


Figure 83 – Compute, Edge and Management Racks

Note: the separation of functions for each type of racks can be logical and not necessarily physical, For example, as it will be discussed later in this chapter, it may be common in smaller deployments to consolidate edge and infrastructure functions together in the same physical racks.

At a very high level, those are the main characteristics of those different types of racks:

- **Compute Racks**: they make up the part of the infrastructure where tenant virtual machines are hosted. They should have the following design characteristics:
  - For new deployments or redesigns:
    - Should not require VLANs for virtual machines.
    - Should not require infrastructure VLANs (VXLAN, vMotion, storage, management) to extend beyond a compute rack .
  - Provide a repeatable-rack design that can be easily scaled out as needed.
- **Edge Racks**: provide tighter interaction with the physical infrastructure while bridging between the overlay world and the physical infrastructure. The following are the main functions provided by the edge racks:
  - Provide on-ramp and off-ramp connectivity to physical networks (north-south L3 routing on NSX-Edge virtual appliances).
  - Allow communication with physical devices connected to VLANs in the physical networks (NSX L2 bridging).
  - Host centralized logical or physical services (firewall, load-balancers, etc.).
- **Management Racks**: they host the management components, including vCenter Server, NSX Manager, NSX Controller, Cloud Management Systems (CMS) and other shared IP storage–related components. It is key that this portion of the infrastructure does not have any tenant-specific addressing. If bandwidth-intense infrastructure services are placed in these racks—IP-based storage, for example—the bandwidth of these racks can be dynamically scaled, as previously discussed in the “High Bandwidth” section.

ESXi Cluster Design in an NSX-v Domain

The cluster design covers the following key attributes:

- vSphere Cluster scale and scope for compute, edge and management racks.
- VDS uplinks connectivity considerations for compute, edge and management racks

- VDS design in an NSX-v domain

Compute, Edge and Management Clusters

The NSX reference design recommends grouping ESXi hosts used for computing in separate clusters striped across dedicated racks. The edge and management clusters can instead be combined in a single rack or dedicated racks depending on the scale of the design. The cluster design along with the service based rack design is shown in Figure 84.

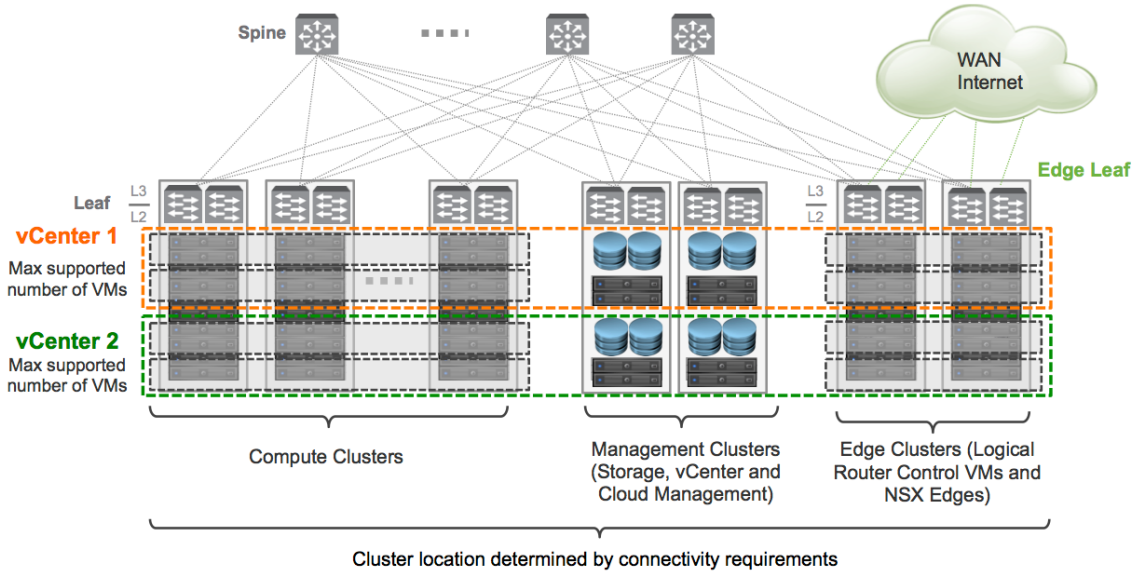


Figure 84 - Cluster Design

The compute, management and edge clusters are laid out based on the purpose they serve. Clusters can be sized up to the vSphere platform limits and span across racks, so that a switch or rack failure only impacts a percentage of the cluster capacity.

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

- The vCenter and NSX Manager have a one-to-one mapping relationship. In other words there is only one NSX Manager (NSX domain) per given vCenter. This implies that it is the scale limit of vCenter that governs the scale of the overall NSX deployment.
- The NSX controllers, as part of the installation process, must be deployed into the same vCenter server that NSX Manager is connected to.
- The difference in small/medium vs. large-scale designs usually consists in the number of vCenters deployed and how they map to NSX Manager.

In a small/medium data center deployments:

- A single vCenter is usually deployed for managing all the NSX components, as shown in Figure 85.

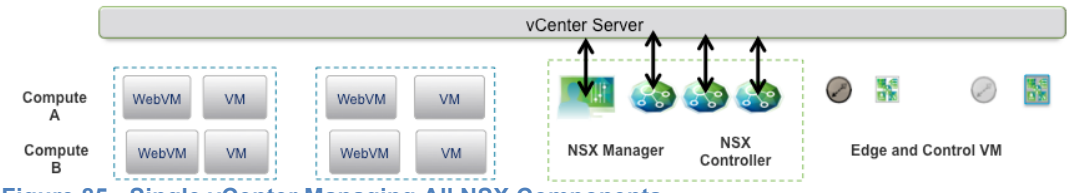


Figure 85 - Single vCenter Managing All NSX Components

- The recommendation is still to dedicate separate clusters and set of racks for compute resources.
- It is also recommended to deploy separate edge and management clusters to accommodate future growth. Differently from what shown in Figure 83 and Figure 84, the edge and management racks are usually consolidated

and the corresponding clusters of ESXi hosts share Top-of-Rack connectivity,

Most enterprise deployments make use of a dedicated vCenter for the management cluster. This vCenter is usually already deployed even before the NSX platform is introduced in the architecture. When that happens, one or more dedicated vCenter servers part of the management cluster are usually introduced to manage the resources of the NSX domain (edge and compute clusters), as shown in Figure 86 below.

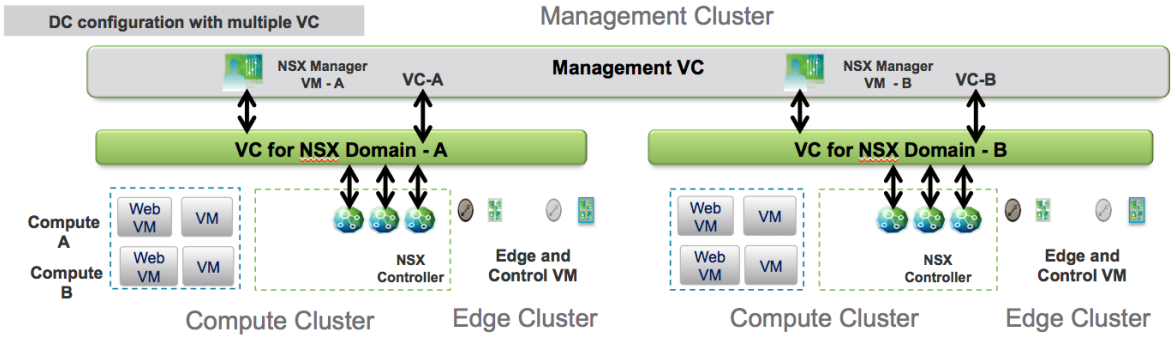


Figure 86 - Dedicate vCenter Servers for Managing NSX Resources

There are several advantages in adopting such approach:

- Avoids circular dependencies – the management cluster should always be outside of the domain it manages.
- Mobility of management cluster for remote DC operation.
- Integration with existing vCenter.
- Ability to deploy more than one NSX-domain.
- Upgrade of main vCenter does not affect the NSX domains.
- SRM and other explicit state management are possible.

Additionally, a large-scale design employs one or more dedicated racks (management racks) and ToR switches to host the management cluster.

Some additional considerations around ESXi host cluster deployment in an NSX domain:

- Compute capacity can scale in two ways, both horizontally by adding more compute racks and vertically by adding vCenter Servers.
- The ESXi hosts part of the management cluster do not normally require to be provisioned for VXLAN. If they are deployed across two racks (to survive a rack failure scenario) they usually require extending L2 connectivity (VLANs) across those racks for management workloads such as vCenter Server, NSX Controllers, NSX Manager and IP Storage. Figure 87 highlights the recommended (even if not the only) way to provide L2 connectivity across the management racks.

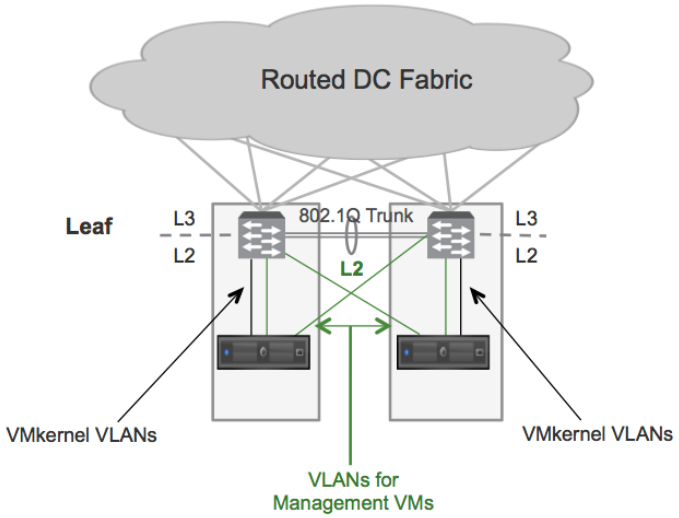


Figure 87 - L2 Connectivity Requirements across Management Racks

The pair of leaf switches is deployed across the two management racks (one physical leaf for rack) and dedicated cross-racks L2 cables are used to interconnect the ToR devices (L2 802.1q trunk connection). Additionally, each server is dual-homed to both ToR switches, leveraging a physical uplink local to the rack where the server is deployed and a cross-rack physical link to reach the leaf in the second rack (green connections in the picture above). Assuming each server is leveraging two physical uplinks for all types of traffic, the local uplink could carry both the VMkernel VLANs (used for infrastructure traffic like VXLAN, vMotion, storage) and the VLANs used for Mgmt VMs, whereas only the latter would be transported on the physical uplink connecting to the leaf in the second rack. Alternatively, different physical uplinks may be dedicated to those types of traffic.

- The edge racks are usually the only racks connected to the external physical network (in addition of being connected to the common data center fabric). This allows the external VLANs to be contained to a small set of racks, rather than being stretched across the entire environment. The NSX Edge then provides connectivity between external networks and compute workloads.
- 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. As it will be discussed more in detail as part of the "Edge Racks Design" session, in order to stripe edge clusters across separate racks it may be needed to extend L2 connectivity between those. While this can be achieved the same way shown in Figure 87, it is more common to leverage a separate L2 network infrastructure.

VDS Uplinks Connectivity in an NSX-v Domain

This section covers the generic design options for connecting ESXi hosts to the ToR switches present in each type of rack (VDS uplinks connectivity). The design criteria used for connecting hosts are as follows:

- 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-v deployments.
- Type of isolation required based on traffic SLA – dedicated uplinks (for example for vMotion/Management) vs. shared uplinks.
- Type of cluster – compute workloads, edge and management with or without storage etc.
- Amount of bandwidth required for VXLAN traffic that may determine the decision of deploying a single or multiple VTEPs.

VDS leverages a special port-group called DVUplink for uplink connectivity. The below Figure 88 shows the teaming options supported for the specific port-group dynamically created when VXLAN 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 2x10G
Route based on Originating Port	✓	✓	Both Active
Route based on Source MAC hash	✓	✓	Both Active
LACP	✓	×	Flow Based - Both Active
Route based on IP Hash (Static EtherChannel)	✓	×	Flow Based - Both Active
Explicit Failover Order	✓	×	Only One Active
Route based on Physical NIC Load (LBT)	×	×	×

Figure 88 – Teaming Options for Uplink Connectivity Options in NSX

As shown above, 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 would have an impact on the way VXLAN encapsulated traffic is sent and received on the VDS uplinks, as well as on the number of VMkernel (VTEP) interfaces that are created.

Depending on the chosen teaming mode for the VXLAN port-group, it may still be possible to leverage LBT for other port-groups (i.e. different types of traffic). Below are some additional considerations on how the selected uplinks connectivity impacts the flexibility of creating port-groups and how it is inherited across the cluster or across the transport zone.

- VDS offers great flexibility for what concerns uplinks connectivity. Every port-group defined as part of a VDS could, in principle, 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 belonging to separate clusters). Still referring to 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. Trying to choose 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 reason is that the ether-channel teaming choice implies a requirement of configuring a port-channel also on the physical network side. 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 (and often in loss of communication).
- The VTEP design impacts the uplink selection choice, since NSX supports single or multiple VTEPs configurations.
    - The single VTEP option offers first of all operational simplicity; if the requirement is to support less than 10G of VXLAN traffic on each given host, then the Explicit Failover Order option is a valid one, as it allows to physically separate 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 independently from any failure. Dedicating a pair of physical uplinks to VXLAN traffic and keeping them in active/standby states can in fact guarantee that if a physical link or switch failed, applications would still have access to the same 10G pipe of bandwidth. Obviously, this comes at the price of deploying more physical uplinks and only actively using half of the available bandwidth.

      If the requirement is to support more than 10G of VXLAN traffic, then Static EtherChannel or LACP can be deployed, marrying the increase of bandwidth with the simplicity of a single VTEP deployment. As mentioned, this comes at the price of additional configuration and coordination required on the physical switches, and often mandates specific functionalities (vPC/MLAG) to be supported on the Top-of-Rack devices. Also, in this case it must be considered the impact that physical failures would have on the applications.

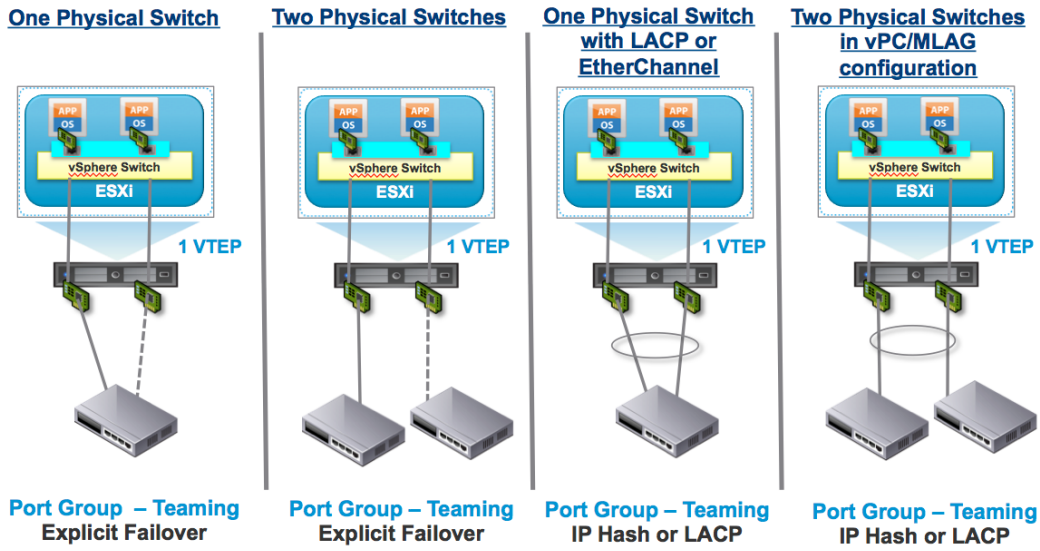


Figure 89 - Single VTEP Uplink Options

Figure 89 shows the Single VTEP options. Notice how a single VTEP is provisioned also 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 as a single logical uplink, since traffic sourced from the VTEP can be hashed on a per-flow basis across both physical paths.

- Multiple VTEPs may be configured to increase the bandwidth available for VXLAN traffic when use of port-channels is not possible (for example with the deployment of blade servers) or if the desire is to maintain the VDS uplink configuration decoupled and independent from the physical switch configuration. The increased operational complexity is the price to pay for this option and in certain cases may outweigh the benefits.

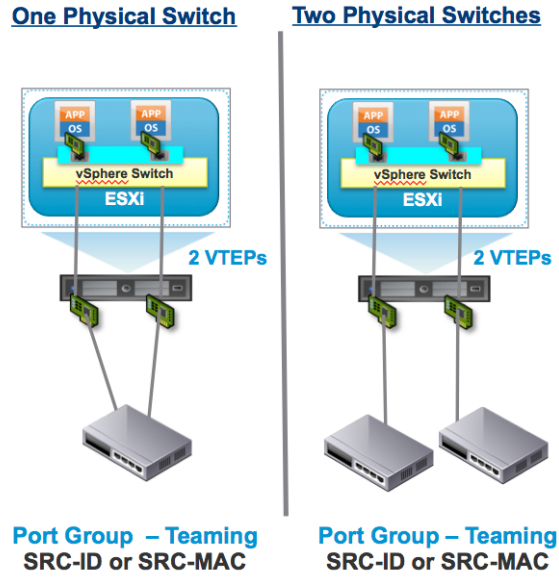


Figure 90 – Multi-VTEPs Uplink Option

Note: the number of provisioned VTEPs is always matching 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 92.

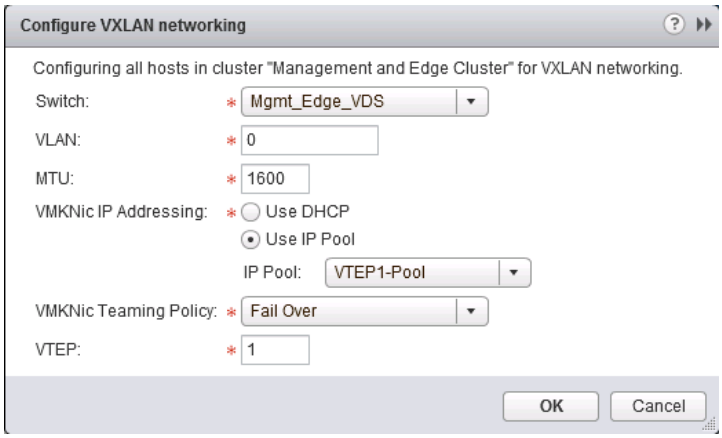


Figure 91 - Configure VXLAN Networking

The selection criteria for type of uplink configuration to deploy can be based on the following considerations:

- Simplicity of configuration – single VTEP vs. Physical switch configurations.
- BW Required for each type of traffic.
- Convergence requirement.
- Cluster specifics – compute, edge and management.
- The uplink utilization factors – flow based vs. MAC based.

The recommended teaming mode for VXLAN traffic for ESXi hosts in compute cluster is LACP. It provides sufficient utilization of both links and reduced failover time. It also offers simplicity of VTEP configuration and troubleshooting at the expense of extra configuration and co-ordination with the physical switch. Obviously, ToR diversity for ESXi attachment can only be achieved assuming the deployed physical switches support a type of multi-chassis ether-channel capability, like vPC (Cisco) or MLAG (Arista, Juniper, etc.).

For ESXi hosts part of the edge clusters is instead recommended avoiding the LACP or Static EtherChannel options. As previously mentioned, selecting LACP for VXLAN traffic implies that the same teaming option must be used for all the other port-groups (traffic types) part of the same VDS. One of the main functions of the edge racks is providing connectivity to the physical network infrastructure and 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.

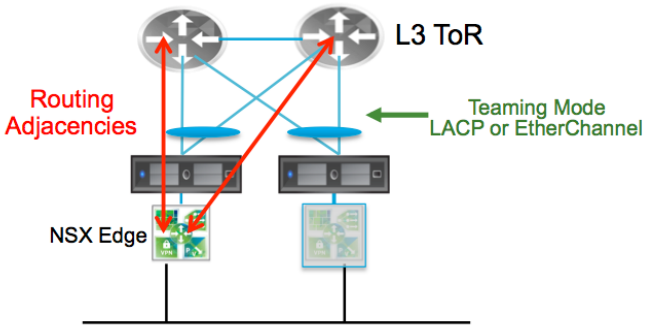


Figure 92 - Routing Peering Over Multi-Chassis EtherChannel

As shown in Figure 92, the ToR switches not only need to support a multi-chassis ether-channel functionality (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 physical network device of choice and may even be an unsupported configuration in many cases. As a consequence, the recommendation for edge clusters is to select either the Explicit Failover Order or the SRC-ID/SRC-MAC Hash as teaming options for VXLAN traffic, based on the evaluation of the pros and cons previously listed.

VDS Design in an NSX-v Domain

The vSphere Virtual Distributed Switch (VDS) is a foundational component in creating VXLAN logical segments. The span of VXLAN logical switches (layer 2 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 93 depicts the relationship between VDS, ESXi clusters and transport zone and the recommended design leveraging separate VDS switches for the compute and the edge clusters.

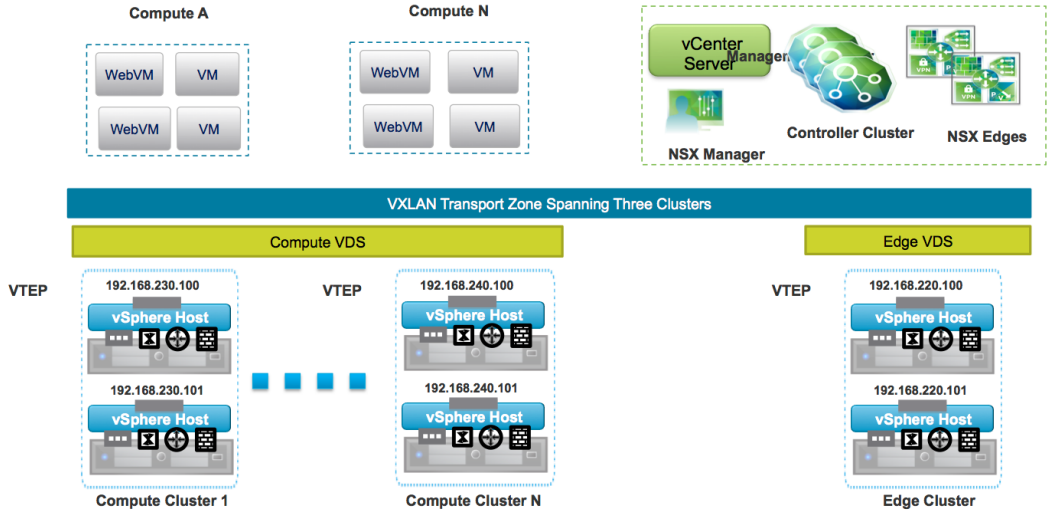


Figure 93 - Cluster, VDS and Transport Zone

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 of span of operational control: typically compute/virtual infrastructure admin and network admin are separate entities and thus each domain can 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.
- Flexibility in managing uplink connectivity on computes and edge clusters - see for example the above discussion on uplink design and the recommendation of using different teaming options for compute and edge clusters.
- Typically the VDS boundary is aligned with the transport zone and thus VMs connected to logical switches can span the transport zone. However, 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 (by mistake or by choice) to ESXi hosts dedicated to other services.
- Flexibility in managing VTEP configuration – see above discussion on VTEP design choices.
- Avoiding exposing VLAN-backed port-groups used by the services deployed in the edge racks (NSX L3 routing and NSX L2 bridging) to the compute racks.

Note: the VDS used for infrastructure services is not part of the VXLAN as typically the management cluster (part of the management racks) represents an independent entity providing functions that go beyond serving a given NSX domain (as previously discussed).

ESXi Host Traffic Types

ESXi hypervisors deployed in an NSX domain typically source different types of traffic. In the following sections, we look at 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. As previously mentioned, VXLAN traffic is usually sourced by ESXi hosts part of compute and edge clusters, but not part of the management clusters. The other types of traffic usually are 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 pair of uplinks is becoming more common.

In this scenario, 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 of those VLANs terminates at the leaf switch (Top-of-Rack device), so the leaf switch will provide a L3 interface for each VLAN. Such interfaces are also known as SVIs or RVIs (Figure 94).

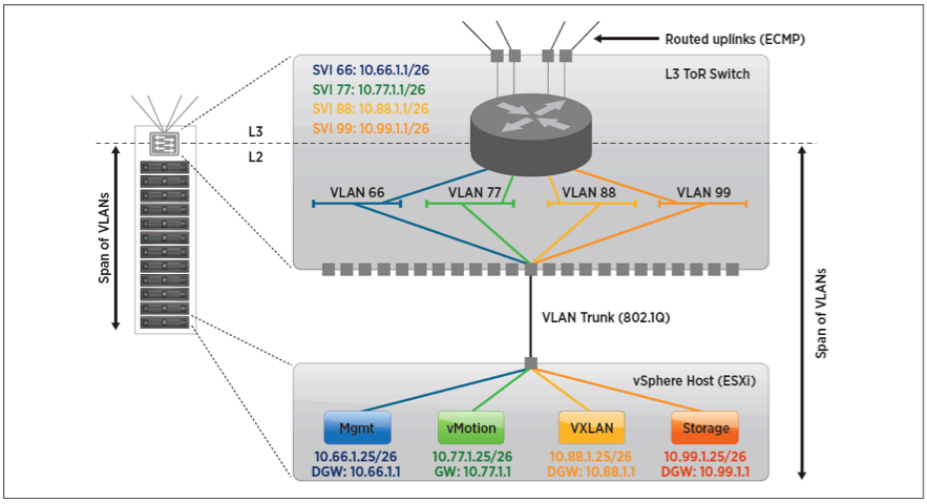


Figure 94 - Example of Host and Leaf Switch Configuration in a Rack

We will now look more closely at different types of host traffic.

VXLAN Traffic

After the vSphere hosts have been prepared for network virtualization using VXLAN, a new traffic type gets 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 to 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. Traffic that flows between virtual machines in the same data center is typically referred to as east–west traffic. For this type of traffic, both the source and destination VTEPs are situated in hypervisors located in compute and edge racks. Traffic leaving the data center will flow between a tenant virtual machine and an NSX edge, for example. This traffic is referred to as north–south traffic.

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 (the so called “VXLAN Transport Subnets”). Combining this with the recommendation of striping compute and edge clusters across separate racks brings an interesting deployment challenge. Provisioning VXLAN on ESXi hosts is in fact done 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 (see previous Figure 91).

The challenge consists of being able to assign to the VTEP interfaces IP addresses in different subnets. This can be achieved in two ways:

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

Management Traffic

Management traffic is sourced and terminated by the management VMkernel interface on the host and 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.

vSphere 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 clearly not possible when designing the network for network virtualization using L3 in the access layer, where it is mandated to select different subnets in different racks for those VMkernel interfaces. Until this design is fully and officially supported by VMware, it is recommended that users go through the RPQ process so VMware can validate the design on a case-by-case basis.

Storage Traffic

A VMkernel interface is used to provide features such as shared or non-directly attached storage. Typically, we refer to storage that can be attached via an IP connection—NAS or iSCSI, for example—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—that is, connected to a leaf switch is part of the same IP subnet. This subnet, however, cannot span beyond this leaf switch. Therefore, the storage VMkernel interface IP of a host in a different rack is in a different subnet.

Host Profiles for VMkernel Interface IP Addressing

In this section, we will discuss how to automate the provisioning of IP addresses to the VMkernel NICs of each traffic type, using the vSphere Host Profile method. 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 required sample configuration has been performed, a Host Profile can be created from that host and applied across the other hosts in the deployment. With this approach, users can quickly configure large numbers of hosts.

Before discussing how the Host Profile method can be used during configuration of whole racks of servers, we will look at the type of sample configuration required on a host in a rack. As previously mentioned, the same set of four VLANs—storage, vSphere vMotion, VXLAN, management—is usually provided in each rack (the VXLAN one limited to compute and edge racks). The VLAN IDs associated to 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 what must change is the IP subnets associated to the same VLAN ID in different racks, as shown in Table 1.

IP ADDRESS MANAGEMENT AND VLANs[1]		
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

[1] Values of VLANs, IP addresses, and masks are an example (not prescriptive to the design)

Table 1 – IP Address Management and VLAN IDs

The following are among the configurations required per host:

1) VMkernel NIC (vmknic) IP configuration per traffic type in the respective IP subnet.

2) Static route configuration per subnet, to handle proper traffic routing to the respective gateways.

The need for static routing is due to the fact that ESXi support only 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 then be configured on this stack for each ESXi pointing to the gateway deployed on the local ToR, and this allows communication with all the remote VTEPs deployed in different Transport subnets.
- Default: this stack is used for all the other traffic types (vMotion, Management, 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 implies that static routing configuration is required to support inter-subnet communication for the other types of traffic.

Back to our example, a given host 1 in rack 1 (referring to Table 1, R_id is 1) would have the following vmknic configuration:

- Storage vmknic with IP address 10.66.1.10.
- vSphere vMotion vmknic with IP address 10.77.1.10.
- 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 – 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 then 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.

Note: the VXLAN vmknic would have an IP address 10.88.1.10 and leverage a dedicated default gateway configuration (part of the TCP/IP VXLAN stack) 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 and helps automate the deployment as well as the upgrades of the ESXi hosts.

Edge Racks Design

The edge racks host services allowing communication between the logical and the physical networks. This can happen at layer 3 (NSX logical routing) or at L2 (NSX bridging); both these functionalities will be discussed in greater detail in this section.

For the sake of the resiliency of the services offered in the edge racks, it is current best practice to deploy a redundant pair of edge racks, just to protect against the catastrophic failure of an entire rack. As it will become clearer later in this chapter, 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 recommended deployment model to allow VLANs to extend between edge racks is shown in Figure 95.

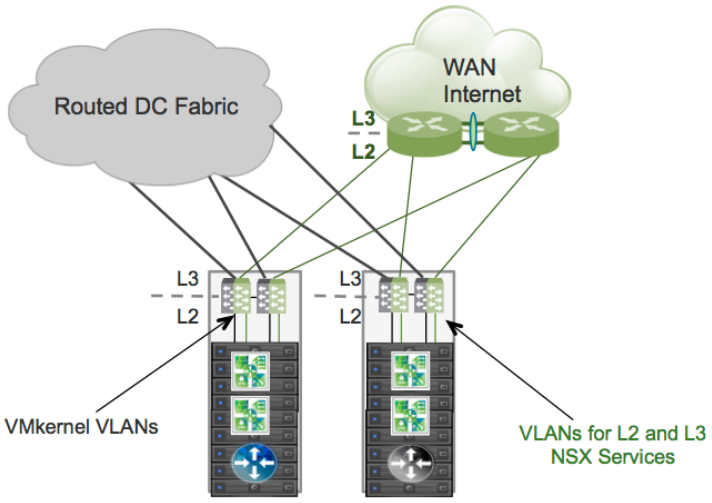


Figure 95 – Connectivity of Edge Racks toward the Physical Network

In this deployment model, redundant pairs of Top-of-Rack (ToR) switches are connected in each edge rack to play a dual role:

- Function as Layer 3 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 default-gateway for all the local VMkernel VLANs used by the servers (Mgmt, VXLAN, vMotion, storage, etc.). The span of those VLANs is limited to each local edge rack (similarly to what happens for the compute and management racks).
- The ToR switches also represent the interface between the DC fabric and the external physical network infrastructure, and as such sometimes they are named “border leafs”. In this case they only provide L2 transport functionality for the VLANs that are used to interconnect to the physical network space (L2 and L3 NSX services). The default gateway for devices connected to those VLANs is usually positioned to a dedicated pair of core routers. A separate VLAN (or set of VLANs) would be used for each deployed tenant.

NSX Edge Deployment Considerations

In the “NSX Edge” section it was mentioned how a pair of Active/Standby Edge Services Gateways can be deployed to provide resiliency to all the services (routing, firewalling, etc.) offered by those NSX component. When focusing specifically on the routing functionalities between the logical and physical networks, there are actually three HA models that can be deployed: Active/Standby, Standalone and ECMP with the latter one representing a newer functionality introduced from NSX SW release 6.1 onward.

As a reminder, Figure 96 highlights the reference topology that will be used for describing the various HA models and that leverages the NSX Edge between the DLR and the physical network (as previously discussed in the “Logical Routing Deployment” section).

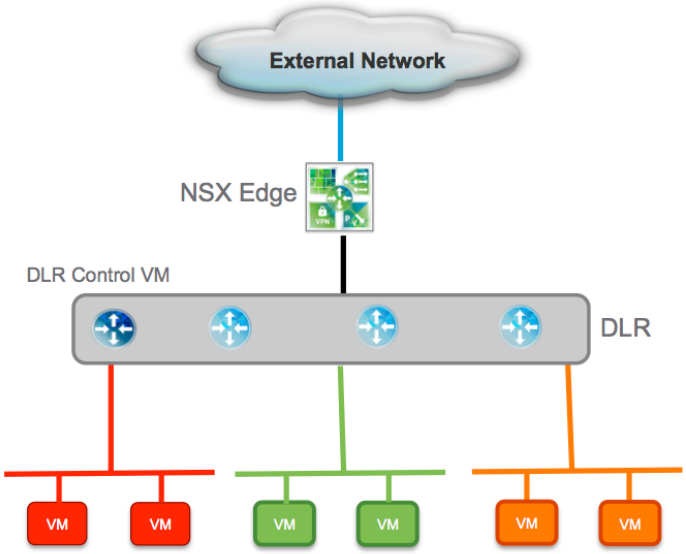


Figure 96 - NSX Logical Routing Reference Topology

The next three sections analyze in detail the HA models mentioned above, specifically focusing on how to provide resiliency to the NSX Edge and DLR Control VM functional components and what is the impact of their failure on the north-south data-plane communications.

Stateful Active/Standby HA Model

This is the redundancy model where a pair of NSX Edge Services Gateways is deployed for each tenant; one Edge functions in Active mode (i.e. actively forwards traffic and provides the other logical network services), whereas the second unit is in Standby state, waiting to take over should the active Edge fail.

Health and state information for the various logical network services is exchanged between the active and standby NSX Edges leveraging an internal communication protocol. The first vNIC interface of type “Internal” deployed on the Edge is used by default to establish this communication, but the user is also given the possibility of explicitly specifying the Edge internal interface to be used. Having this degree of control on the interface to use is important because it allows choosing between a VLAN-backed port-group or a logical switch (VXLAN-backed port-group) as communication channel. The former option mandates the used vNICs of the NSX Edges to be L2 adjacent (i.e. connected to the same VLAN), whereas the former provides more topological flexibility.

Note: it is mandatory to have at least one Internal interface configured on the NSX Edge to b able to exchange keepalives between the Active and Standby units. Deleting the last Internal interface would break this HA model.

Figure 97 highlights how the Active NSX Edge is active both from a control and data plane perspectives. This is the reason why it can establish routing adjacencies to the physical router (on a common “External VLAN” segment) and to the DLR control VM (on a common “Transit VXLAN” segment). Traffic between logical segments connected to the DLR and the physical infrastructure always flow only through the active NSX Edge appliance.

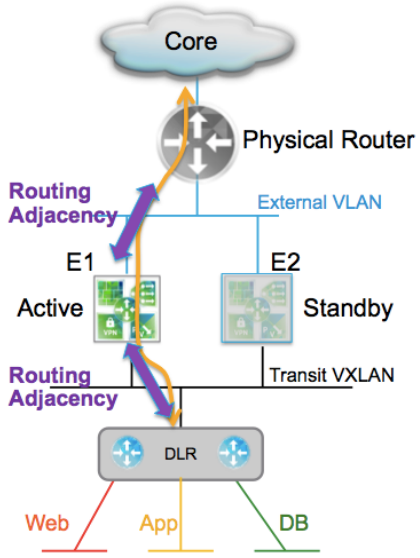


Figure 97 - NSX Edge Active/Standby HA Model

If the Active NSX Edge fails (for example because of an ESXi host failure), both control and data planes must be activated on the Standby unit that takes over the active duties (Figure 98).

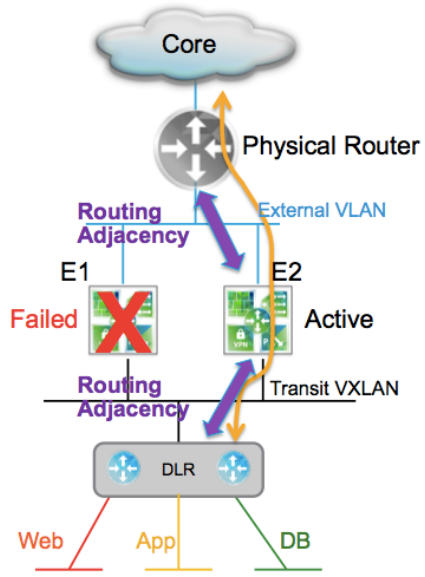


Figure 98 - Traffic Recovery after Active NSX Edge Failure

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 UI or API call) and it is the main factor influencing the traffic outage experienced with this HA model.
- Once the Standby gets activated, it starts all the services that were running on the failed Edge (routing, firewalling, etc.). While the services are restarting, traffic can still be routed leveraging the information in the NSX Edge forwarding table that was kept in synch 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.

  Note: for the LB service only the persistence state is synchronized between the Active and the Standby units.

- 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. Three conditions need to be met for this to happen:

    1. 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 and maintain active the previously established routing adjacencies. This can be achieved by setting the hello/hold-time timers to a value long enough to ensure that the newly activated NSX Edge has enough time to restart the routing protocol before the hold-time on the DLR and the physical router expires. If the hold-time expired, the DLR and the physical router would time out the routing adjacencies, consequently removing from the forwarding tables the routing information learned from the NSX Edge. This would cause a second traffic outage until new adjacencies are re-established and routing information exchanged again. The recommendation is hence to configure OPSF or BGP hello/hold-time timers with a 40/120 seconds value between those devices.

        Note: it is important to stress the fact that those timers are not a determining factor for the length of traffic outage, hence the recommended values are quite high to cover the worst case where a restarting NSX Edge may take longer to activate the network services.

    2. At the data-plane level, the DLR kernel modules and the physical router keep sending traffic to the same next-hop IP address (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 mapping information on those devices.

    3. Once the newly activated NSX Edge has re-started 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 start a new 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 another traffic outage.

- 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 Gateways virtual machines reside. 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 also important to consider the scenario where the DLR Active Control VM fails, to determine what are the consequences of this event in terms of traffic outage.

    The DLR Control VM leverages the same Active/Standby HA model just 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 hosts part of the NSX domain.

    The high value setting (40/120 seconds) 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. In fact, 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) and keep sending data-plane traffic to the “Forwarding Address” IP address 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 keep the adjacency up and continue forwarding the traffic.

Standalone HA Model (NSX 6.0.x Releases)

The standalone HA model inserts two independent NSX Edge appliances between the DLR and the physical network (Figure 99) and it is supported when running NSX 6.0.x SW releases.

Note: from NSX SW release 6.1 it is possible instead to adopt the evoluted ECMP HA model discussed in the next

section.

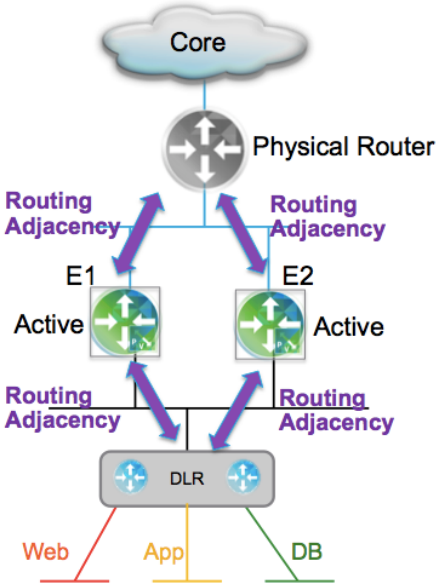


Figure 99 - NSX Edge Standalone HA Model

In this case, both NSX Edge devices are active, both from a data and control planes point of view and can establish routing adjacencies with the physical router and the DLR Control VM. However, in all the 6.0.x NSX SW releases the DLR cannot support Equal Cost Multi-Pathing. As a consequence, even when receiving routing information from both NSX Edges for IP prefixes existing in the physical network, the DLR only installs in its forwarding table one possible next-hop (active path). This implies that all traffic in the south-to-north direction will only flow through a single NSX Edge and cannot leverage both appliances. Traffic load balancing may instead happen in the north-to-south direction since most physical routers and switches are ECMP capable by default.

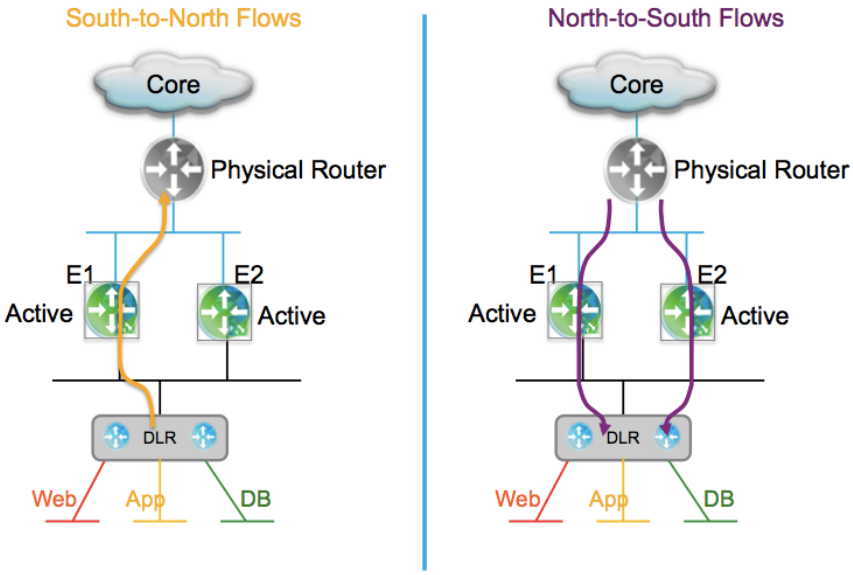


Figure 100 - Traffic Flows with the Standalone HA Model

The asymmetric behavior shown in figure above by traffic flows in opposite directions may prevent communication if firewall services are enabled on the NSX Edge Services Gateways. The recommendation is hence to disable the centralized firewall when deploying this NSX Edge redundancy model. Alternatively, it is possible to change the cost of the Edge interfaces to ensure symmetry of traffic flows in both north-to-south and south-to-north directions, even if this may increase the overall operational complexity of the solution.

Traffic outage experienced when an NSX Edge fails is now dependent on how fast the physical router and the DLR can detect the failure and consequently update their forwarding tables to start pointing to the other (previously unutilized) NSX Edge (Figure 101).

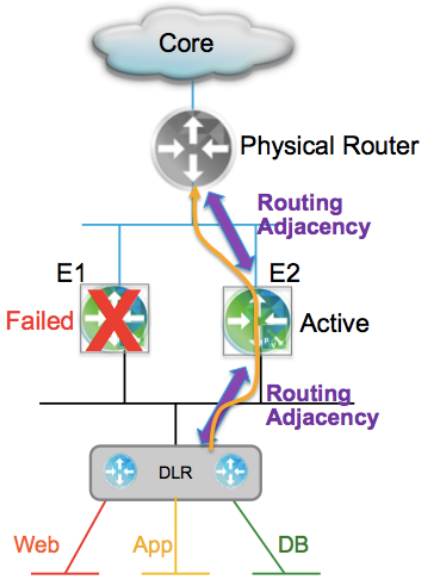


Figure 101 - Traffic Recovery after Standalone NSX Edge Failure

Some important considerations regarding this standalone HA model are listed below:

- In this model, the recommendation is to tune aggressively the routing hello/hold-time timers (to the minimum supported values 1/3 seconds) on the NSX Edge, on the DLR Control VM and on the physical router. This is required because the routing protocol hold-time is in this case the main factor influencing the entity of the traffic outage. When E1 fails, the physical router and the DLR keep 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 to start using E2 as next-hop (if the physical router was already using also E2 for north-south flows, it simply has to remove E1 as a viable next-hop).
- The recommendation with this HA model is to deploy two independent Active HA Edge Gateways, without relying on the Active/Standby functionality. This is because in order to reduce traffic outage to a minimum, it is preferable to lose an Edge rather than waiting for the Standby to take over.
- That said, since the Edge Gateway is still a virtual machine, it is also recommended to leverage vSphere HA to ensure the failed Edge can be restarted on a different ESXi host. At that point, it will start being used by the physical router for north-south communication and it will remain ready to be used by the DLR (for south-north flows) should E2 fail at some point.
- This Standalone HA model should be adopted only when there is no need to deploy any logical services on the edge (firewall, load-balancing, NAT, etc) other than routing. This is because the two deployed Edge Gateways are not syncing any information between them, so no statefulness for those logical services can be achieved. Firewalling can still be deployed in a distributed fashion leveraging the DFW functionalities but, as previously mentioned, it is recommended to disable the FW logical function on the NSX Edges, to be able to tolerate the asymmetric traffic path behavior shown in the example in Figure 100.

  Note: in NSX SW releases 6.0.x, the logical FW function can only be disabled (via UI or APIs) when deploying the X-Large form factor for the NSX Edge. From SW release 6.1, this can instead be done for any form factor.
- The aggressive setting of routing protocol timers required for this standalone HA model has an important implication when dealing with the specific failure scenario of the Active Control VM. This failure would now cause both NSX Edges to bring down the routing adjacencies previously established with the Control VM (in less than 3 seconds). This means that both NSX Edges flush their forwarding tables removing all the prefixes originally learned from the DLR, causing the north-south communications to stop.

  A possible workaround for this issue consists in installing static routing information in the routing tables of both NSX Edges, as shown in Figure 102.

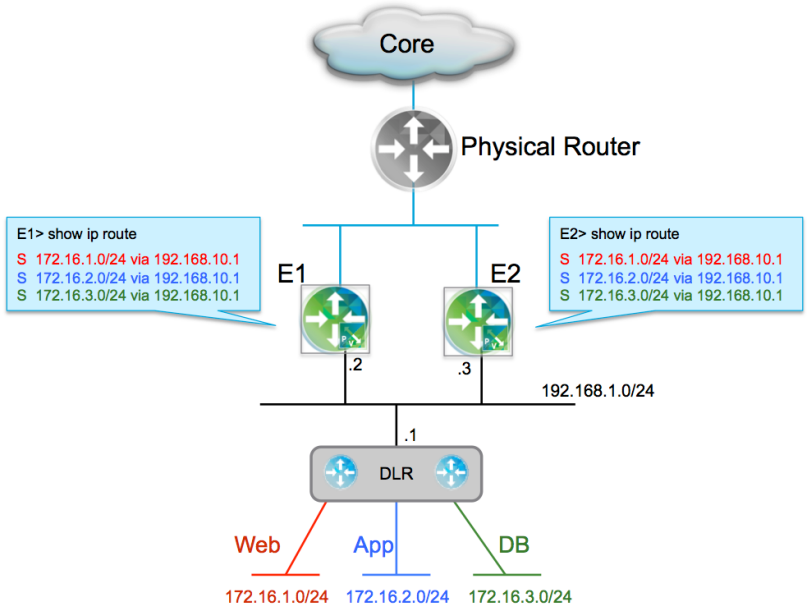


Figure 102 - Configuring Static Routes on the Active NSX Edges

That way, when the adjacencies time out and the dynamically learned routes (associated to 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 OSPF to advertise them toward the physical routers.

Note: there is no need to configure static routing information on the DLR side. Once the Active Control VM fails and the Standby takes over and begins restarting the routing services, the routing information in its forwarding table (and consequently the routing tables in the kernels of the ESXi hosts) remains unchanged independently from the fact that the adjacencies between the DLR Control VM and the ESG devices are lost. This means that all the traffic directed to the northbound IP prefixes will keep 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 formed again and IP prefixes will be dynamically exchanged with them. New routing information received from the Edges (if present) will be at this point pushed to the controller and consequently to the kernels of the ESXi hosts.

The task of installing static routes on the edge can be automated, so that every time a new logical switch is created, a corresponding static route is added on both NSX Edge Gateways, whereas it is removed if the logical switch is deleted or disconnected from the DLR.

ECMP HA Model (NSX 6.1 Release Onward)

NSX software release 6.1 introduces support for a new Active/Active ECMP HA model, which can be considered the improved and evoluted version of the previously described Standalone one.

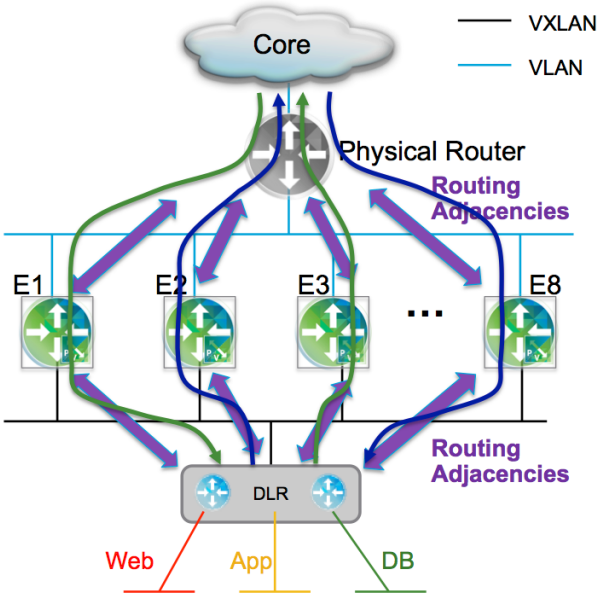


Figure 103 - NSX Edge ECMP HA Model

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 table. Focusing for the moment on the ECMP capabilities of the DLR, this means that up to 8 active NSX Edges can be deployed at the same time and all the available control and data planes will be fully utilized, as shown in Figure 103.

This HA model provides two main advantages:

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

Notice from the diagram in figure above that traffic flows are very likely to follow an asymmetric path, where the north-to-south and south-to-north legs of the same communications are handled by different NSX Edge Gateways. 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 way the physical router distributes north-to-south flows depends instead on the specific HW capabilities of that device.

Traffic recovery after a specific NSX Edge failure happens in a similar fashion to what described in the previous standalone HA model, as the DLR and the physical routers would have to quickly time out the adjacency to the failed unit and re-hash the traffic flows via the remaining active NSX Edge Gateways (Figure 104).

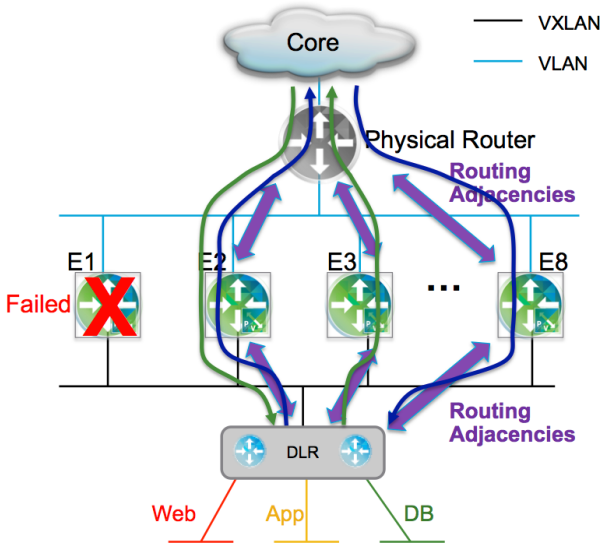


Figure 104 - Traffic Recovery after ECMP NSX Edge Failure

Some deployment considerations specific to this HA model:

- Once again, the length of the outage is determined by how fast the physical router and the DLR Control VM can time out the adjacency to the failed Edge Gateway. It is hence recommended to tune aggressively the hello/hold-time timers (1/3 seconds).
- However, differently from the HA standalone model, 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 better recovery functionality with this ECMP HA model.
- When deploying multiple Active NSX Edges, there is no support for state synchronization for the various logical services available on the NSX Edge Services Gateway. It is hence recommended to leverage DFW and one-arm load-balancer deployments and deploy only routing services on the different NSX Edge Gateways.
- The diagram in Figure 103 showed a simplified view of the physical network infrastructure north of the NSX Edge Gateways. It is expected that a redundant pair of physical routers would always be deployed in a production environment, and this raises the question on how to best connect (from a logical perspective) the NSX Edges to those routers.

  The first decision to make is how many logical uplinks should be deployed on each NSX Edge: the recommended design choice is to always map the number of logical uplinks to the number of VDS uplinks available on the ESXi servers hosting the NSX Edges.

  In the example shown in Figure 105 below, all the deployed Active NSX Edge Gateways E1-E8 are running on ESXi hosts connected to the Top-of-Rack switches via 2 uplinks. As a consequence, the recommendation is to deploy two logical uplinks on each NSX Edge. 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.

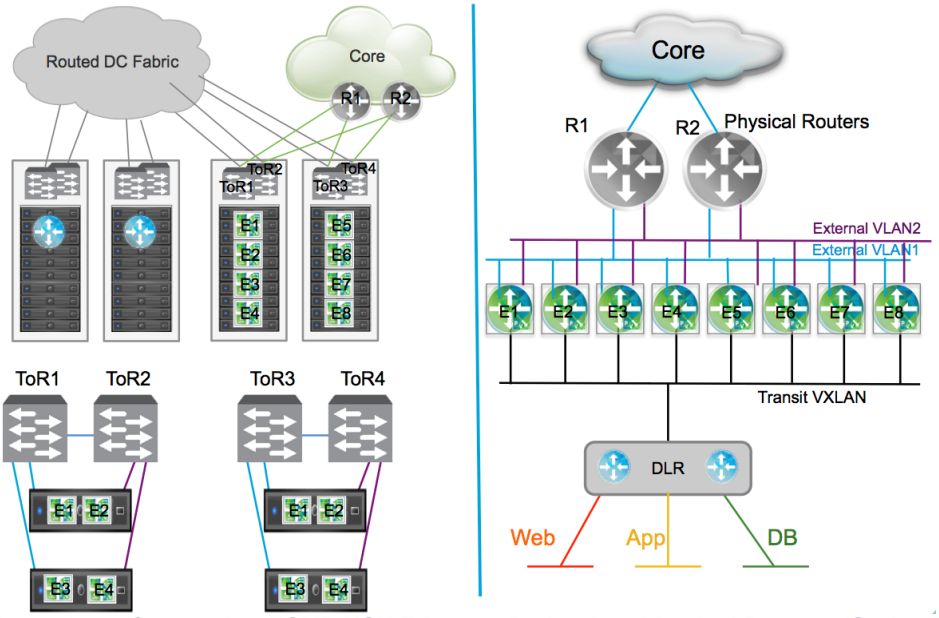


Figure 105 – Connecting ECMP NSX Edges to Redundant Physical Routers (Option 1)

Each external VLAN should then be carried only on one ESXi uplink (in the example above "External VLAN1" is carried on the uplink toward ToR1/ToR3 and "External VLAN2" on the uplink toward ToR2/ToR4). This is done so that under normal circumstances both ESXi uplinks can be concurrently utilized to send and receive north-south traffic, even without requiring the creation of a port-channel between the ESXi host and the ToR devices. Also, 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 and the Edge would continue sending and receiving traffic leveraging the second logical uplink (the second physical ESXi NIC interface).

In order to build a resilient design capable of tolerating the complete loss of an edge rack, it is also recommended to deploy two sets of four Edge Gateways in two separate edge racks. This implies the need to extend the External VLANs between the two edge racks; as previously discussed 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 106.

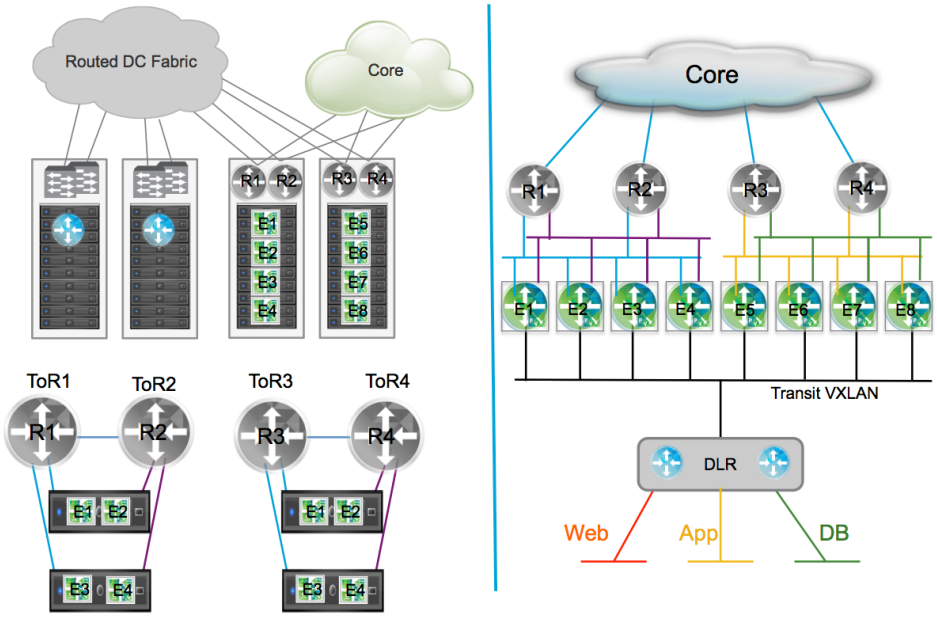


Figure 106 - Connecting ECMP NSX Edges to Redundant Physical Routers (Option 2)

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. This implies that it is now possible to deploy two independent pairs of physical routers and position them as Top-of-Rack devices (instead than connecting the edge racks to an external L2 infrastructure). NSX Edges deployed in edge rack 1 establish routing peering with R1 and R2, whereas NSX Edges deployed in edge rack 2 peer with R3 and R4.
- The VLAN IDs associated to the peering segments in separate edge racks could actually be the same (i.e. VLAN 10 and VLAN 11 can be used for both edge rack 1 and edge rack 2), but they will be associated to separate IP subnets.
- Since we don't want to extend VLANs between racks, the recommendation is to ensure that NSX Edge VMs can be dynamically moved (for example by vSphere HA) between ESXi hosts belonging to the same edge rack (and never across racks).

- Finally, the same considerations on the handling of the failure of the control VM made for the Standalone HA model apply also in this ECMP HA model. As a consequence, there is the same need for configuring static routes on all the deployed Edges for all the IP subnets created in the logical space and that need to be reachable from the external network.

NSX Layer 2 Bridging Deployment Considerations

The NSX L2 Bridging functionality is used to provide access to logical networking 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. What is important to discuss now is what are the design implications of deploying an NSX L2 bridging function on the design of the network.

First of all, let's recall that the L2 bridging function for a given VXLAN-VLAN pair is always performed on one specific ESXi host. Since the L2 bridge instance configuration is always associated with the configuration of a specific DLR, the active bridge functionality is performed by the specific ESXi hosts where the active control VM for that DLR is running.

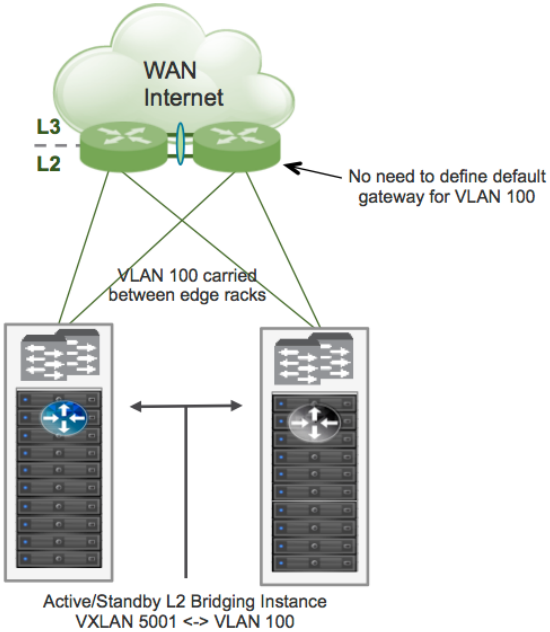


Figure 107 - Active/Standby NSX L2 Bridging Instance

As shown in Figure 107, 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. That said, it is possible to create multiple sets of bridging instances and associate them with different DLRs, so to be able to spread the bridging load across different

ESXi hosts.

Some important deployment considerations are below:

- Assuming 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 to 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.
- It is recommended, when possible, 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 they belong to only to those racks and not to the other compute racks connected to different ToR switches.

  Note: when building the separate L2 physical infrastructure depicted in figure above, there is the option of deploying a dedicated set of racks for bare-metal servers, with their own Top-or-Rack switches directly hooked up to the switched physical network.
- The VLAN where traffic is bridged to (VLAN 100 in the example above) is carried on the same switching infrastructure previously introduced and utilized for the L3 North-South communication. The default gateway for the bare metal servers could be deployed in two ways, shown in Figure 108.

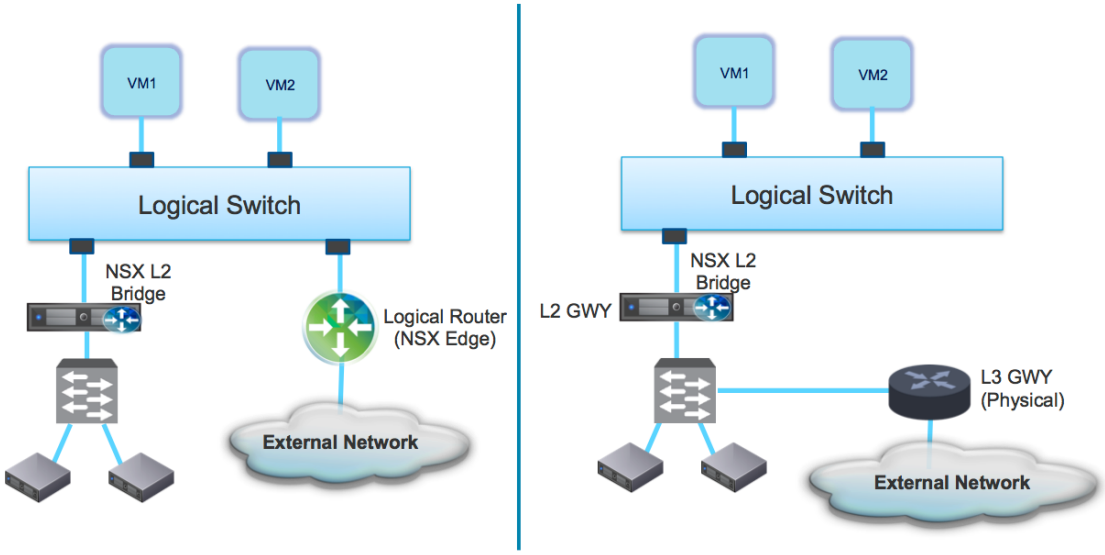


Figure 108 - Default Gateway Deployment Options for Physical Servers

1. The default gateway is deployed in the logical space and the physical servers access it through the NSX L2 bridging function. In this case, it is important to clarify that currently it is not supported to have an active L2 bridge on a logical switch that is also connected to the Distributed Logical Router. As a consequence, the default gateway for the VMs connected to the logical switch and the bare-metal servers needs to be placed on an NSX Edge Services Gateway.
2. The default gateway is deployed in the physical space and the virtual machines connected to the LS access it via the L2 Gateway service.

- The keepalive messages between the active and standby Control VMs can be exchanged leveraging VXLAN, so it is not required to extend an additional VLAN for that purpose. Alternatively, it is possible to exchange those messages on one of the bridged VLANs that are already extended across the edge racks.

  Note: as previously mentioned, the keepalives can only be exchanged on Internal Edge interfaces (and not on the Uplinks).

Conclusion

VMware NSX-v network virtualization solution addresses current challenges with physical network infrastructure and brings 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 then reproduce the properties and services in the virtual environment.

NSX-v reproduces in the logical space typical network services traditionally delivered by the physical infrastructure, as switching, routing, security, load-balancing, Virtual Private Networking and allows to extend connectivity into the logical space to physical devices connected to the external network (Figure 109).

Figure 109 - Logical Network Services Provided by NSX-v

References

[1] *What's New in VMware vSphere 5.5*
http://www.vmware.com/files/pdf/vsphere/VMware-vSphere-Platform-Whats-New.pdf

[2] *vSphere 5.5 Configuration Maximums*
http://www.vmware.com/pdf/vsphere5/r55/vsphere-55-configuration-maximums.pdf