White Paper

VXLAN Multipod Design for Intra-Data Center and Geographically Dispersed Data Center Sites

May 17, 2016

Authors

Max Ardica, Principal Engineer INSBU
Patrice Bellagamba, Distinguish System Engineer EBG Enterprise Networking
Yves Louis, Technical Solution Architect EBG Data Center

Key Contributors

Lukas Krattiger, Principal TME Enterprise Switching
Lilian Quan, Principal Engineer INSBU

Contents

Summary

Recently, fabric architectures have become a common and popular design option for building new-generation data center networks. Virtual Extensible LAN (VXLAN) with Multiprotocol Border Gateway Protocol (MP-BGP) Ethernet VPN (EVPN) is becoming the de-facto standard technology used for deploying network virtualization overlays in data center fabrics.

Data center networks usually require the interconnection of separate network fabrics (referred to as pods in this document) that may also be deployed across geographically dispersed sites. Consequently, organizations want to explore how the use of the VXLAN technology can help provide disaster avoidance and workload mobility by extending Layer 2 and Layer 3 connectivity between separate pods.

From a high-level perspective, two approaches can be considered.

In the first approach, every data center site is deployed and managed as an independent domain. This model increases the resiliency of the overall design by deploying each site as an independent availability zone. This model is usually referred to as the multisite design, and it often is used to address the disaster recovery use case, although it can be extended to the deployment of active-active data centers. A data center interconnect (DCI) technology (Cisco® Overlay Transport Virtualization [OTV], Virtual Private LAN Service [VPLS], Provider Backbone Bridging [PBB] EVPN, or even VXLAN) is used to extend Layer 2 and Layer 3 connectivity across separate sites. This approach is also often considered in migration scenarios in which a new VXLAN fabric must be connected to an existing data center network, usually built with more traditional technology (spanning tree, virtual port channel [vPC], Cisco FabricPath, etc.). This approach is not discussed in this document.

The second approach treats multiple data center fabrics as one single large administrative domain, which simplifies the operational aspects of the deployment. This model is referred to as the VXLAN multipod design. It can be applied when the pods represents rooms deployed in the same physical data center location or when the pods represent separate data center locations. This document discusses this multipod model, with deployment considerations for the specific use case of interconnecting geographically dispersed pods.

Goals of This Document

This document discusses the deployment model for extending the VXLAN EVPN fabric across geographically dispersed locations or between separate data center rooms deployed within the same location (for example, within an enterprise campus).

It provides design and configuration recommendations that apply specifically to the interconnection of geographically dispersed pods. It also emphasizes the enhanced features that Cisco provides in its VXLAN EVPN control-plane implementation.

Note that unless you are using the EVPN control plane and Cisco enhanced features, you should avoid extending VXLAN outside a physical data center location.

Note: Cisco does **not** recommend the use of the basic data-plane-only implementation of VXLAN, also known as VXLAN flood and learn, for interconnecting data center fabrics.

Introduction

The introduction of the MP-BGP EVPN control plane represents an important evolutionary step for VXLAN as a data center networking technology. The elimination of the flood-and-learn behavior makes this technology of greater interest for extending Layer 2 and Layer 3 communication beyond a single fabric.

VXLAN was originally designed for intra–data center fabric deployments: that is, for deployments within the same data center. When VXLAN is extended to provide Layer 2 and Layer 3 connectivity between fabrics that are deployed in separate physical locations, it is often compared to a technology such as OTV, which was built from the start for DCI and hence offers more native DCI functions. Note that VXLAN and OTV both use a similar data-plane encapsulation format, but differ in the control-plane technology used (Intermediate System–to–Intermediate System [IS-IS] for OTV and MP-BGP EVPN for VXLAN).

In comparing these two technologies, however, you should not focus exclusively on their data-plane and control-plane details. Instead, you should analyze the services and functions that they can offer for interconnecting geographically dispersed data center fabrics.

This document does not discuss the viability of VXLAN as a DCI technology (as an alternative to OTV, VPLS, or PBB-EVPN). Instead, it focuses on a specific deployment model: the VXLAN multipod design. This design extends the use of VXLAN to interconnect separate data center rooms and sites, generically called pods, that are part of a single administrative domain.

Although the VXLAN multipod design is likely to be more commonly adopted for interconnecting pods deployed in the same physical data center (as a mean of scaling out the data center fabric in a structured and modular fashion), this document also discusses some of the specific design considerations and recommendations for interconnecting geographically dispersed pods in separate data center sites.

Note: This document assumes that the reader has knowledge of how to deploy VXLAN with the EVPN control plane. For more configuration information, see the VXLAN network with MP-BGP EVPN control-plane design guide at http://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/guide-c07-734107.html.

VXLAN Multipod Design Overview

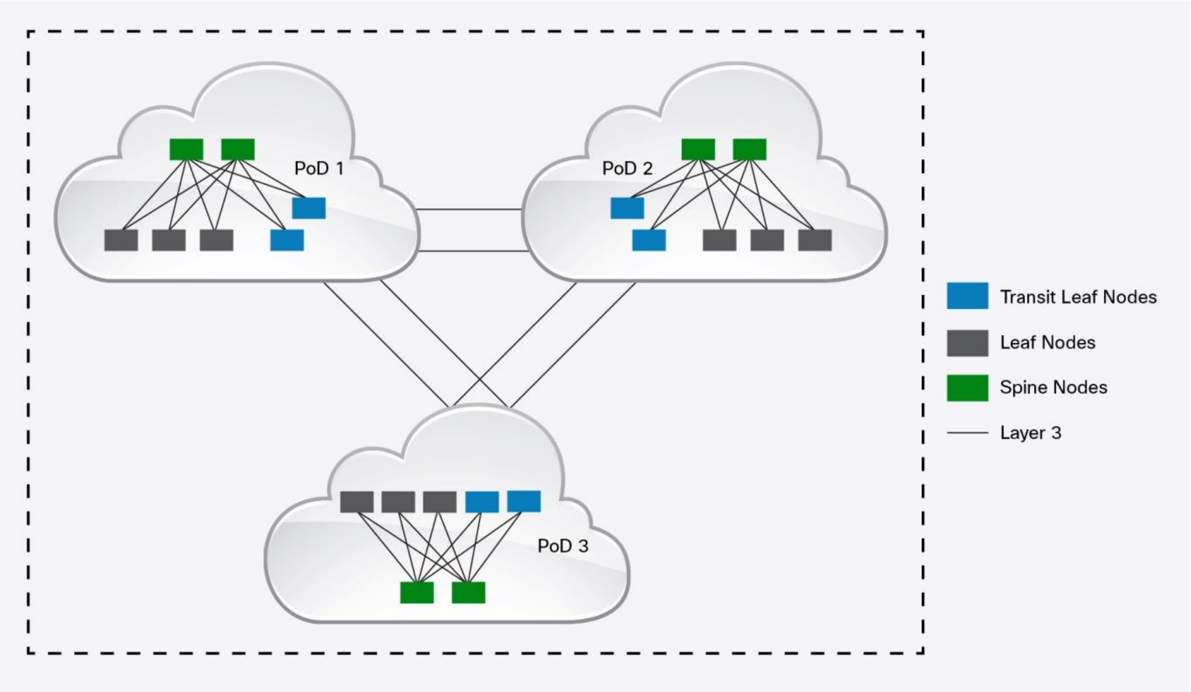
VXLAN today functions as the de-facto standard overlay technology proposed by multiple vendors for the deployment of next-generation data center fabrics. The introduction of the MP-BGP EVPN control plane has improved the resiliency of the overall solution, making VXLAN even more a mainstream technology for providing multitenancy, Layer 2 adjacency, and mobility across the data center network.

The different pods are normally connected with point-to-point fiber connections when they are deployed in rooms within the same physical location. Interconnection of pods dispersed across a metropolitan area usually uses direct dark-fiber connections (or dense wavelength-division multiplexing [DWDM] circuits) to extend Layer 2 and Layer 3 connectivity end-to-end across locations. However, from a technical point of view, the same design can apply to scenarios in which a generic Layer 3 network is used to interconnect the remote pods deployed at longer distances. In those cases, you must be sure to assess the impact of distance and latency on the applications deployed across different pods as well as verify that the available bandwidth is dimensioned accordingly.

Several physical topologies can be deployed for the VXLAN multipod design:

- The first option, shown in Figure 1, introduces a pair of transit leaf nodes, which represent the Layer 3 transit devices, that interconnect the data center pods through direct links.

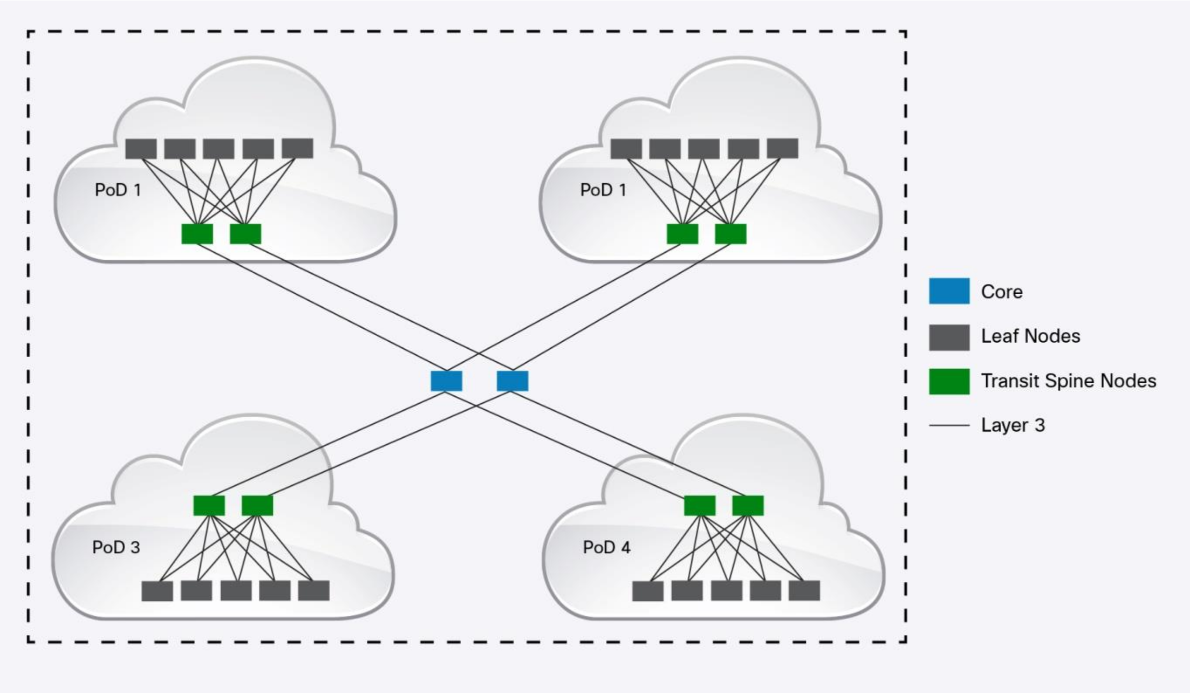
Figure 1. VXLAN Multipod Design Interconnecting Leaf Nodes



As will be clarified later in this document, the transit leaf nodes can be pure Layer 3 devices. They can also be used as computing leaf nodes (locally connecting endpoints) and as border leaf nodes (providing Layer 3 connectivity to the WAN edge devices).

- A second deployment model interconnects separate pods through the spine layer devices, as shown in Figure 2.

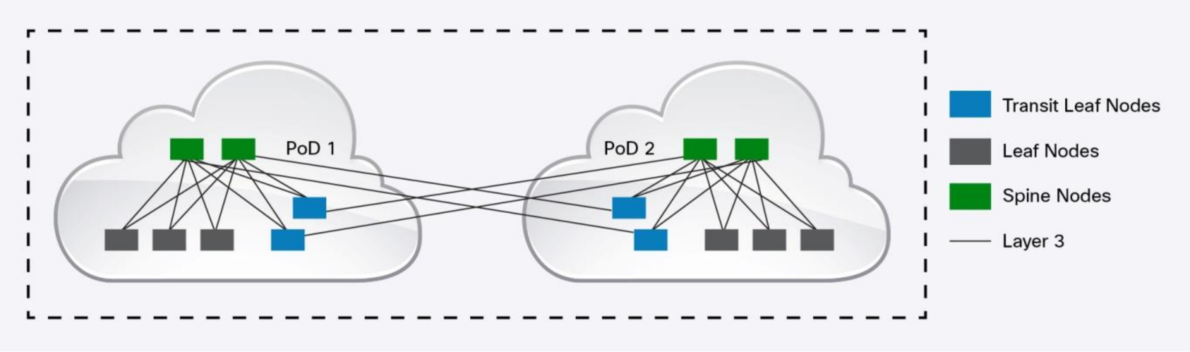
Figure 2. VXLAN Multipod Design Interconnecting Spine Nodes



Note that no additional functions are required on the spine layer devices, other than the capability to route VXLAN traffic between leaf switches deployed in separate pods. This is the case because VXLAN tunnels are established between leaf nodes, independent of the pod to which they are connected.

- A third option is an M-shaped design (a partial mesh), shown in Figure 3.

Figure 3. VXLAN Multipod M-Shaped Design



In this scenario, the transit leaf nodes connect to both the local spines and the spines deployed in the other pods. The number of transit leaf nodes and uplinks to use it is determined mainly by the amount of resiliency that needs to be built into the system and by the amount of interpod traffic expected.

The design considerations presented in this document are based on the validation of the specific topology shown in Figure 1. However, they can easily be extended to the other two models because, as in a single VXLAN pod deployment, in a multipod design the VXLAN tunnels run end to end between VXLAN tunnel endpoint (VTEP) devices that can belong to separate pods. VXLAN encapsulation and decapsulation occurs only on the ingress and egress VTEPs, respectively. The other devices along the forwarding path only need to route the Layer 3 encapsulated VXLAN packets. This approach provides end-to-end, single-tunnel overlay data-plane processing.

Note, when considering the distance between geographically dispersed pods, that the multipod design is usually positioned to interconnect data center fabrics that are located at short distances. DWDM in protected mode and dark fiber, respectively, offer 99.997 and 99.999 percent availability. It is crucial that the quality of the DWDM links is optimal. Consequently, the optical services must be leased in protected mode. They must also have the remote port shutdown feature enabled, to allow immediate detection of physical link-down events across the entire system. Because the network underlay is Layer 3 and can be extended across long distances, you should also check the continuity of the VXLAN tunnel established between pods. You can use Cisco Bidirectional Forwarding Detection (BFD) for that purpose.

When and Why to Use the VXLAN Multipod Design

After VXLAN EVPN has been selected as the technology of choice for building a greenfield, or completely new, data center pod, it becomes logical to extend VXLAN between fabrics that are managed and operated as a single administrative domain. This choice makes sense because a multipod fabric functionally and operationally is a single VXLAN fabric, and its deployment is a continuation of the work performed to roll out the pod, simplifying the provisioning of end-to-end Layer 2 and Layer 3 connectivity.

Because all the devices deployed in the interconnected pods are functionally part of the same VXLAN fabric, you also must consider the scalability of the solution: the maximum number of leaf and spine nodes and VXLAN segments, etc. For very large-scale designs, you may need to split the deployment into separate independent VXLAN fabrics (the previously mentioned multisite approach).

The VXLAN multipod design is targeted at two main use cases:

- Interconnection of pods that are deployed within the same physical data center location, such as in different rooms at the same site: The design can be useful, for example, in a scenario in which the cabling has been laid out for the deployment of a classic three-tier network infrastructure (access, aggregation, and core layers) and cannot easily be adapted to a leaf-and-spine topology. Building separate pods interconnected by a core (or super-spine) layer, as shown earlier in Figure 2, would be a less expensive solution than recabling the entire data center. The same considerations apply when the different pods are mapped to separate buildings on the same campus and there is not enough connectivity between buildings to fully mesh all the leaf nodes with the spine nodes.
- Interconnection of pods that are geographically dispersed, as in different physical data centers deployed in a metropolitan area: The design recommendations for this scenario are a superset of those described for the previous use case. The goal is to increase as much as possible the resiliency and the optimal behavior of the overall design by, for example:
    - Deploying the underlay and overlay control planes in a more structured fashion (using Open Shortest Path First [OSPF] in multiple areas, separate BGP autonomous systems, etc.)
    - Enabling storm control on the geographical links to reduce the fate sharing between pods

- Enabling functions such as Bridge Protocol Data Unit (BPDU) Guard by default on the edge interfaces to help ensure that the creation of a Layer 2 backdoor connection between sites does not cause an end-to-end Layer 2 loop that affects all the pods

This use case is the main focus of this document. The following sections discuss in greater detail the design recommendations listed here and provide specific configuration and deployment guidance.

Deployment of a VXLAN Multipod Design

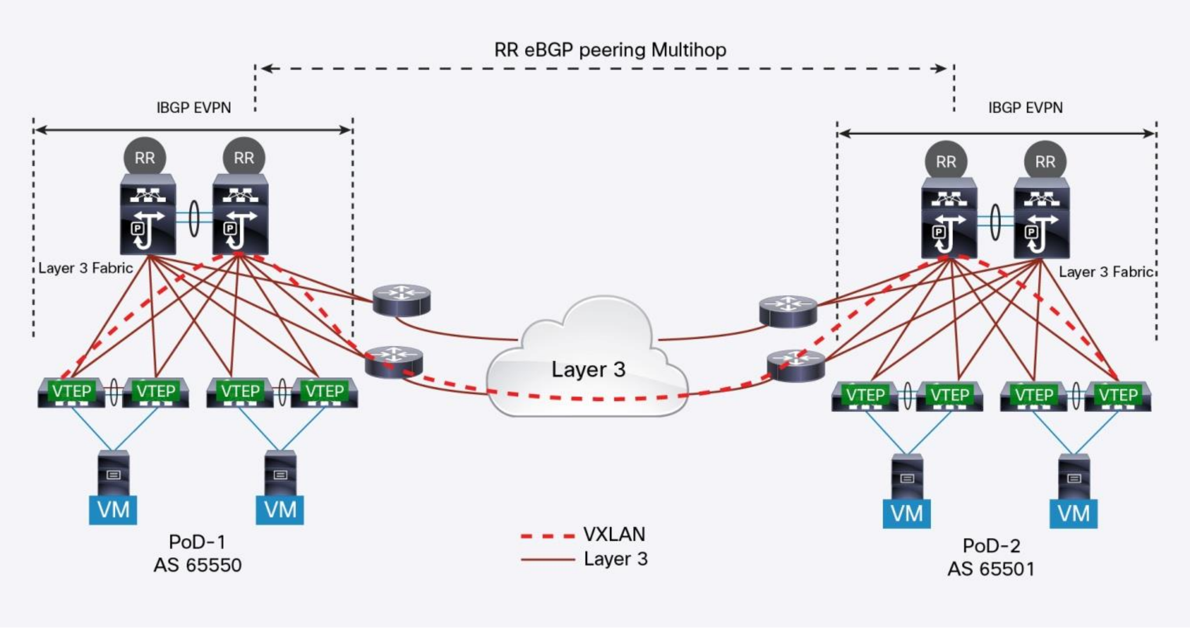
This section provides detailed guidance about how to deploy a VXLAN multipod design.

Design Overview

As previously mentioned in the introductory section, the VXLAN multipod design extends the use of VXLAN EVPN end to end across different data center rooms and sites that are managed and operated as a single administrative domain.

Figure 4 shows one possible multipod deployment model in which the pods are connected through dedicated Layer 3 network devices. Alternatively, direct links configured as Layer 3 interfaces on leaf nodes can be used.

Figure 4. Use of Direct Layer 3 Links for VXLAN Multipod Deployments

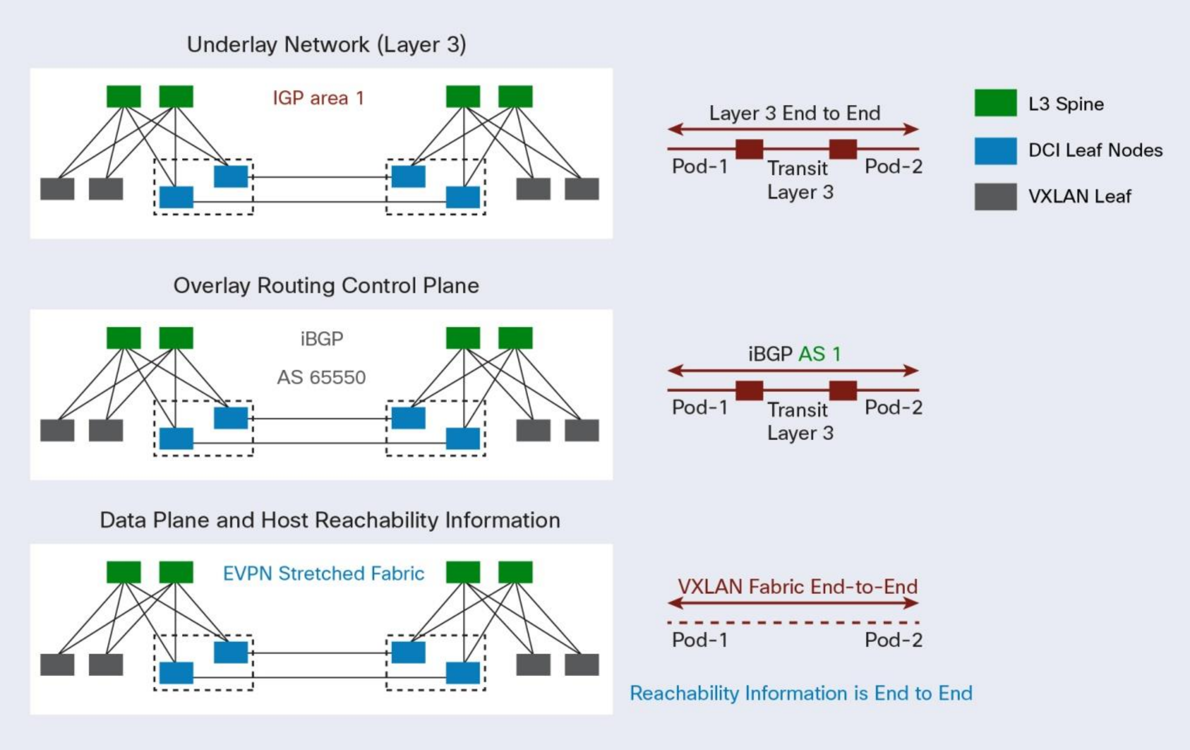


Dark fiber connections or DWDM circuits are the most common options for interconnecting rooms and sites with direct links. For dark fiber, Cisco offers various optical transceivers, depending on the distance between pods. Please refer to the transceiver data sheets available on the Cisco website for updated information.

At a high level, multiple design choices are possible for the deployment of the VXLAN multipod solution. Note that independent of the chosen deployment model, the overall functional behavior of the multipod fabric is essentially the same, because switch and endpoint information must be propagated end to end across all the deployed pods, as will be discussed in this document.

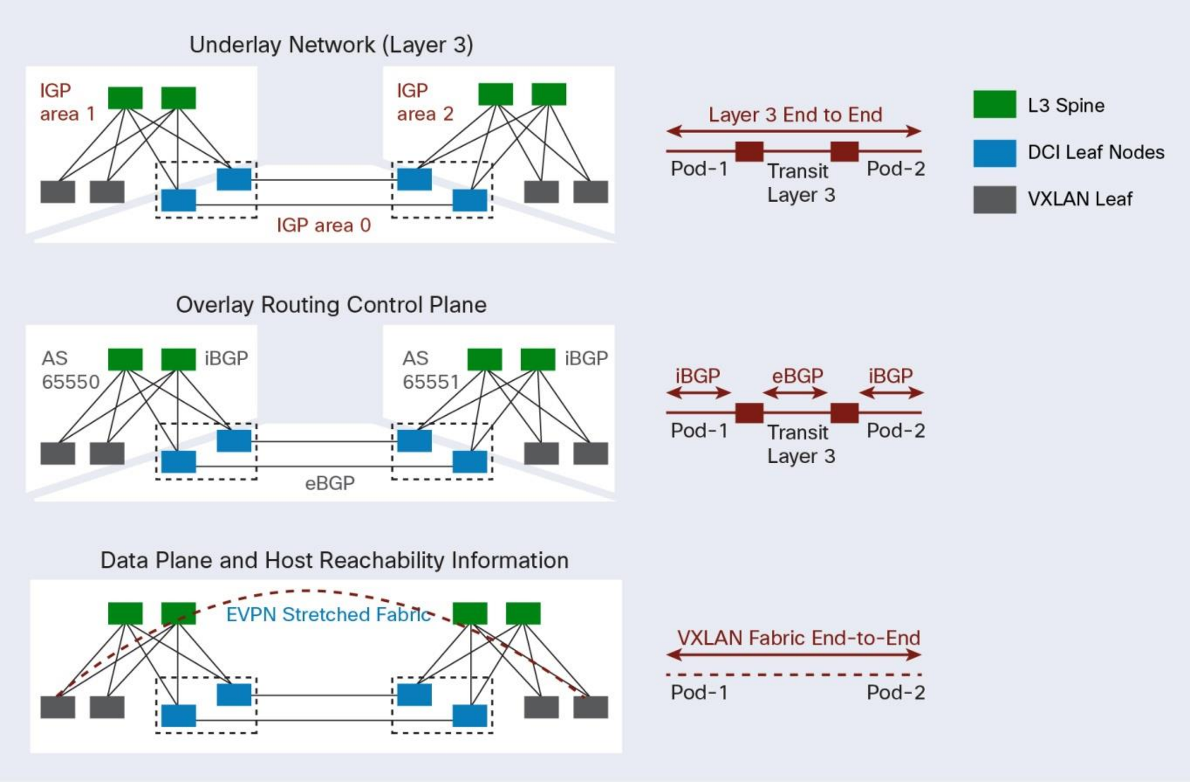
For multipod deployments in a single physical location, a model using a single control plane for underlay and overlay connectivity, as shown in Figure 5, is often preferred.

Figure 5. Single Control Plane for Underlay and Overlay Connectivity



Although technically and functionally the option shown in Figure 5 is fine, the deployment model discussed in this document recommends separation of the control planes across geographically dispersed pods to provide a simpler and more structured design, as shown in Figure 6.

Figure 6. Multipod Deployment with Control-Plane Separation Across Pods



High-level deployment considerations for this model include the following:

- The underlay control plane is used to exchange reachability information for the VTEP IP addresses (the VTEP is the device that performs the VXLAN encapsulation and decapsulation tasks and is usually positioned at the leaf layer). In the most common VXLAN deployments, IGP (OSPF, IS-IS, or Enhanced Interior Gateway Routing Protocol [EIGRP]) is used for this purpose.

  In the example in Figure 6, OSPF is the IGP mechanism of choice, and each pod is deployed in a separate OSPF area, with Area 0 used on the interpod links. Multiple areas can be useful when interconnecting pods deployed in separate physical locations, because it reduces the effects of interpod link bounce, normally a concern with lower-quality geographical links.

  To achieve better underlay (IGP) separation between the different pods, you can use BGP for the transit routing between the pods. With BGP used in the transit network, all the VTEP IP addresses still need to be exchanged between the pods. Nevertheless, the nodes in the transit network itself don’t need to learn all the VTEP IP addresses, and summarization can be used (External BGP [eBGP] multihop between transit leaf nodes). In addition to state reduction in the transit network, the impact of IGP poisoning can be mitigated. The propagation of false information can be prevented, allowing the remote pod to function independently. With eBGP, routing policies can easily be implemented, allowing intra-area changes to be masked in the transit network.

Note: OSPF is not the only choice for the underlay protocol, although it is the most common. Any IGP or even BGP option can be used for the same function. However, a recommended practice is to build a control plane for the transport network independent of the overlay control plane. The deployment of IGP (OSPF, IS-IS, etc.) in the underlay offers this separation of underlay and overlay control protocols. This approach provides a lean routing domain for the transport network that consists of only loopback and point-to-point interfaces. At the same time, MAC and IP address reachability for the overlay exists in a different protocol: MP-BGP EVPN. When interconnecting geographically dispersed pods, you can also mix IGP and BGP for the deployment of the underlay control plane. However, this mixture requires you to configure mutual redistribution between those protocols. To keep the operational aspects of the solution discussed here simpler, this document uses the same IGP option end to end, even with the more structured design shown earlier in Figure 6.

- The overlay control plane is used by the VTEP devices to exchange tenant-specific routing information for endpoints connected to the VXLAN EVPN fabric or to distribute inside the fabric IP prefixes representing external networks. MP-BGP EVPN is the control plane used in VXLAN deployments. The validated design suggests that you deploy each pod in a separate MP-iBGP autonomous system (AS) interconnected through MP-eBGP sessions. When compared to the single autonomous system model depicted in Figure 5, the model using MP-eBGP EVPN sessions simplifies interpod protocol peering. At the same time, it requires some additional configuration tuning to preserve Cisco Evolved Programmable Network (EPN) route attributes across pods (for more information, see the "Overlay Network Deployment Considerations" section of this document). In some case, the use of MP-eBGP between geographically dispersed pods may be required, because each fabric may already be deployed as part of a different autonomous system for interconnection to the external routed domain.
- The transit leaf nodes interconnecting the different pods through the interpod links normally simply perform Layer 3 routing functions, allowing VXLAN encapsulated frames to be carried transparently between VTEPs deployed in separate rooms and data center sites. However, depending on the specific requirements of the deployment, those devices can also be used to interconnect endpoints (thus assuming the role of computing leaf nodes) or to provide connectivity to the external Layer 3 network domain (thus becoming border leaf nodes).

The lab setup used to validate the design discussed in this document included the following components:

- All the devices used to build the pods are Cisco Nexus® 9000 Series Switches. Various Cisco Nexus 9300 platform switches are used as leaf nodes, and Cisco Nexus 9500 platform modular switches are deployed as spines.
- The minimum software release recommended to deploy the design discussed in this document is Cisco NX-OS Software Release 7.0(3)I2(2b). However, be sure to consult the release notes to determine the recommended software release to use before implementing a production deployment.

Underlay Network Deployment Considerations

The first step in a VXLAN deployment is to configure the network that will carry the VXLAN encapsulated traffic to allow connectivity between endpoints connected to the defined logical networks. The physical network is usually referred to as the underlay. The following sections describe the required configuration steps.

Connectivity between Fabric Network Devices

In a typical VXLAN fabric, leaf nodes connect to all the spine nodes, but no leaf-to-leaf or spine-to-spine connections are usually required. The only exception is the connection (vPC peer link) required between two leaf nodes configured as part of a common vPC domain.

With the exception of the vPC peer link, all the connections between the fabric nodes are point-to-point routed links that can be configured as shown here:

```
interface Ethernet2/1
  description Layer 3 Link to Spine 1
  no switchport
  ip address 192.168.11.1/30
  no shutdown
```

Note: Because only two IP addresses are needed for those Layer 3 point-to-point connections, a /30 or even a /31 network mask can be configured to optimize the use of the IP address space. Starting with NX-OS Release 7.0(3)I3(1), IP unnumbered interfaces are also supported, to simplify the operational aspects of connecting leaf and spine nodes.

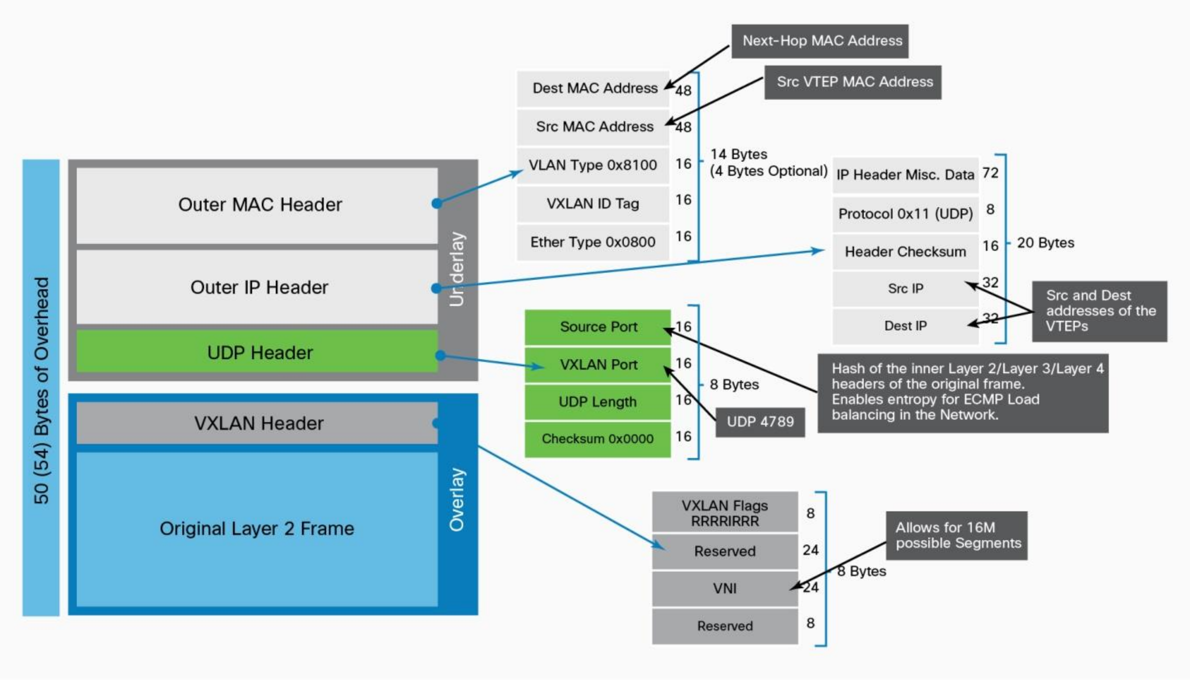
Note that when separate point-to-point links are available between the same pair of leaf and spine devices, you should first consider scaling out the spine before you consider using a Layer 3 port channel.

In the specific VXLAN multipod scenario, additional Layer 3 point-to-point links are required between the interconnected pods. Note that the main function of the transit leaf nodes in this design is to carry east-west VXLAN encapsulated traffic between leaf nodes deployed in separate pods. The example in Figure 6 shows a port-channel connection between the two transit leaf nodes (a vPC peer link), because the transit leaf nodes were also used as regular computing leaf nodes.

Maximum Transmission Unit Size in the Underlay Network

VXLAN is a MAC address in User Datagram Protocol (MAC-in-UDP) encapsulation technology. Ethernet frames originating from endpoints connected to the fabric are encapsulated in a UDP frame to be carried across the network and in this way provide Layer 2 and Layer 3 endpoint communication (Figure 7).

Figure 7. VXLAN Packet Format



To help ensure that VXLAN encapsulated packets can be successfully carried across the fabric, the increased maximum transmission unit (MTU) size must be configured on all the Layer 3 interfaces connecting the fabric nodes, including the interpod connections.

For VXLAN, this configuration requires increasing the MTU by at least 50 bytes (54 bytes if an IEEE 802.1Q header is present in the encapsulated frame) in the transport network. Therefore, when the overlay network requires the use of a 1500-byte MTU, the transport network needs to be configured to accommodate at least 1550-byte (or 1554-byte) packets. Jumbo-frame support in the transport network is strongly recommended if the overlay applications tend to use frame sizes larger than 1500 bytes.

The sample here shows how to increase the supported MTU size on a Layer 3 interface of a Cisco Nexus device:

```
interface Ethernet2/1
  description Layer 3 Link to Spine 1
  mtu 9216
```

Note: The same configuration is required for Layer 3 switch virtual interfaces (SVIs), if those are used to establish a point-to-point Layer 3 connection between fabric nodes (that is, connection between two vPC peers). Increased MTU size on edge Layer 2 interfaces and Layer 3 SVIs associated with the endpoint IP subnets is required only to support packets with larger frame sizes generated directly by the endpoint. (You can use the default value if the endpoint interfaces are also configured with the default size of 1500 bytes.)

Defining the VXLAN Tunnel Endpoint

All the leaf nodes shown previously in Figure 4 are labeled as VTEPs, because they perform the VTEP functions. A VTEP is the network device that performs the following two tasks:

- Receives traffic from locally connected endpoints and encapsulates it into VXLAN packets destined for remote VTEP nodes
- Receives VXLAN traffic originating from remote VTEP nodes, decapsulates it, and forwards it to locally connected endpoints

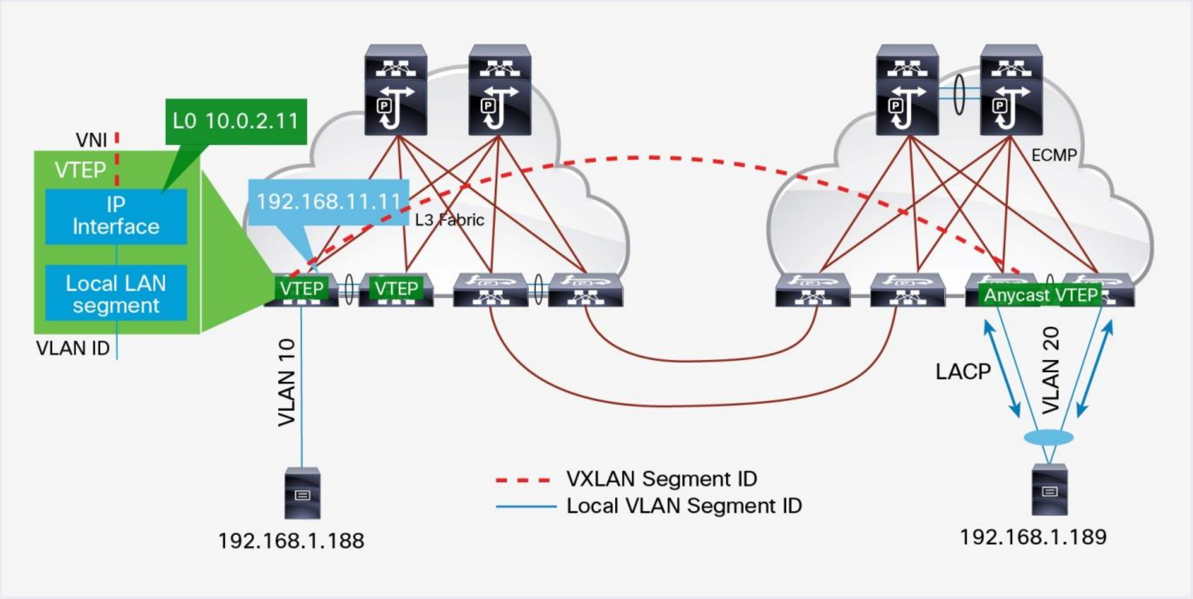
Each VTEP is functionally connected to the classic Layer 2 segment at which the endpoints are deployed, and to the routed underlay Layer 3 network to exchange VXLAN traffic with other remote VTEPs.

The VTEP dynamically learns the destination information for VXLAN encapsulated traffic for remote endpoints connected to the fabric by using a control protocol, as discussed in greater detail in the section “Overlay Network Deployment Considerations.”

Using the received EVPN information used to build information in the local forwarding tables, the ingress VTEP encapsulates the original Ethernet traffic with a VXLAN header (plus additional Layer 3 and Layer 4 external headers) and sends it over a Layer 3 network to the target egress VTEP. This VTEP then decapsulates it to present the original Layer 2 packet to its final destination endpoint. The VTEP has a unique IP address that identifies each device on the transport IP network. This IP address is used to source VXLAN encapsulated Layer 2 frames over the IP transport network.

For a data-plane perspective, the VXLAN multipod system behaves as a single VXLAN fabric network. As shown in Figure 8, VXLAN encapsulation is performed end to end across pods. The spines and the transit leaf switches perform only Layer 3 routing of VXLAN encapsulated frames to help ensure proper delivery to the destination VTEP.

Figure 8. VTEP-to-VTEP VXLAN Encapsulation



Underlay Control-Plane Deployment

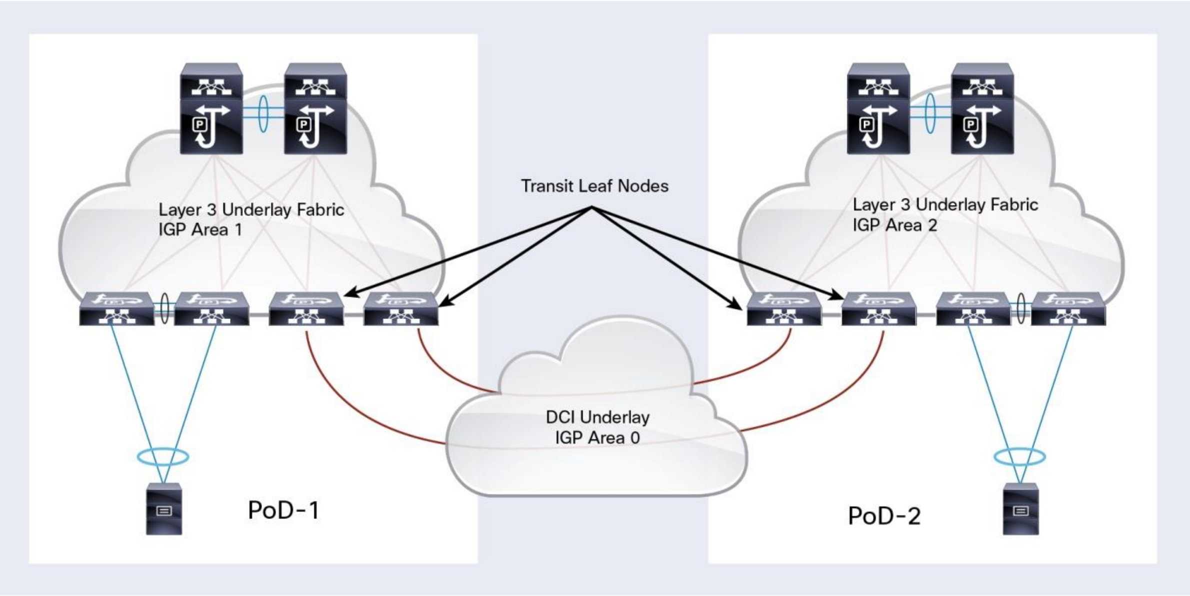
A main goal of the underlay control plane is to exchange IP reachability information for the VTEP addresses of all the leaf nodes connected to the VXLAN fabric. This exchange helps ensure that when a VTEP leaf node performs VXLAN encapsulation for locally originated frames, traffic can be routed successfully across the fabric.

Figure 9 shows the structured OSPF design proposed here for the deployment of the underlay control plane in a VXLAN multipod fabric. Each pod represents a different OSPF area, with the transit leaf nodes playing the role of area border routers (ABRs). The Layer 3 links (or generic Layer 3 network) interconnecting the pods are part of area 0.

For the best separation of the individual OSPF areas, you should use a different routing protocol between the pods. In the case of IGP separation, the OSPF ABRs become autonomous system border routers (ASBRs) through redistribution, for example, to BGP. In this way, OSPF provides the underlay information to BGP, and the flooding of OSPF link-state advertisements (LSAs) is contained within a single pod. The result is no OSPF-related changes to any remote pods, because the LSA flooding and related shortest-path first (SPF) calculation are limited to the local pod. Nevertheless, in the event of underlay IP changes (add or delete actions), pod local information will be propagated through redistribution and BGP peering.

This approach allows the removal of the VTEP IP address state from the transit segment. Pod local prefixes are exchanged through multihop BGP to remote pods without the need to know the individual VTEP IP addresses in the transit network. A summarization for the transit network provides additional masking for the pod-related underlay routing details. This approach requires the use of OSPF-to-BGP redistribution, but the BGP peering used for the overlay (EVPN) can easily be extended with the IPv4 address family.

Figure 9. Definition of OSPF Areas for Underlay Fabric Connectivity



Depending on the deployment scenario, the topology, and the number of devices and subnets, additional OSPF configuration may be needed to improve convergence. OSPFv2 includes a number of timers that control the behavior of protocol messages and SPF calculations. OSPFv2 includes the following optional timer parameters:

- LSA arrival time: This parameter sets the minimum interval allowed between LSAs that arrive from a neighbor. LSAs that arrive more quickly are dropped.

  timers lsa-arrival *msec*

- Throttle LSAs: This parameter sets the rate limit for LSA generation. This timer controls the frequency with which LSAs are generated after a topology change occurs.

  timers throttle lsa *start-time hold-interval max-time*

- Throttle SPF calculation: This parameter controls the frequency with which the SPF calculation is run.

  timers throttle spf *delay-time hold-time max-time*

The SPF throttle behavior uses an initial timer that waits for LSAs to arrive or exit. This interval should be long enough to let the network stabilize and short enough to achieve the expected convergence. In the case of continuous instability caused by network events, the timer is increased exponentially using a multiplier of the hold time until the maximum time is reached. For example, an initial delay of 100 ms could be chosen followed with a hold time of 500 ms and a maximum time of 5 seconds; this combination can help ensure stability.

Note: Each specific deployment scenario may require its own tuning. You thus should take into considerations all the parameters related to your specific deployment prior to changing any timer.

The OSPF configuration shown here applies to all the devices (leaf and spine nodes), in this specific case, the ones deployed in Pod-1.

```
interface Ethernet1/1
  description L3 Link to the Spine Node 1
  ip address 192.168.14.10/24
  ip ospf network point-to-point
  ip router ospf intra-fabric-routing area 0.0.0.1
  ip pim sparse-mode
  no shutdown
!
router ospf intra-fabric-routing
  timers throttle spf 100 500 5000
  timers lsa-arrival 100
  timers throttle lsa 100 500 5000
```

Each network infrastructure may require different values, depending on the number of networks and hosts. You should tune these timers after you validate the impact on the specific environment with the default timers.

Underlay Multicast Configuration

One task of the underlay network is to transport Layer 2 multidestination traffic between endpoints connected to the same logical Layer 2 broadcast domain in the overlay network. This type of traffic consists of Layer 2 broadcast, unknown unicast, and multicast traffic.

Two approaches can be used to allow transmission of broadcast, unknown unicast, and multicast traffic across the VXLAN fabric:

- Use multicast deployment in the underlay network to use the replication capabilities of the fabric spines to deliver traffic to all the edge VTEP devices.
- In scenarios in which multicast can't be deployed or is not desirable, you can use the source-replication capabilities of the VTEP nodes to create multiple unicast copies of the broadcast, unknown unicast, and multicast frames to be sent to each remote VTEP device.

Note that the choice you make for the replication process affects the entire multipod deployment. The use of ingress replication will prevent the application of storm control for loop mitigation across the whole fabric.

A commonly valid assumption for VXLAN multipod designs is that the customer owns the entire network infrastructure across which the fabric is stretched (often these are DWDM circuits). As a consequence, deploying multicast in the underlay is generally possible, and it is recommended to reduce the replication overhead of the VTEP nodes and to reduce bandwidth use. Reduced bandwidth use is particularly important in a geographically dispersed multipod scenario because less bandwidth is generally available across pods.

The following sections discuss the recommended configuration for enabling multicast in the underlay network.

Deploying Rendezvous Points Across Pods

When using multicast in the underlay network, the type of Protocol-Independent Multicast (PIM) commonly supported for VXLAN EVPN across all the Cisco Nexus platforms is PIM Sparse Mode (PIM-SM). This PIM type requires deployment of rendezvous points (RPs) across pods.

Note: At the time of this writing, PIM-SM is the only multicast type supported in the underlay network for VXLAN EVPN deployments that use the Cisco Nexus 9000 Series.

Several methods are available to achieve a highly available rendezvous-point deployment, including, for example, the use of protocols such as autorendezvous point and bootstrap. However, to improve convergence in a rendezvous-point failure scenario, the recommended approach is to deploy an anycast rendezvous point, which consists of using a common IP address on different devices to identify the rendezvous point. Simple static rendezvous-point mapping configuration is then applied to each node in the fabric to associate multicast groups with the rendezvous points, so that each source or receiver can then use the local rendezvous point that is closest from a topological point of view.

Note: Remember that the VTEP nodes represent the sources and destinations of the multicast traffic used to carry broadcast, unknown unicast, and multicast traffic between endpoints connected to those devices.

In a VXLAN multipod design, the recommended approach is to deploy a different set of rendezvous points in each connected room and site. Normally, the rendezvous points are deployed on the spine nodes, given the central position those devices play in the fabric.

When deploying an anycast rendezvous point, be sure to synchronize information between the different rendezvous points deployed in the network, because sources and receivers may join different rendezvous points, depending on where they are connected in the network. This synchronization is especially relevant for a multipod design, because the VTEPs that source and receive multicast traffic may be deployed in different pods.

Cisco Nexus platforms support several alternative mechanisms to synchronize (S, G) information between rendezvous points:

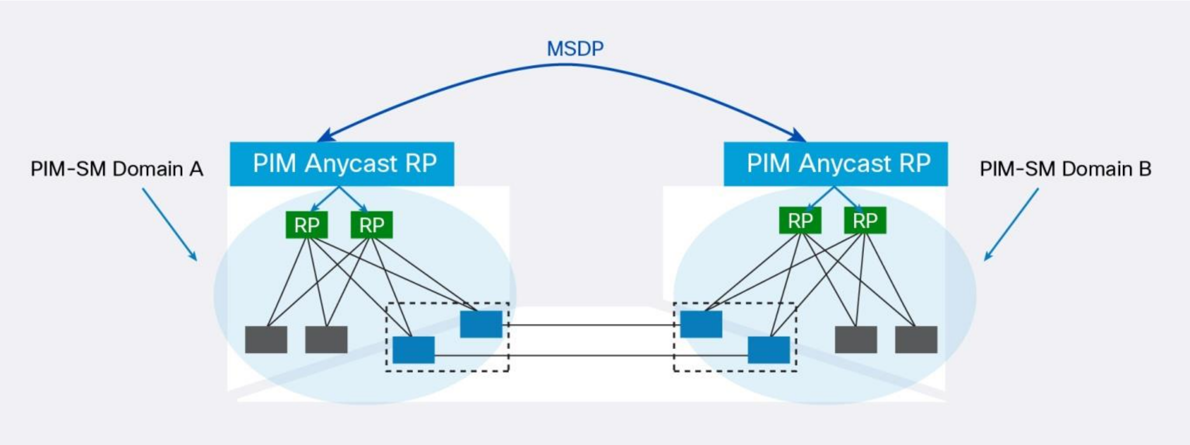
- Multicast Source Discovery Protocol (MSDP): This option has been available for a long time, and it is widely available across different switches and routers. MSDP sessions are established between rendezvous-point devices to exchange information across pods about sources and receivers for each given multicast group.
- PIM with anycast rendezvous points: This option is currently supported only on Cisco Nexus platforms and uses PIM as the control plane to synchronize the (S, G) state between rendezvous points.

For the VXLAN multipod design, the goal is to try to isolate as much as possible the control planes used in the different pods. As a consequence, the validated approach uses both mechanisms described here: PIM between the rendezvous points defined within each room and site, and MSDP between rendezvous points in separate pods. This approach also follows the recommendation to separate the IGP domains of each pod and interconnect them with BGP. In cases in which an IGP mechanism is used within and between pods, PIM-based anycast rendezvous points in each pod suffice.

Note: This model is documented here because it requires additional configuration steps. Use of PIM with anycast rendezvous points across pods is also a viable option if you want to simplify the configuration.

As shown in Figure 10, a pair of anycast rendezvous points is defined in each pod, locally using PIM to exchange (S, G) information. Each pair is in a separate PIM-SM domain and has an MSDP peering relationship with the rendezvous points deployed in a separate PIM domain. The MSDP connections between rendezvous points are established across the underlying routed infrastructure, and the receiving rendezvous points use the source lists to establish a source path.

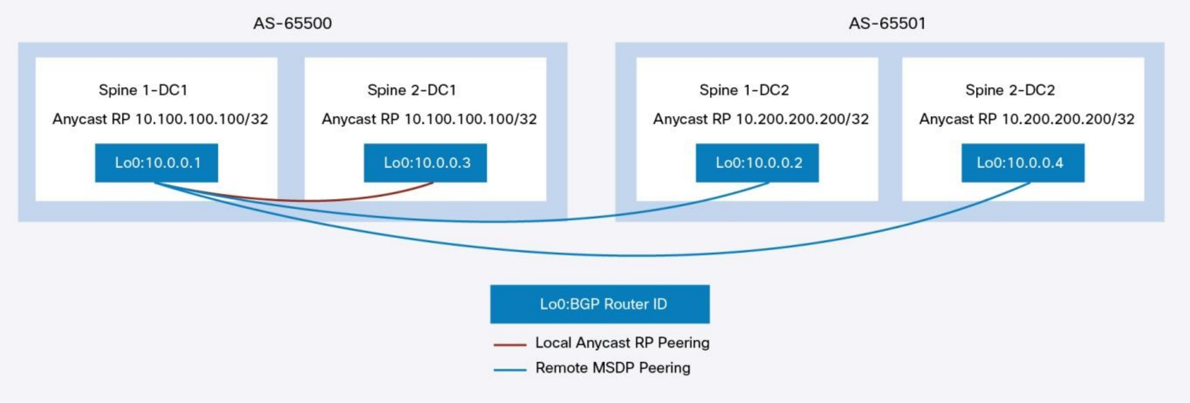
Figure 10. Anycast Rendezvous-Point PIM and MSDP in a Multipod Design



Note: The use of MSDP enforces the sturdiness of the whole fabric by separating each PIM-SM domain. Alternatively, you can use the same mechanism (PIM or MSDP) between all the rendezvous points deployed across pods

The code sample that follows shows the multicast configuration required on the spine devices (a similar configuration must be replicated on all the spines that act as rendezvous points). Figure 11 shows a logical view of the configuration.

Figure 11. Logical View of Anycast Rendezvous Point and MSDP Peering



```
feature pim
feature msdp
!
interface loopback0
  description MSDP/BGP peering
  ip address 10.0.0.1/32
  ip router ospf intra-fabric-routing area 0.0.0.1
!
interface loopback2
  description Anycast RP
  ip address 10.100.100.100/32
  ip router ospf intra-fabric-routing area 0.0.0.1
  ip pim sparse-mode
!
ip pim rp-address 10.100.100.100 group-list 239.0.0.0/8
ip pim anycast-rp 10.100.100.100 10.0.0.1
ip pim anycast-rp 10.100.100.100 10.0.0.3
!
ip msdp originator-id loopback0
ip msdp peer 10.0.0.2 connect-source loopback0 remote-as 65501
ip msdp peer 10.0.0.4 connect-source loopback0 remote-as 65501
!
interface Ethernet1/1
  ip pim sparse-mode
interface Ethernet1/2
  ip pim sparse-mode
<snip>                 ← Repeat for all Layer 3 links connecting to leaf nodes
```

- The **loopback2** interface is the anycast rendezvous-point interface that is commonly defined on all the spine devices (across pods).
- The **loopback0** interface is used for control-plane communication, including the establishment of the MSDP sessions. Every spine defines a unique interface for that.
- The **ip pim rp-address** command statically defines the mapping between the rendezvous point and the multicast groups. This command is required on all the nodes in the VXLAN fabric (leaf and spine nodes). Note that each pod uses a unique rendezvous-point address. In the simple example of a multipod design across two rooms or sites, 10.100.100.100 identifies the pair of anycast rendezvous-point devices in Pod-1, and 10.200.200.200 identifies the anycast rendezvous-point devices in Pod-2.
- The **ip pim anycast-rp** command specifies the rendezvous points between which PIM is used as the mechanism to synchronize (S, G) information (that is, only the rendezvous points local to the pod).
- The **ip msdp originator-id** command is required when deploying the same anycast rendezvous-point address on different devices, to help ensure a successful exchange of MSDP information between them. (The originator ID should be an IP address that uniquely identifies each device.)
- The **ip msdp peer** command establishes the MSDP session with the peers. It is required to establish a full mesh of MSDP sessions between all the spine devices that are defined as rendezvous points. In a VXLAN multipod fabric, each rendezvous point thus peers with the other rendezvous points in the local pod and with all the rendezvous points deployed in the other pods.

Overlay Network Deployment Considerations

After the physical network infrastructure has been properly configured following the steps discussed in the preceding section, and after the pods are interconnected with each other (both for the underlay control plane and the data plane), you can continue with the deployment of overlay networks, supporting the multitenancy and endpoint mobility functions required in modern data center designs.

Deploying the Overlay Control Plane

The first required step is to deploy an overlay control plane, to help ensure successful exchange of reachability information for endpoints connected to the logical overlay networks.

As the overlay control plane, VXLAN uses an evolved version of BGP called Multiprotocol BGP, or MP-BGP, with a specific EVPN address family that allows the exchange of both Layer 2 and Layer 3 information. The use of MP-BGP EVPN enhances the original flood-and-learn VXLAN behavior.

Note: Although MP-iBGP and MP-eBGP can both be deployed as overlay protocols within a given VXLAN fabric, MP-iBGP deployments are common, and so this is the model presented in this document.

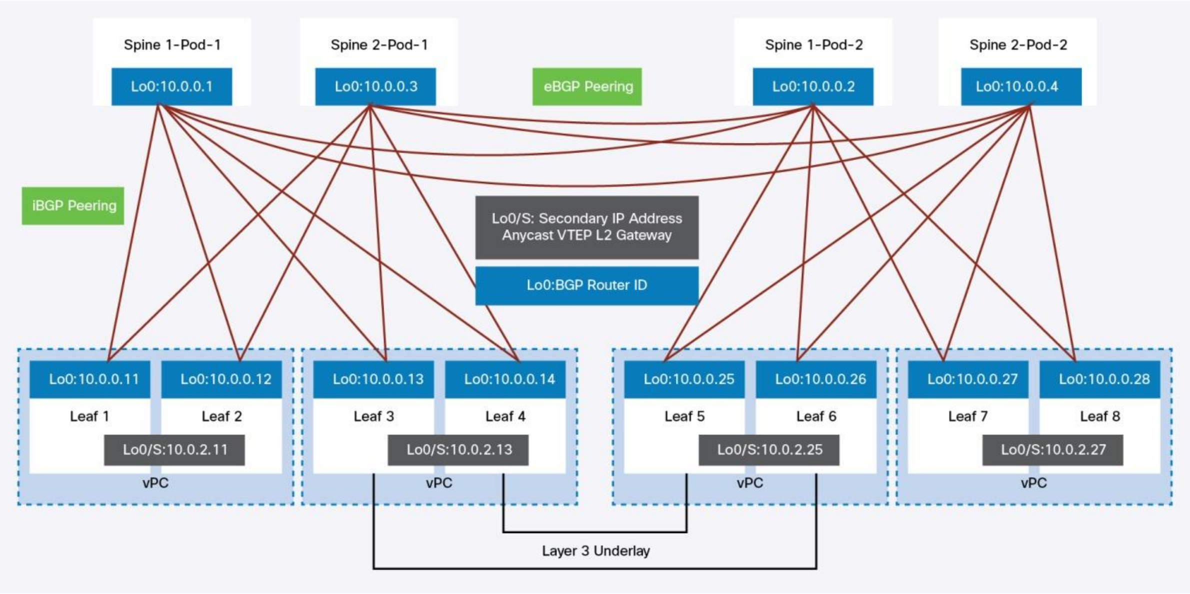
For the VXLAN multipod design discussed in this document, an independent MP-iBGP domain has been created within each VXLAN pod (spine and leaf nodes belong to the same autonomous system number (ASN) within the same data center). A full mesh of MP-iBGP sessions must be established between all the leaf nodes (deployed as VTEPs), so a best practice to simplify operations is to define a pair of MP-iBGP route reflectors (RRs) in each pod. The logical choice is to deploy them on the spine nodes.

Note: Deploying route reflectors on the spine nodes implies that the network devices used in that role must support the MP-BGP control plane. Support for VXLAN encapsulation and decapsulation hardware capabilities on the spine nodes is not required, because those devices usually simply route VXLAN data-plane traffic exchanged between leaf VTEPs. If support for MP-BGP control-plane traffic on the spine nodes is not possible, the route-reflector function can be moved to an alternative place in the network.

Endpoint reachability information must then be exchanged across separate pods to help ensure end-to-end communication. For the reasons discussed in the "VXLAN Multipod Design Overview" section, the deployment option presented here uses interpod MP-eBGP EVPN sessions. These MP-eBGP sessions can be established between the pairs of transit leaf nodes deployed within each pod, or they can be established between the route reflectors on the spine nodes. A good practice for the choice of interpod connectivity is to follow the physical connection. In multipod networks interconnected at the spine, the overlay and underlay MP-eBGP sessions can be established between spine nodes. In multipod networks interconnected at the leaf, the overlay and underlay MP-eBGP sessions can be established between leaf nodes.

To clearly separate the underlay and overlay routing peerings in this document, the design discussed here uses underlay peering at the leaf node and overlay peering at the spine node. This approach is discussed in this section and shown in Figure 12.

Figure 12. BGP Control-Plane Loopback Address Assignment



Note: Stretching the fabric across pods requires pure Layer 3 connectivity. However, in Figure 12, the vPC domain is used for local attached endpoints in conjunction with Layer 3 outside links.

MP-BGP EVPN can transport Layer 2 information such as MAC addresses as well as Layer 3 information such as host IP addresses (host routes) and IP subnets. For this purpose, it uses two forms of routing advertisement:

- Type 2
  - Used to announce host MAC and host IP address information for the endpoint directly connected to the VXLAN fabric
  - Extended community: Router MAC address (for Layer 3 virtual network identifier [VNI]), sequence number (for Layer 2 VNIs), and route-target (RT) value.

- Type 5
  - Advertises IP subnet prefixes or host routes (associated, for example, with locally defined loopback interfaces)
  - Extended community: Router MAC address to uniquely identify each VTEP node and route-target value.

In the VXLAN multipod deployment, Type 2 is used to distribute host reachability information (MAC and IP addresses) to all VTEPs that exist over the local and remote pods that share the same Layer 2 segments. Type 5 is relevant for all the IP prefixes that are redistributed in the EVPN control plane. These include external IP prefixes (advertised by border leaf nodes), IP subnets to which the workloads are connected (advertised by each computing VTEP), and loopback IP addresses (inside the tenant address space).

The following sample shows the configuration required on the fabric node acting as the route reflector (in the specific scenario discussed in this document, it is deployed on a spine node).

```
interface loopback0
  description BGP/OSPF Router-id
  ip address 10.0.0.1/32
  ip router ospf intra-fabric-routing area 0.0.0.1
  ip pim sparse-mode
!
route-map NH-Unchanged               ← Allows you to preserve next-hop
information
  set ip next-hop unchanged
!
router bgp 65500
  router-id 10.0.0.1
  log-neighbor-changes
  template peer LEAFS
    remote-as 65500
    update-source loopback0
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community both
      route-reflector-client
  neighbor 10.0.0.2 remote-as 65501  ← Remote BGP router ID in Pod-2 (Spine-1)
    update-source loopback0
    ebgp-multihop 10                       ← Allows multihop peering
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community both
      route-map NH-Unchanged out
  neighbor 10.0.0.4 remote-as 65501  ← Remote BGP router ID in Pod-2 (Spine-2)
    update-source loopback0
    ebgp-multihop 10
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community both
      route-map NH-Unchanged out
  neighbor 10.0.0.11                        ← Leaf-1 in Pod-1
    inherit peer LEAFS
  // snip//                                 ← Repeat for all local leaf switches
```

Note the following points about this configuration:

- A single **loopback0** interface is used for the BGP router ID as well as for establishing route-reflector peering sessions with local VTEPs and remote route reflectors.

Note: All control planes running in the fabric use the same loopback interface, Lo0, including IGP (OSPF), MP-BGP, PIM, and MSDP.

- A route map must be defined to help ensure that MP-BGP routing information is exchanged across pods while preserving the original next-hop information. This requirement allows any given VTEP to associate the received MAC and IP address information with a remote VTEP node, not with the spine node from which it receives the update.

Note: In the MP-iBGP deployment discussed in this document, the spine nodes perform only route-reflection tasks (and Layer 3 forwarding), and the VXLAN encapsulation is performed leaf node to leaf node.

- The design discussed in this document establishes eBGP sessions between loopback IP addresses of spine devices deployed in separate pods while the pods are physically interconnected through transit leaf nodes. Multihop eBGP must be configured (because several Layer 3 hops exist between the route-reflector devices).

Note: An alternative deployment model consists of establishing MP-eBGP EVPN adjacencies between the transit leaf nodes in different pods. In scenarios in which the transit leaf nodes are connected back to back through dark fiber or DWDM circuits, peering can be implemented directly through the physical interfaces, effectively eliminating the need to establish multihop eBGP sessions.

The configuration required to deploy the overlay control plane on the leaf nodes is much simpler and basically identical across all the nodes deployed in the same data center pod, as shown here:

```
interface loopback0
  description BGP/OSPF Router-id
  ip address 10.0.0.111/32
  ip router ospf intra-fabric-routing area 0.0.0.1
!
interface loopback1
  description Anycast VTEP address
  ip address 10.0.0.11/32
  ip address 10.0.2.11/32 secondary
  ip router ospf intra-fabric-routing area 0.0.0.1
  ip pim sparse-mode
!
router bgp 65500
  router-id 10.0.0.111
  neighbor 10.0.0.1 remote-as 65500        ←  Spine-1 Pod-1 (RR1)
    update-source loopback0
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community both
  neighbor 10.0.0.3 remote-as 65500        ←  Spine-2 Pod-1 (RR2)
    update-source loopback0
    address-family ipv4 unicast
    address-family l2vpn evpn
      send-community both
  vrf Tenant-1
    router-id 10.0.0.111
    address-family ipv4 unicast
      advertise l2vpn evpn
      redistribute direct route-map CONNECTED
    neighbor 10.10.10.1              ←  Per-tenant MP-iBGP session (VRF-Lite)
      remote-as 65500
      address-family ipv4 unicast
  vrf Tenant-2
    router-id 10.0.0.111
    address-family ipv4 unicast
      advertise l2vpn evpn
      redistribute direct route-map CONNECTED
    neighbor 10.10.10.1              ←  Per-tenant MP-iBGP session (VRF-Lite)
      remote-as 65500
      address-family ipv4 unicast
//snip//                                   ←  Repeat for each tenant
```

Note: On the leaf nodes, all control planes running in the fabric use the same loopback interface, lo0, including IGP (OSPF), MP-BGP, PIM, and MSDP. In MP-BGP, the next hop for the EVPN advertisement is instead a separate loopback interface (lo1) representing the VTEP IP address.

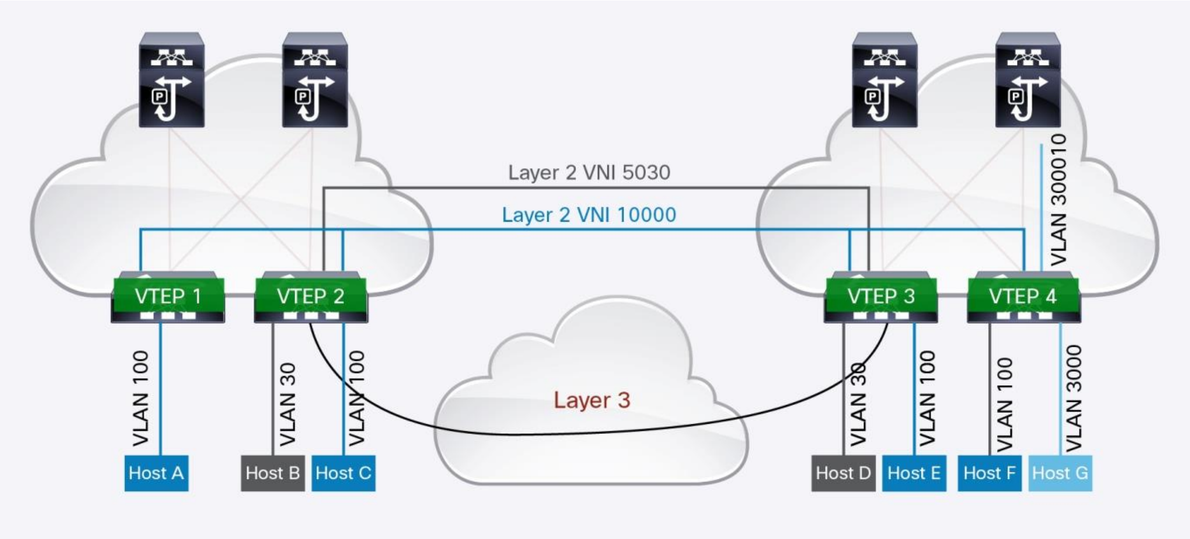
Layer 2 Logical Isolation (Layer 2 Virtual Network Identifiers)

VXLAN overlay networks provide the logical abstraction allowing endpoints connected to different leaf nodes separated by multiple Layer 3 fabric nodes to function as they were connected to the same Layer 2 segment. This logical Layer 2 segment is usually referred to as the Layer 2 VNI.

Note: This document uses the terms “Layer 2 VNI,” “VXLAN segment.” and “vn-segment” interchangeably.

The VXLAN segments are independent from the underlying network topology. Likewise, the underlying IP network between VTEPs is independent from the VXLAN overlay. The combination of locally defined VLANs and their mapping to associated Layer 2 VNIs allows you to create Layer 2 logical segments that can be extended across the pods, as shown in Figure 13.

Figure 13. VLAN-to-Layer 2 VNI Mapping



The creation of the VXLAN segments shown in Figure 13 provides Layer 2 connectivity between endpoints connected to the same VNI independent of the specific pod to which they are connected. At the same time, different Layer 2 VNI segments provide logical isolation between the connected endpoints. As in traditional VLAN deployments, communication between endpoints belonging to separate Layer 2 VNIs is possible only through a Layer 3 routing function, as will be discussed in the section “Layer 3 Multitenancy (Virtual Routing and Forwarding and Layer 3 VNIs).”

The sample here shows the creation of VLAN-to-VNI mappings on a VTEP device, which is usually a leaf node:

```
vlan 10
  vn-segment 10000
vlan 20
  vn-segment 20000
```

After the VLAN-to-VNI mappings have been defined, you then must associate those created Layer 2 VNIs with an network virtualization edge (NVE) logical interface, as shown in the following configuration sample:

```
interface nve1
  no shutdown
  source-interface loopback1          ← Specify the VTEP IP address
  host-reachability protocol bgp      ← Use the MP-BGP control plane
  member vni 10000
```

```
    suppress-arp                   ← Enable ARP suppression
    mcast-group 239.1.1.1          ← Associate a multicast group for broadcast,
unknown unicast, and multicast traffic
  member vni 20000
    mcast-group 239.1.1.2
//snip//                           ← Repeat for each locally defined Layer 2 VNI
```

The final configuration step in the creation of Layer 2 VNIs is shown in the following sample:

```
evpn
  vni 5000 l2
    rd auto
    route-target import 100:50             ← Please read note underneath
    route-target export 100:50
  … snip …
  vni 10000 l2
    rd auto
    route-target import 100:100
    route-target export 100:100
  vni 20000 l2
    rd auto
    route-target import 200:200
    route-target export 200:200
  vni 300000 l2
    rd auto
    route-target import 200:3000
    route-target export 200:3000
```

- The route-distinguisher and route-target configurations shown here apply only to the Layer 2 (MAC address) information exchanged between the VTEP devices through the EVPN control plane (Type 2 updates).
- Each routing update sent by a specific VTEP to the other VTEPs has an associated route-target value (an **export** configuration). The receiving VTEPs import the routing updates in the Layer 2 routing information base (RIB) table only when at least one local Layer 2 VNI matches the **import** value.
- The same considerations do not apply to the route distinguisher value, because it is always uniquely generated for each VTEP device.

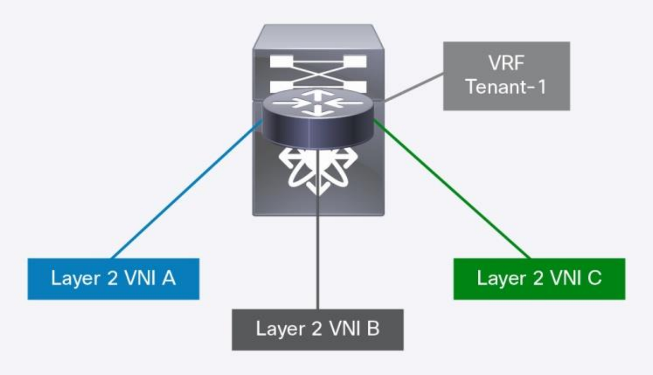
Important: In the multipod solution presented in this document, the route-target values must be manually defined to match across VTEPs deployed in different pods in different BGP autonomous systems. The autogenerated values for route-target auto are dependent on the BGP autonomous system because the values use the “ASN:VNI” format. With different BGP autonomous systems, the autogenerated route target would result in different route-target values for the same Layer 2 VNI, and the import would fail.

Layer 3 Multitenancy (Virtual Routing and Forwarding and Layer 3 VNIs)

The logical Layer 2 segment created by mapping a locally significant VLAN to a globally significant Layer 2 VNI is normally associated with an IP subnet. When endpoints connected to the Layer 2 VNI need to communicate with endpoints belonging to different IP subnets, they send the traffic to their default gateway. Deploying VXLAN EVPN allows support for a distributed default gateway on each leaf node, a deployment model commonly referred to as an anycast gateway.

In a VXLAN deployment, the various Layer 2 segments defined by combining local VLANs and Layer 2 VNIs (as discussed in the previous section) can be associated with a Virtual Routing and Forwarding (VRF) instance if they need to communicate, as shown in the logical diagram in Figure 14.

Figure 14. VRF and Associated Layer 2 VNIs



Local endpoints connected to different Layer 2 VNIs can communicate through normal Layer 3 routing in the context of the VRF (that is, VXLAN encapsulation is not required). The additional required configuration for each defined VRF instance is shown here:

```
vlan 3001
  name L3_Tenant1
  vn-segment 300001
!
vrf context Tenant-1
  vni 300001                                             ← Define the Layer 3 VNI
  rd auto
  address-family ipv4 unicast
    route-target import 100:300001          ← Optional for intrafabric VXLAN
    route-target import 100:300001 evpn
    route-target export 100:300001          ← Optional for intrafabric VXLAN
    route-target export 100:300001 evpn
!
interface Vlan3001
  description L3_Tenant1
  no shutdown
```

```
    mtu 9216
    vrf member Tenant-1
    no ip redirects
    no ipv6 redirects
    ip forward
  !
  interface nve1
    member vni 300001 associate-vrf           ← Bind the Layer 3 VNI to the NVE
```

As for Layer 2 VNIs, in a multipod design with a separate BGP autonomous system in each pod, the route-target values must be manually defined (for example, 100:300001 is used for Tenant 1).

Important: In the multipod solution presented in this document, the route-target values must be manually defined to match across VTEPs deployed in different pods in different BGP autonomous systems. Because the autogenerated values for **route-target auto** are dependent on the BGP autonomous system because the values use the “ASN:VNI” format. With different BGP autonomous systems, the autogenerated route target would result in different route-target values for the same Layer 2 VNI, and the import would fail.

Layer 3 Communication and Mobility Considerations

One requirement for an efficient VXLAN multipod deployment is a distributed Layer 3 default gateway at all the pod locations to reduce traffic hair-pinning when endpoints migrate across pods.

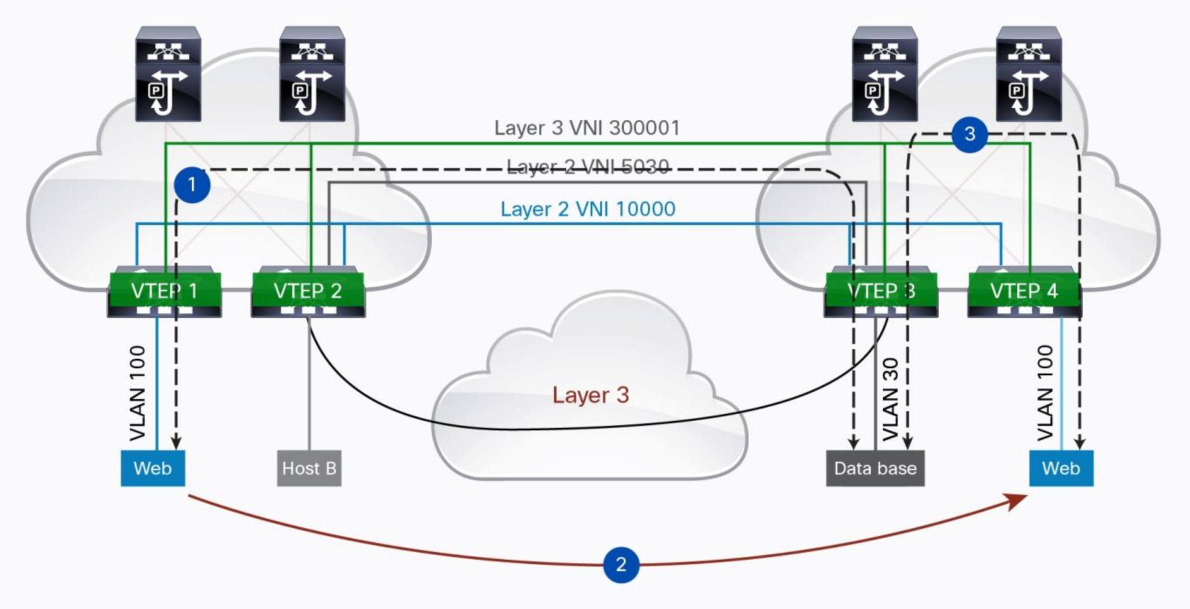
When a virtual machine or the whole multitier application implements a hot live migration to a different pod, the host must continue to process communications without any interruption while using its default gateway locally. Without a local default gateway, the performance of multitier applications (relying on east-west communication patterns) may suffer from the hair-pinning of traffic destined for the default gateway in the original pod.

This requirement is especially critical when the pods are deployed across separate physical locations: in this scenario, back-and-forth traffic hair-pinning may affect the application’s behavior.

The anycast Layer 3 gateway function is natively embedded in the VXLAN EVPN control plane distributed to all VTEPs. Consequently, a fully deployed VXLAN EVPN domain fabric can support as many anycast Layer 3 gateways as there are top-of-rack switches. All leaf nodes are configured with the same default gateway IP address for a specific subnet as well as the same virtual MAC (vMAC) address (used for all the defined IP subnets).

Figure 15 shows what happens when a web server, communicating with the end user (not represented here) and its database (1), performs a live migration (2) across the multipod fabric to the remote location where its database server resides.

Figure 15. Anycast Layer 3 Gateway



The MP-BGP EVPN control plane notices the movement and the new location of the web server. Subsequently, it increases the BGP sequence number value for that particular endpoint and immediately notifies all VTEPs of the new location. The sequence number is now higher than the original value, so all VTEPs that receive the update and in which the web server's Layer 2 VNI is locally defined update their forwarding tables accordingly with the new next-hop information (egress VTEP 4).

Without any interruption of the current active sessions, the web server continues to transparently use its default gateway, represented now by VTEP 4, which is locally available at its new location. The routed east-west communication between the web and the database occurs locally (3) in Pod-2 after the server migration (using routed traffic optimization).

Network Services Integration in VXLAN Multipod Design

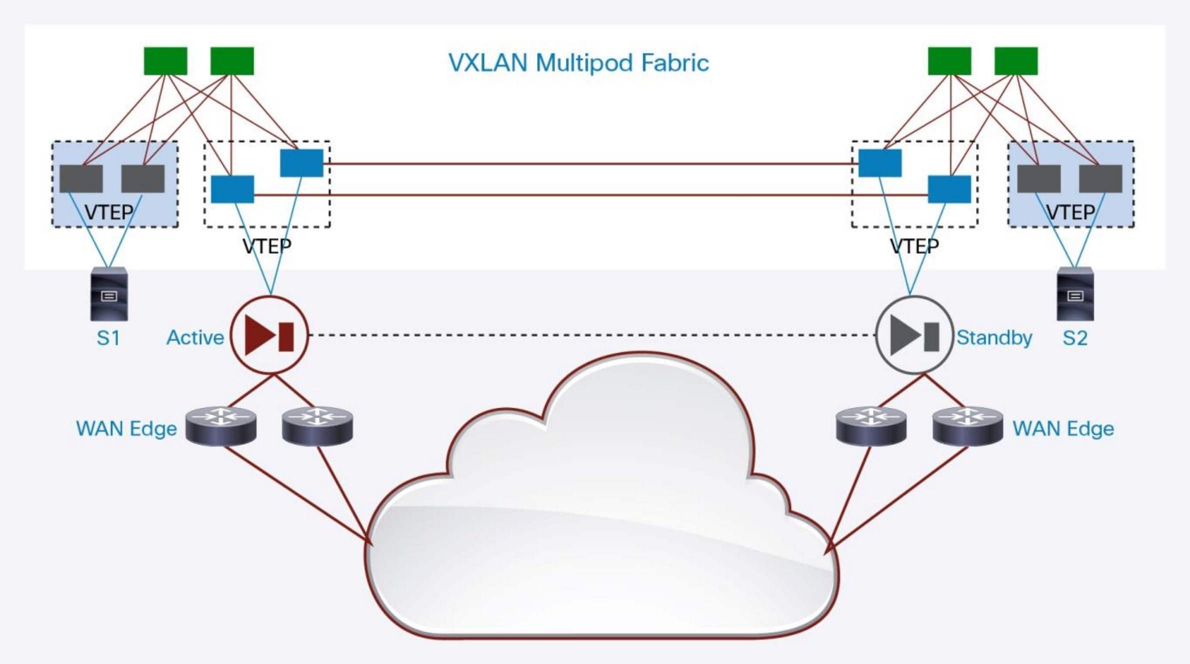
The previous section discussed how Layer 2 and Layer 3 communication is handled within the VXLAN multipod fabric. This section discusses how to integrate network services. The discussion here focuses on the specific example of firewall integration, but similar considerations can be applied to other network services (for example, load balancers).

A typical requirement when integrating network services into a VXLAN multipod design is verifying that the various service nodes are physically connected to different pods to help ensure that the service remains available even if an entire pod fails or becomes isolated from the rest of the network. As will be clarified in the following discussion, this requirement entails some particular design considerations that differ from those for a more traditional single-pod scenario in which all the service nodes are usually connected to the same pair of VTEP devices (usually called service leaf nodes).

Network Services in Active-Standby Mode

The first deployment model consists of a pair of devices (physical or virtual) connected in active-standby mode in two separate pods (Figure 16). A typical use case is the deployment of a perimeter firewall enforcing security policies for traffic entering and leaving the multipod fabric.

Figure 16. Active-Standby Edge Firewall Deployment



Initial design considerations include the following:

- The firewall service nodes can be physical or virtual appliances. Physical appliances are usually deployed to enforce perimeter security.
- The firewall service nodes can be deployed in routed or transparent mode. Routed mode is the most commonly used option.
- The firewall service nodes can also be deployed as first-hop routers, offering the function of a default gateway (this capability is usually determined by the security rules and policies of the individual enterprise). In this case, the VXLAN EVPN fabric performs only Layer 2 functions for all the endpoints whose gateway is deployed on the firewall service node. This specific scenario is not discussed in this document.
- When using routed mode, you can use a dynamic protocol between the firewall nodes and the network devices to exchange routes with the VXLAN fabric and the WAN edge devices. Alternatively, you can use static routing, commonly used in deployments.
- To support multitenancy, different firewall interfaces (or virtual contexts) can be dedicated to each tenant that needs to communicate with the external network domain. This approach is normally used in conjunction with the deployment of physical firewall appliances. Alternatively, a dedicated pair of active-standby virtual appliances can be deployed for each tenant.

In the specific design discussed in this document, firewalls are deployed in routed mode, and the static routing scenario is used because it involves several important design aspects that need to be considered when integrating a multipod solution.

Note: This document focuses on the deployment of a perimeter firewall to enforce security policies across tenants and between each tenant and the external world. Firewall service nodes are not considered for communication between IP subnets that belong to a given tenant and that are internal to the VXLAN fabric.

Static Routing Between Firewalls and Network Devices

As previously mentioned, static routing between the service appliances and the network is common in real-life deployments. This choice is usually based on capital expenditure considerations (dynamic routing often triggers additional licensing costs for the service appliances). It is also based on security and operational considerations, because this option is perceived to more tightly control the information that can be exchanged with the network. Additionally, not every firewall device offered by vendors is capable of performing dynamic routing.

The following sections describe the steps required to deploy a static routing solution.

Interconnecting the Firewalls to the VXLAN Multipod Fabric

The first step is to connect the perimeter firewall devices to the separate pods. The main function of these firewalls is to secure all the communication between the fabric and the external network domain, and to maintain this function even after the total failure of a specific pod. This capability is especially relevant when you are deploying a multipod solution across geographically dispersed pods.

Static routing provides multiple options for interconnecting the firewall devices to the fabric switches, including the use of a port channel, as shown in Figure 17.

Figure 17. Firewall-to-VXLAN Multipod Fabric Connectivity (Physical View)

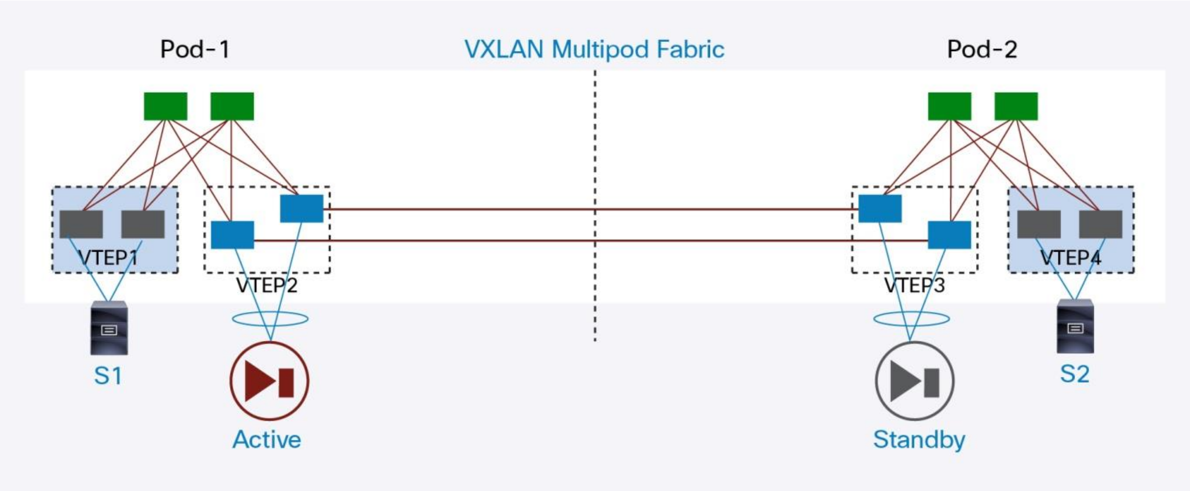
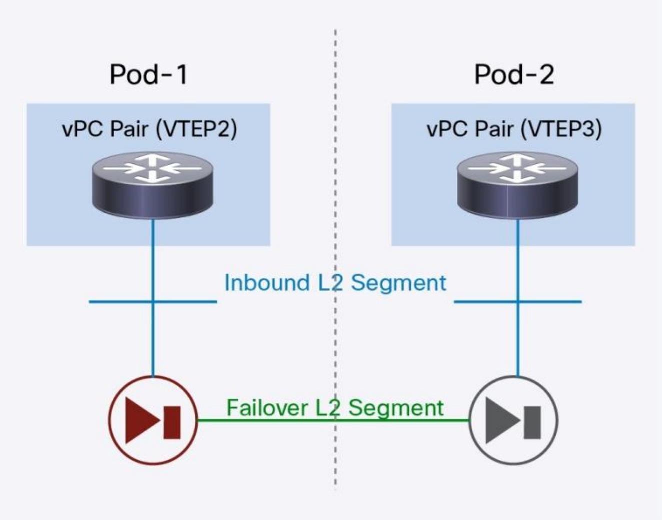


Figure highlights shows how a port channel is used to interconnect each firewall appliance with a separate pair of VXLAN leaf nodes (deployed in separate pods), using vPC on the fabric leaf layer to establish active-active communication with the service node. The use of vPC for the individual firewalls is a design option and not a requirement. vPC can provide connectivity resiliency, but it also introduces additional complexity, especially with firewall attachment in Layer 3 mode. vPC is neither required nor needed for firewall insertion.

The logical view depicted in Figure 18 provides additional insight.

Figure 18. Firewall-to-VXLAN Fabric Connectivity (Logical View)



Some important design considerations for the described deployment model are listed here:

- VTEP2 and VTEP3 each represent two pairs of physical leaf nodes that are part of the same vPC domain and deployed in separate pods.

Note: For more information about the interaction between Cisco vPC technology and VXLAN EVPN, please refer to http://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/guide-c07-734107.html.

- The failover Layer 2 segment is used between the active and standby firewall devices to synchronize state information and exchange keepalive messages. Because the two firewalls are deployed in separate pods, a dedicated VXLAN segment (Layer 2 VNI) in the multipod fabric must be used to extend this logical Layer 2 connectivity between them.

Note: This deployment option has been successfully validated with the Cisco Adaptive Security Appliances (ASA) firewall. You should be sure to perform the same validation for other Cisco or third-party service appliances before deploying them in a production environment.

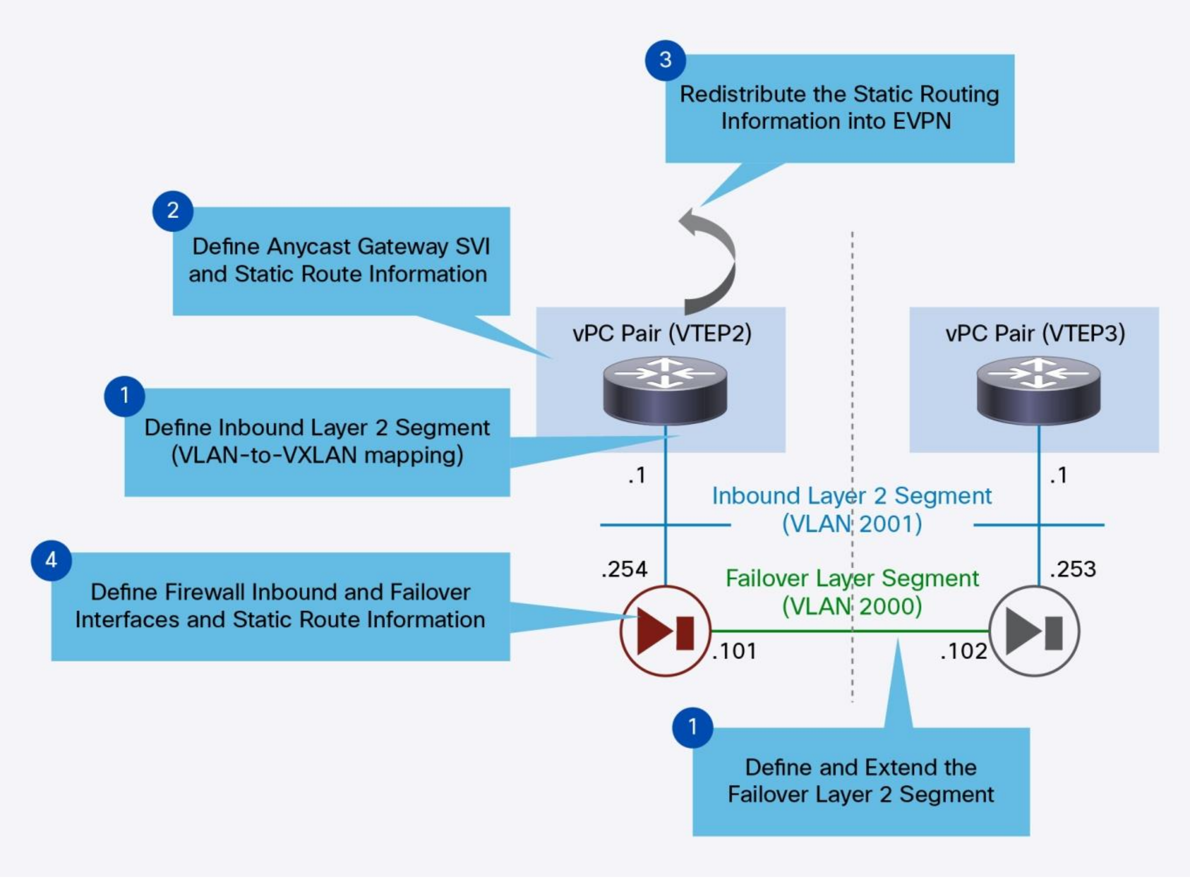
- An inbound Layer 2 segment is used to route traffic between the fabric and the inside interface of the firewall (and vice versa). The VXLAN anycast gateway function is used to help ensure that a consistent Layer 3 next hop is offered to the firewall toward the internal IP subnets in the VXLAN fabric independent from the location (the specific VTEP) to which the active firewall appliance is connected.

- As shown in Figure 18, the inbound Layer 2 segment is not stretched between the two pods. This design assumes that no endpoints are connected to that Layer 2 segment and that the firewall is used as the default gateway (that is, communication between any endpoint and the firewall is always through a routed hop represented by the VXLAN fabric). If that were not the case, and if the endpoints were directly connected to the inbound Layer 2 segment, then the recommended approach would be to map the segment to a Layer 2 VNI and extend it across the pods.

Figure 19 shows the specific configuration steps required to establish connectivity between the firewalls and the VXLAN fabric for a given tenant. This configuration must be applied to the devices (VXLAN leaf nodes, WAN edge routers, and firewalls) deployed in both pods.

Note: These steps assume that the other aspects of the configuration are already complete (vPC domain creation, physical connectivity of the firewall nodes to the VXLAN leaf nodes, etc.).

Figure 19. Configuring Connectivity Between Firewalls and VXLAN Multipod Fabric


3
Redistribute the Static Routing Information into EVPN
2
Define Anycast Gateway SVI and Static Route Information
vPC Pair (VTEP2)
vPC Pair (VTEP3)
1
Define Inbound Layer 2 Segment (VLAN-to-VXLAN mapping)
.1
.1
Inbound Layer 2 Segment (VLAN 2001)
4
Define Firewall Inbound and Failover Interfaces and Static Route Information
.254
.253
Failover Layer Segment (VLAN 2000)
.101
.102
1
Define and Extend the Failover Layer 2 Segment

1. On each VXLAN leaf node (part of logical vPC pairs VTEP2 and VTEP3), define the Layer 2 segment used to connect the inside interface of the firewall to the VXLAN leaf nodes. Perform the same operation for the Layer 2 segment used to exchange state information and keepalive messages between the active and standby firewall nodes. As mentioned, this segment must also be extended across pods.

- Define the Layer 2 VLANs and map them to the corresponding Layer 2 VNIs:

```
vlan 2000
  name ASAv-HA
  vn-segment 200000
vlan 2001
  name ASAv-Inbound
  vn-segment 200001
```

- Associate the failover Layer 2 VNI with the NVE interface and with the EVPN portion of the configuration:

```
interface nve1
  no shutdown
  source-interface loopback0
  host-reachability protocol bgp
  member vni 200000
    mcast-group 239.1.1.2
!
evpn
  vni 200000 l2
    rd auto
    route-target import 100:2000
    route-target export 100:2000
```

2. On each VXLAN leaf node, define the Layer 3 SVI associated with the inbound Layer 2 segment, which represents the Layer 3 next hop for the firewall to the fabric internal subnets. Also configure the static route to allow the fabric to use the firewall as the Layer 3 next hop to reach the external destination. Use of a default route is usually the simplest way to achieve this result.

   This configuration must be duplicated on both pairs of leaf nodes that reside in each pod:

```
interface Vlan2001
  description ASAv-Inbound
  no shutdown
  vrf member Tenant-1
  ip address 20.1.1.1/24 tag 1234
  fabric forwarding mode anycast-gateway
!
vrf context Tenant-1
  ip route 0.0.0.0/0 20.1.1.254 tag 1234
```

Note: The anycast gateway is the only default gateway option for the SVI associated with a VXLAN extended VLAN (Layer 2 VNI). Other First-Hop Redundancy Protocol (FHRP) mechanisms such as Hot Standby Router Protocol (HSRP) and Virtual Router Redundancy Protocol (VRRP) are not supported.

3. On each VXLAN leaf node, redistribute the static route on the MP-BGP control plane:

```
route-map TAG-1234 permit 10
  match tag 1234
!
router bgp 1
  vrf Tenant-1
    router-id 10.0.0.13
    address-family ipv4 unicast
      advertise l2vpn evpn
      redistribute static route-map TAG-1234
      default-information originate
```

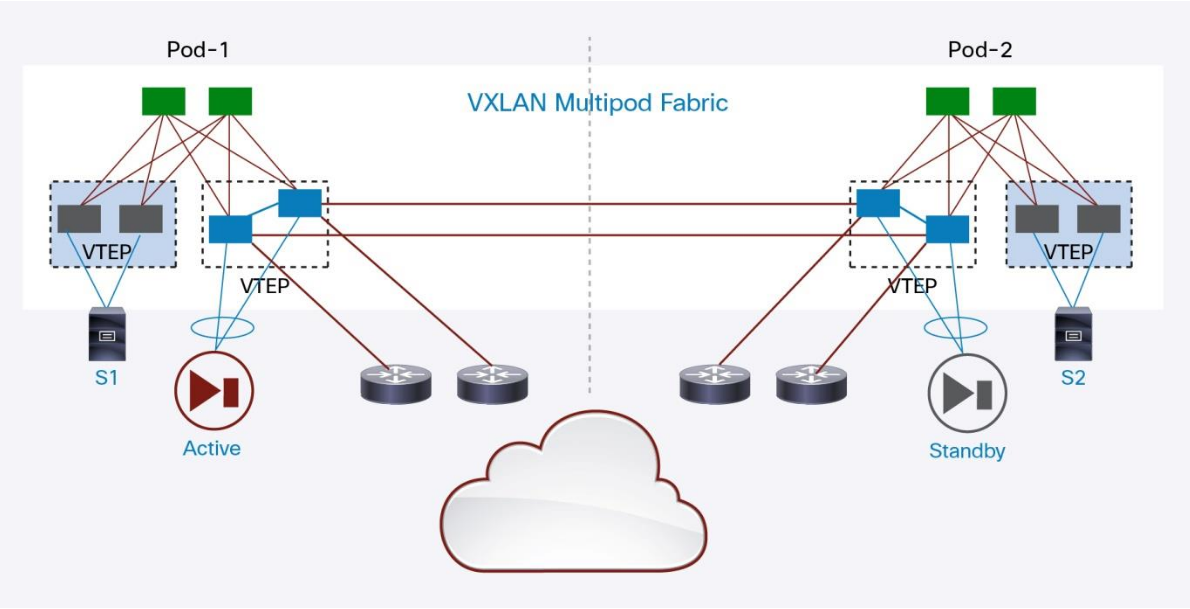
Note: In the specific example of a default static route, you must add the default-information originate configuration option.

4. On the first firewall device, configure the local port channel that will be used for communication with the internal and external networks and create the inside subinterface. (This example uses subinterfaces because the goal is to use the same physical port channel for inbound and outbound connectivity.) Please refer to the Cisco ASA documentation for specific configuration information.

Interconnecting the Firewalls to the WAN Edge Routers

The firewalls must also be connected (logically or physically) to the WAN edge routers that offer the Layer 3 path to the external Layer 3 network domain. Figure 20 shows a physical view of the validated topology.

Figure 20. Connectivity Between Firewalls and WAN Edge Routers



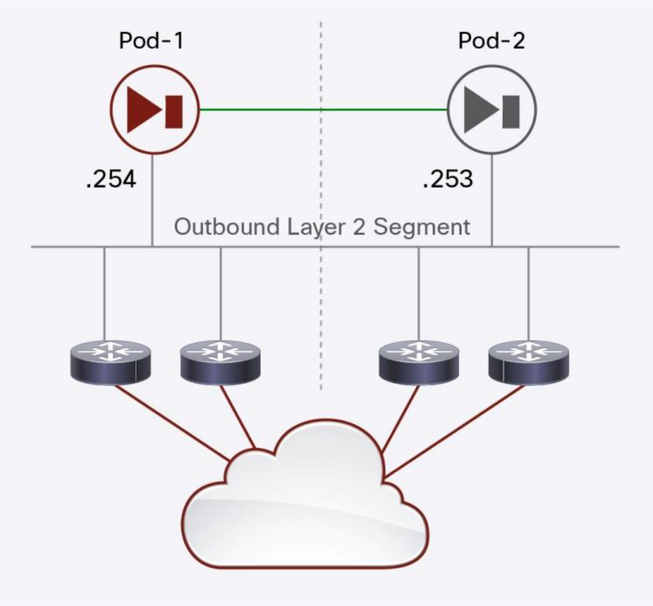
Note the following considerations for the topology:

- The topology assumes that each pod uses its own pair of WAN edge routers to connect to the external Layer 3 network domain (using the traditional FHRP mechanism). This is usually the case even when a multipod design is deployed over metropolitan-area distances. This is even more likely the case over longer distances.
- When using physical firewalls, you can directly connect them to the WAN edge devices using dedicated physical interfaces. In the scenario discussed in this document, the WAN edge routers are instead connected to the VXLAN fabric, which simply provides Layer 2 adjacency (no anycast gateway functions) between the firewall outside interface and the WAN edge routers.

Note: The same port-channel interface on the physical firewall can be used to provide connectivity to both inside and outside logical firewall interfaces. Alternatively (and depending on the specific hardware model), a separate set of interfaces can be used for this purpose.

Figure 21 shows a logical view of the connectivity between the firewall nodes and the WAN edge routers.

Figure 21. Connectivity Between Firewalls and WAN Edge Routers (Logical View)

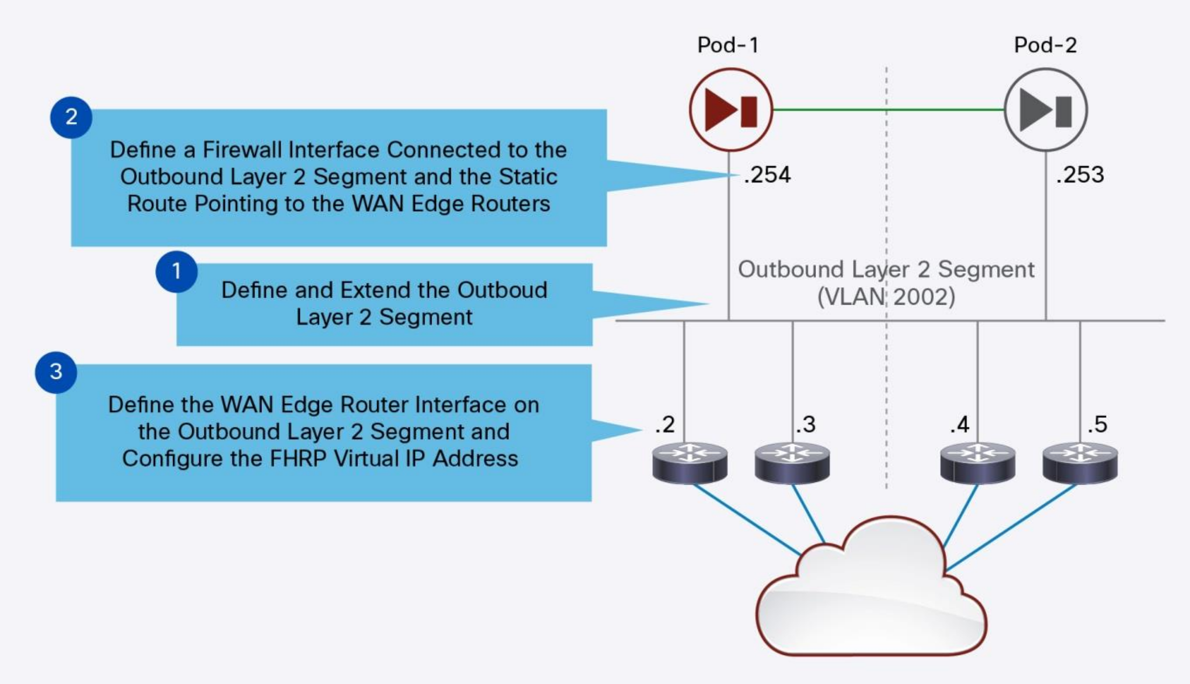


An outbound Layer 2 segment is used to connect the outside interfaces of the active and standby firewall nodes with the WAN edge routers. To achieve this connectivity across pods, the Layer 2 segment is logically extended with VXLAN.

The main reason for connecting all these devices to the same outbound Layer 2 segment is to protect against a specific WAN isolation failure scenario, in which a major outage in the service provider network may totally isolate one pod from the external Layer 3 network domain. Under such circumstances, the active firewall node can start using the path through the WAN edge routers in a remote pod to communicate with external resources.

Figure 22 shows the configuration steps required to establish connectivity between the firewalls and the WAN edge routers for a given tenant.

Figure 22. Configuring Connectivity Between Firewalls and WAN Edge Routers


Pod-1
Pod-2
2
Define a Firewall Interface Connected to the Outbound Layer 2 Segment and the Static Route Pointing to the WAN Edge Routers
.254
.253
1
Define and Extend the Outboud Layer 2 Segment
Outbound Layer 2 Segment (VLAN 2002)
3
Define the WAN Edge Router Interface on the Outbound Layer 2 Segment and Configure the FHRP Virtual IP Address
.2
.3
.4
.5

1. On each VXLAN leaf node (across both pods), define the Layer 2 segment used to connect the outside interface of the firewall and the WAN edge router interfaces to the outbound Layer 2 segment, and be sure that this segment is extended across the pods with VXLAN. Notice that this Layer 2 VNI segment requires only Layer 2 connectivity (that is, no associated Layer 3 SVI needs to be defined on the VXLAN VTEPs).

    - Define the Layer 2 VLAN and map it to the corresponding Layer 2 VNI:

```
vlan 2002
  name ASAv-Outbound
  vn-segment 200002
```

    - Associate the Layer 2 VNI with the NVE interface and with the EVPN portion of the configuration:

```
interface nve1
  no shutdown
  source-interface loopback0
  host-reachability protocol bgp
  member vni 200002
    suppress-arp
    mcast-group 239.1.1.1
!
evpn
vni 200002 l2
    rd auto
```

```
      route-target import 100:2002
      route-target export 100:2002
```

2. On the active firewall node, define the outside interface connected to the outbound Layer 2 segment. Configure the static route (usually a default route) pointing to the FHRP virtual IP address defined on the WAN edge routers (refer to the Cisco ASA documentation for configuration details).
3. On the WAN edge routers, define the Layer 2 VLANs and the Layer 3 SVIs connecting to the outbound Layer 2 segment. Also configure the static routing information pointing to the firewall external interface for the IP subnets defined in the VXLAN that needs to be externally accessible.

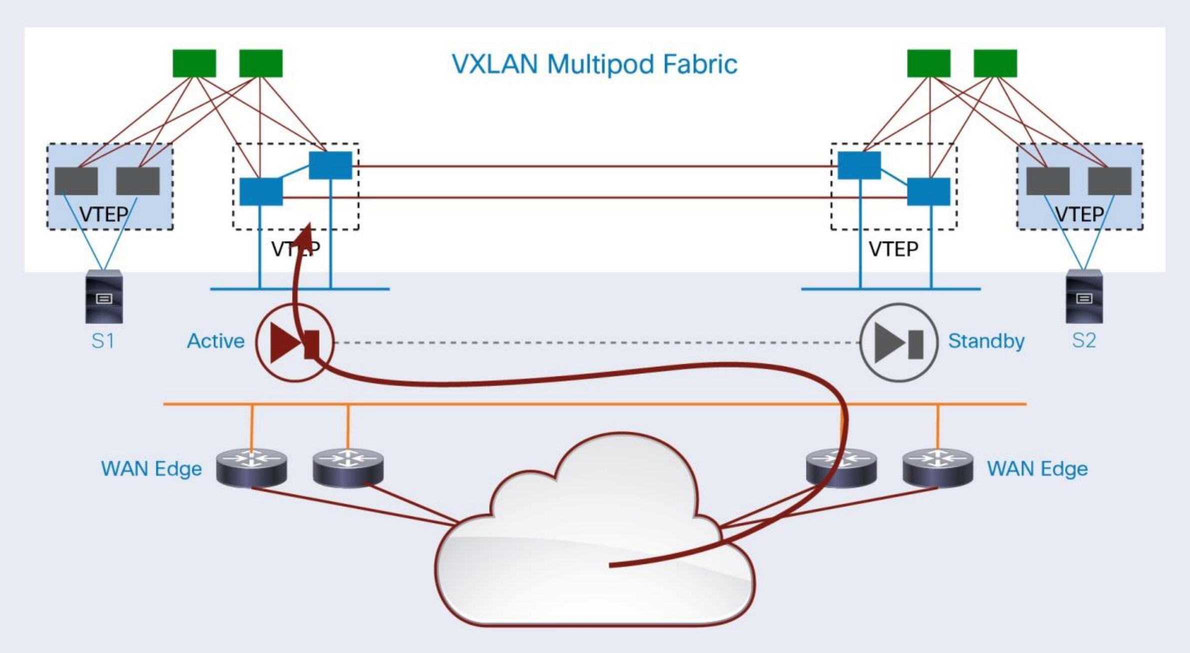
   FHRP (HSRP, VRRP, etc.) is used to provide a consistent Layer 3 next-hop virtual IP address to the active firewall. Because the outbound Layer 2 segment is stretched across pods, the four WAN edge routers are part of the same FHRP domain, provisioned in each pod for failover purposes. Because, as previously stated, the goal under normal conditions is for traffic to enter and exit Pod-1, the FHRP priority for those devices needs to be set accordingly. For example, in an HSRP scenario, Pod-1 WAN edge Router-1 would have the highest priority (and become active), Pod-1 WAN edge Router-2 would have the second highest priority (to make it the standby router), and the two Pod-2 WAN edge routers would go into the listening state.

Note: In this example, the SVIs defined on the WAN edge devices are part of the default VRF instance (global routing table). Thus, routes for all the tenants defined in the VXLAN fabric are mixed together to connect to the external routing domain. In some scenarios, you may need to maintain the logical tenant isolation across the WAN. In that case, you would also need to define VRF instances on the WAN edge router that could perform the Multiprotocol Label Switching (MPLS) provider edge or multiple-VRF instance customer edge role.

Inbound Traffic-Path Considerations

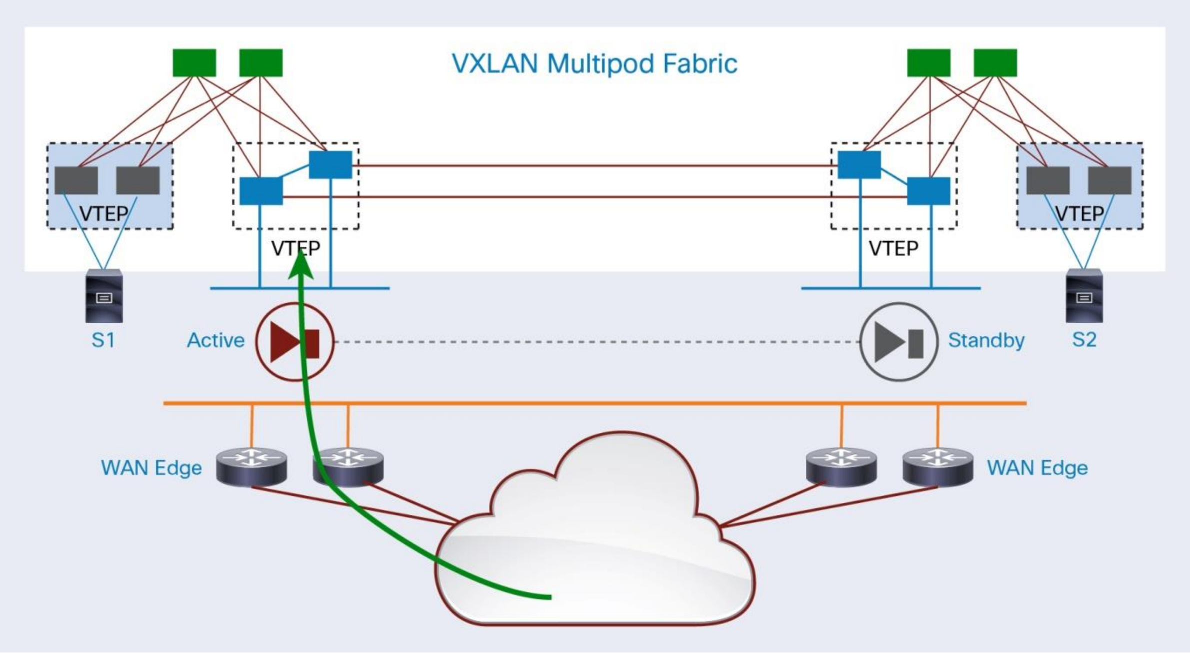
The deployment of a pair of firewalls in active-standby mode implies by definition that inbound traffic can enter the VXLAN multipod fabric only from the pod in which the active firewall node is running. You must be sure to properly configure the system to help ensure that inbound traffic originating in the WAN is not steered to the pod in which the standby firewall is deployed, because that would then force the traffic to cross the interpod links on the outbound Layer 2 segment to reach the active firewall node (Figure 23).

Figure 23. Suboptimal Inbound Traffic Path



Assuming that the active firewall in a steady state is deployed in Pod-1 and moves to Pod-2 only as a consequence of a failover event, you can control the routing information that is injected into the WAN network to help ensure that inbound flows are always steered to the WAN edge routers in Pod-1, as shown in Figure 24.

Figure 24. Optimized Inbound Traffic Path



You can achieve this behavior in several ways, depending on the specific routing protocol used between the WAN edge devices and the WAN routers. Note that after the routing protocol metrics have been tuned, traffic originating from the WAN will always be steered to Pod-1. This will happen even after a firewall failover event that causes the deployment of the active firewall node in Pod-2. In that case, traffic will be rerouted on the interpod links to Pod-2.

The following sample shows how to use autonomous system path prepending (**as-path prepend**) on the WAN edge routers in Pod-2 to make their route advertisement in the WAN less preferable. This option is valid only when BGP is used as the routing protocol between the WAN edge devices and the WAN routers: a common choice in real-life deployments.

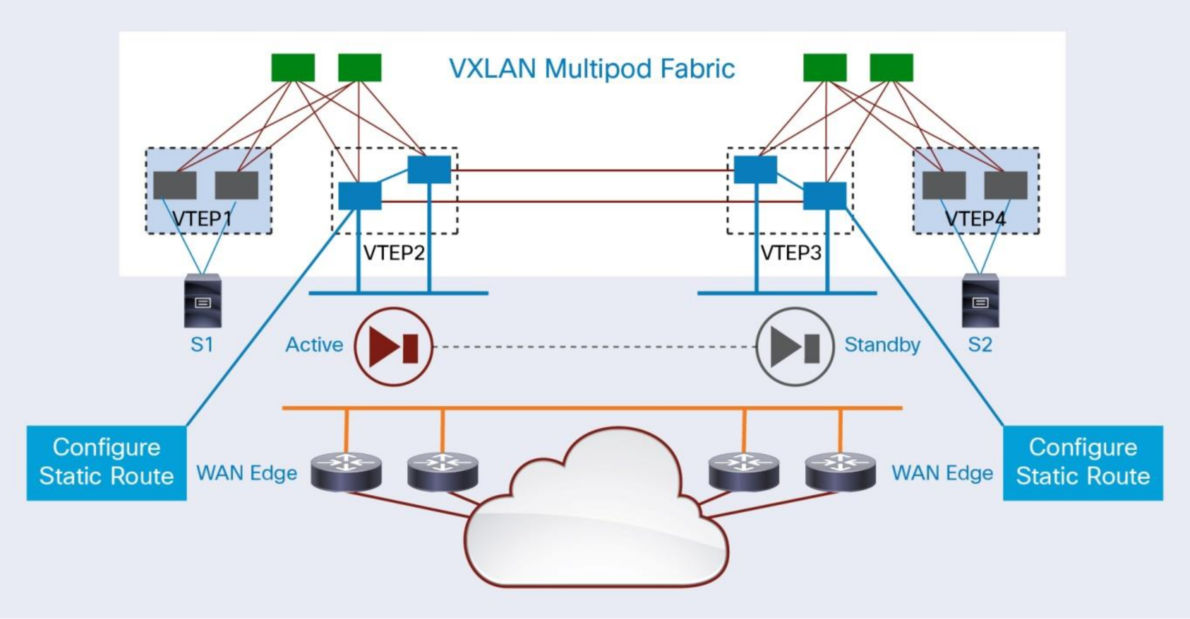
```
route-map AS-PREPEND permit 10
  set as-path prepend 2
!
router bgp 1
  router-id 1.1.1.200
  address-family ipv4 unicast
    redistribute static route-map REDIST-TAG-1234
  neighbor 21.2.1.2 remote-as 100
    address-family ipv4 unicast
      route-map AS-PREPEND out
       route-map AS-PREPEND out
```

Outbound Traffic-Path Considerations

Traffic in the outbound direction must eventually be sent to the pair of VTEP nodes connected to the active firewall node, because they represent the only active path to the external network domain.

When using static routing between the fabric and the firewall nodes, you must configure the static routes on both pairs of VTEP devices—the ones connected to the active firewall and the ones connected to the standby firewall—as shown in Figure 25.

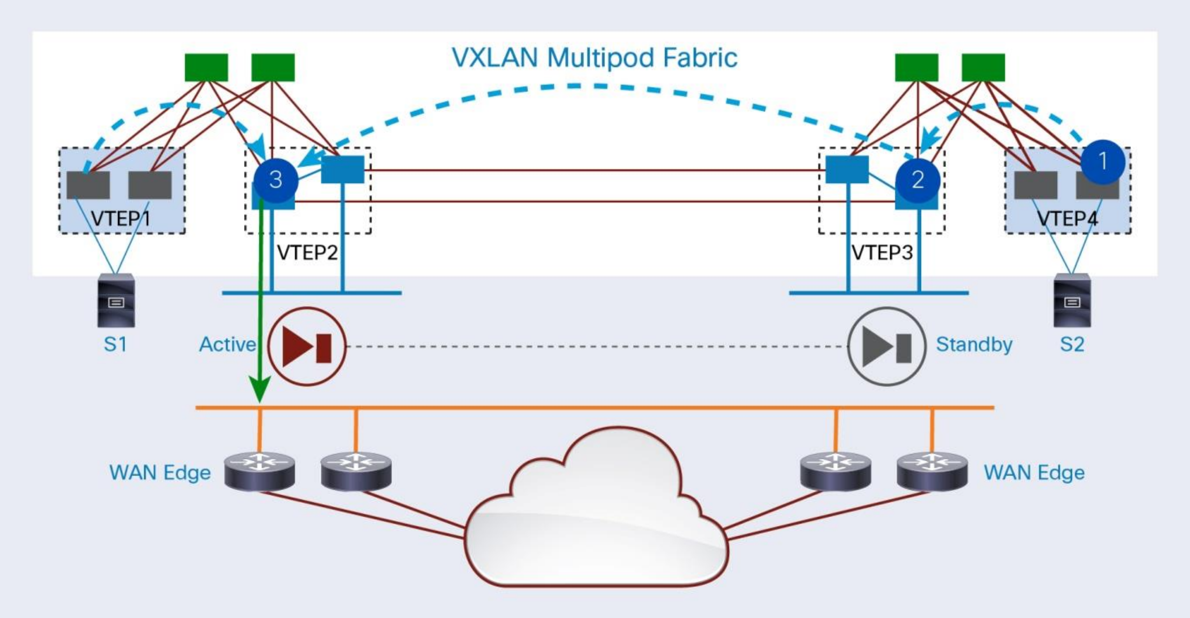
Figure 25. Static Routes Configuration



The static route pointing as a next Layer 3 hop to the IP address associated with the internal interface of the active firewall node is needed on both pairs of VTEP devices to handle the firewall failover scenario. After the static routes are defined, both pairs of VTEPs by default redistribute this information to the EVPN control plane to inform the remote computing leaf nodes deployed in the multipod fabric.

In a simple example in which the VTEPs nodes connected to the firewall devices redistribute only a default route to the fabric, the outbound traffic flows originating from endpoints connected to different computing leaf nodes would look like the ones shown in Figure 26.

Figure 26. Outbound Traffic Flows



The various computing leaf nodes always prefer the default route advertisement originating from the pair of VTEPs connected to the firewall in the local pod. They prefer this advertisement because the metric to reach the local BGP EVPN next hop (anycast VTEP) is preferable to that to reach the remote anycast VTEP. This behavior is independent of whether the local VTEPs are connected to the active or standby firewall node.

The results are summarized in the following sequence of steps (shown in Figure 26):

1. After receiving a data packet directed to an external destination, the local VTEP4 performs a Layer 3 lookup and finds the default route information advertised by VTEP3 in the local pod connected to the standby firewall node. As a consequence, it performs VXLAN encapsulation and sends the packet to VTEP3 (anycast VTEP).
2. One of the receiving physical leaf nodes that is part of the vPC domain decapsulates the packet and performs the Layer 3 lookup. The result is that the next hop for the default route is the IP address of the active firewall interface. Hence, VTEP3 performs a recursive Layer 3 lookup to discover that the active firewall interface is learned through MP-BGP from VTEP2 located in Pod-1. VTEP3 hence reencapsulates the frame and sends it to VTEP2.
3. One of the leaf nodes that is part of VTEP2 decapsulates the frame and routes it to the active firewall node. The firewall node then has a default route configured to point to the active HSRP virtual IP address available on the local WAN edge routers.

The required configuration for the static route is simple and is shown in the sample here (this configuration applies to the four physical leaf nodes connected to the active and standby firewall devices):

```
vrf context Tenant-1
  vni 300001
  ip route 0.0.0.0/0 20.1.1.254 tag 1234      ← Firewall inside interface as next
Layer 3 hop
!
route-map TAG-1234 permit 10
  match tag 1234
!
router bgp 65500
  router-id 10.0.0.13
  vrf Tenant-1
    router-id 10.0.0.13
    address-family ipv4 unicast
      redistribute static route-map TAG-1234 ← Redistribution to the EVPN
control plane
      default-information originate
```

The static route (in the example here simply a default route) points to the firewall inside interface as the next Layer 3 hop, and it is redistributed to the EVPN control plane as soon as it is defined. Thus, VTEPs in Pod-1 will have that route in their routing table pointing to the anycast VTEP of the local leaf nodes connected to the active firewall, whereas VTEPs in Pod-2 will have a route pointing to the anycast VTEP of the local leaf nodes connected to the standby firewall, as shown in the output samples that follow:

VTEP in Pod-1

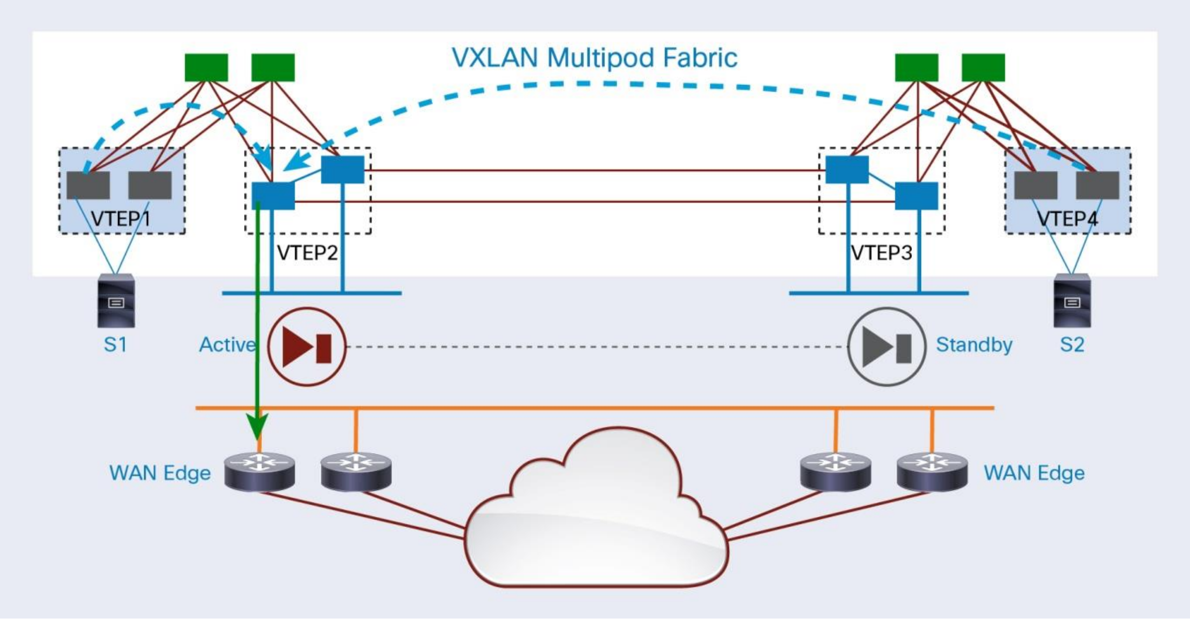
```
Leaf1-Pod1# sh ip route vrf Tenant-1
IP Route Table for VRF "Tenant-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
0.0.0.0/0, ubest/mbest: 1/0
    *via 10.0.2.13%default, [200/0], 00:00:13, bgp-65500, internal, tag 65500
(evpn) segid: 300001 tunnelid: 0xa00020d encap: VXLAN
```

VTEP in Pod-2

```
Leaf7-Pod2# sh ip route vrf tenant-1
IP Route Table for VRF "Tenant-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
0.0.0.0/0, ubest/mbest: 1/0
    *via 10.0.2.25%default, [200/0], 00:00:37, bgp-65501, internal, tag 65501
(evpn) segid: 300001 tunnelid: 0xa000219 encap: VXLAN
```

To avoid the behavior just described, in which the VTEPs connected to the standby firewall node have to decapsulate and reencapsulate the traffic, only the VTEP leaf nodes connected to the active firewall should inject static routing information into the EVPN control plane. This approach supports the behavior shown in Figure 27, in which VTEPs deployed in separate pods send VXLAN encapsulated traffic only to the VTEPs connected to the active firewall node.

Figure 27. Optimal Traffic Forwarding



NX-OS Release 7.0(3)I2(2) introduced a new function: Host Mobility Manager (HMM) route tracking. This function activates a static route (and consequently redistributes it to the routing control plane) only if the specified next-hop IP address is discovered to be locally connected. Therefore, only the VTEP nodes connected to the active firewall will advertise the static route and start attracting traffic flows destined for the outside network, creating the traffic pattern shown in Figure 27.

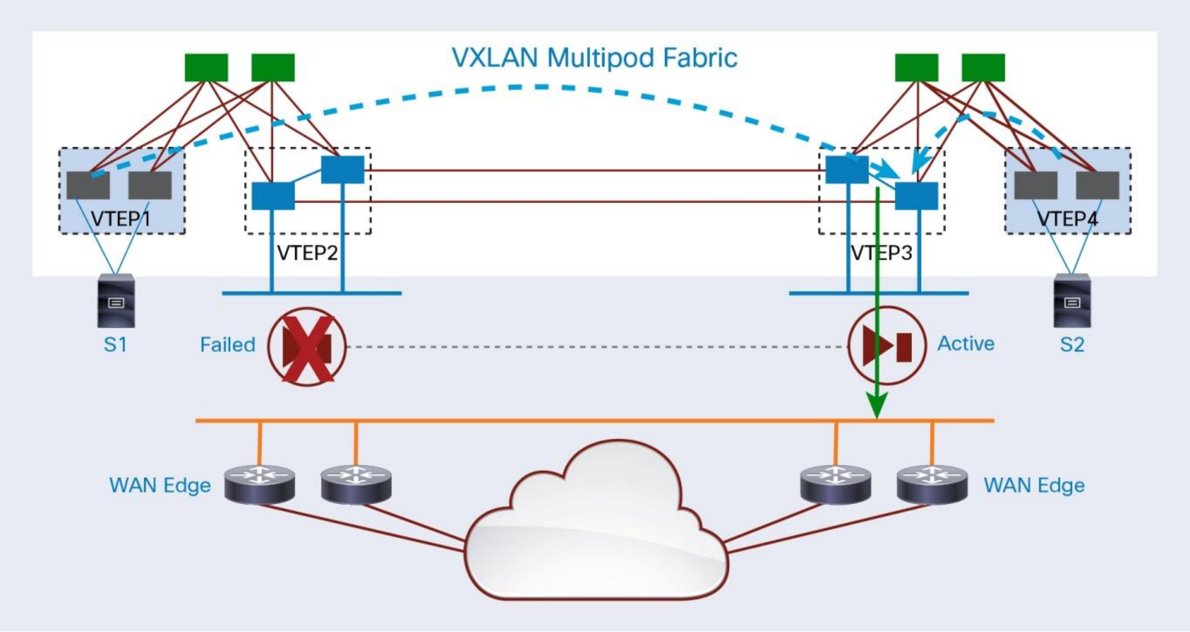
So that the VTEPs can determine whether they are connected to the active firewall node that represents the static route next hop, you can track the local discovery of the next-hop address (the IP address of the active firewall interface). As a result, only VTEPs connecting to the active firewall node can inject the static route of interest (the default route in this specific example) into the VXLAN EVPN control plane.

Note: The tracking mechanism discussed in this context does not rely on the generation of probes sent to the specified IP address. It simply tracks the existence of a specific host route in the local Layer 2 RIB table of the VTEP and the mechanism through which it has been learned. If the host route was learned locally, it is marked with the HMM label, and the result of the tracking is positive (Up state). If the host route was learned through the MP-BGP control plane (meaning that the active firewall node is connected to a different VTEP), it is marked with the BGP label, and the result of the tracking is negative (Down state).

After the firewall experiences a failover event, the HMM process on the VTEPs connected to the newly activated firewall node locally learns the active firewall IP address, causing the activation of the static route and its injection into the MP-BGP EVPN control plane. Traffic flows originating from endpoints connected to the computing leaf nodes will hence be redirected to the new pair of VTEPs, as shown in Figure 28.

In Pod-1, VTEP2 immediately receives from the BGP control plane an advertisement for the endpoint firewall with the same identifiers (IP and MAC addresses), but with a higher sequence number. As a result, the VTEP updates the firewall endpoint information in its forwarding tables and withdraws the static route from the control plane.

Figure 28. Firewall Failover Scenario



The configuration required to enable HMM route tracking is simple and is shown in the following sample (this configuration must be applied to all four VTEPs connected to the active and standby services nodes):

```
track 1 ip route 20.1.1.254/32 reachability hmm ← Track the static route next hop

  vrf member Tenant-1

!

vrf context Tenant-1

  vni 300001

  ip route 0.0.0.0/0 20.1.1.254 track 1 tag 1234
```

As a consequence of this configuration, all the VTEPs (local and remote) start pointing only to the anycast VTEP for the pair of leaf nodes connected to the active firewall node, as shown in the following examples:

VTEP in Pod-1

```
Leaf1-Pod1# sh ip route vrf Tenant-1

IP Route Table for VRF "Tenant-1"

'*' denotes best ucast next-hop

'**' denotes best mcast next-hop

'[x/y]' denotes [preference/metric]

'%<string>' in via output denotes VRF <string>

0.0.0.0/0, ubest/mbest: 1/0

    *via 10.0.2.25%default, [200/0], 00:01:26, bgp-65500, internal, tag 65501 (evpn) segid: 300001 tunnelid: 0xa000219 encap: VXLAN
```

VTEP in Pod-2

```
Leaf7-Pod2# sh ip route vrf tenant-1
IP Route Table for VRF "Tenant-1"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
0.0.0.0/0, ubest/mbest: 1/0
    *via 10.0.2.25%default, [200/0], 00:23:46, bgp-65501, internal, tag 65501
(evpn) segid: 300001 tunnelid: 0xa000219 encap: VXLAN
```

Note: Route tracking is needed only in cases in which the static routes are configured only on the VTEP nodes closest to the firewall. An alternative approach is to configure the static routes on all remote leaf nodes (computing leaf nodes) with a next hop of the firewall (recursive next hop). With the existence of a next hop behind a VXLAN tunnel, no special tracking needs to be configured, because the respective IP address is seen only behind the VTEP of the active firewall. This capability is provided through the generic host-mobility learning, advertisement, and move sequence.

Improving the Sturdiness of the VXLAN Multipod Design

Neither VXLAN flood and learn nor the current implementation of VXLAN EVPN natively offer embedded functions that allow multihoming (that is, creation of more than one Layer 2 path outside a single VXLAN fabric) while they detect the creation of end-to-end Layer 2 loops and protect against them.

Although this scenario may seem unlikely in real-life deployments, this problem may occur as a result of a misconfigured network device or a mistake in setting up patch-panel connections.

Note: The considerations discussed in this section are not specific to a VXLAN multipod design, but also apply to an intrasite VXLAN EVPN fabric deployment.

Until all functions are embedded in the VXLAN to prevent the creation of Layer 2 loops, alternative short-term solutions are needed. The approach suggested in this document consists of the following two processes:

- Use edge-port protection features natively available on Cisco Nexus platforms in conjunction with VXLAN EVPN to **prevent** the creation of a Layer 2 loops. The best-known feature is BPDU Guard. BPDU Guard is a Layer 2 security tool that should be enabled on all the edge switch interfaces that connect to endpoints (hosts, firewalls, etc.). It prevents the creation of a Layer 2 loop by disabling the switch port after it receives a spanning-tree BPDU (no BPDUs should ever be received on an edge interface). BPDU Guard can be configured at the global level or the interface level. It is discussed in more detail in the section “Prevention of End-to-End Layer 2 Loops ”
- A different mechanism is required to mitigate the effects of end-to-end Layer 2 loops for all the scenarios in which BPDU Guard is not effective (for example, in cases in which the BPDUs are not reaching the switch interface, or in which connecting the VXLAN fabric to an external Layer 2 network requires BPDU Guard to be disabled on those interfaces). The section “Mitigation of End-to-End Layer 2 Loops” discusses the use of the storm-control function available on Cisco Nexus platforms to rate-limit the broadcast storm generated between pods as a consequence of the creation of a Layer 2 loop.

Prevention of End-to-End Layer 2 Loops

In overlay network technologies such as VXLAN, management of broadcast, unknown unicast, and multicast traffic is a crucial requirement. In a case in which interconnection across separate pods is multihomed, one—and only one—edge device must forward the broadcast, unknown unicast, and multicast traffic to the core or to the remote pod as appropriate.

If by mistake two Layer 2 segments are welded together (locally or across different pods), a tool should automatically detect the presence of another active device (for example, sending and listening for Hello messages) for the same Ethernet segment.

In the current absence of a native mechanism built in to VXLAN, you can use specific edge-port-centric services to protect against potential Layer 2 loops. The best-known feature for that purpose is BPDU Guard. BPDU Guard is a Layer 2 edge-port protection feature that disables the switch interface when it receives a spanning-tree BPDU.

BPDU Guard provides an active response to invalid configurations, requiring the network manager to manually put the Layer 2 LAN interface back in service after an invalid configuration. This feature can be enabled globally or at the interface level:

- When configured at the interface level, BPDU Guard shuts down a port as soon as it receives a BPDU, regardless of the port-type configuration.
- When configured globally on the leaf node, BPDU Guard is effective only on operational spanning-tree edge ports.

In a common configuration, Layer 2 edge interfaces should not receive BPDUs because they provide endpoint connectivity. When a BPDU is seen on one of the edge interfaces, this event signals an invalid configuration and will cause the disablement of the interface, preventing a potential Layer 2 loop from occurring.

The recommended configuration is shown in the following sample. It consists of globally enabling BPDU Guard on all the **port type edge** interfaces. To help ensure that BPDU Guard is enabled by default on all the interfaces of the switch, you can also globally configure **type edge** as the default on all the ports.

```
spanning-tree port type edge bpduguard default
spanning-tree port type edge default
```

BPDU Guard is active on interfaces (access or trunks) configured as **port type edge**. The following example shows an interface edge trunk connected to a hypervisor:

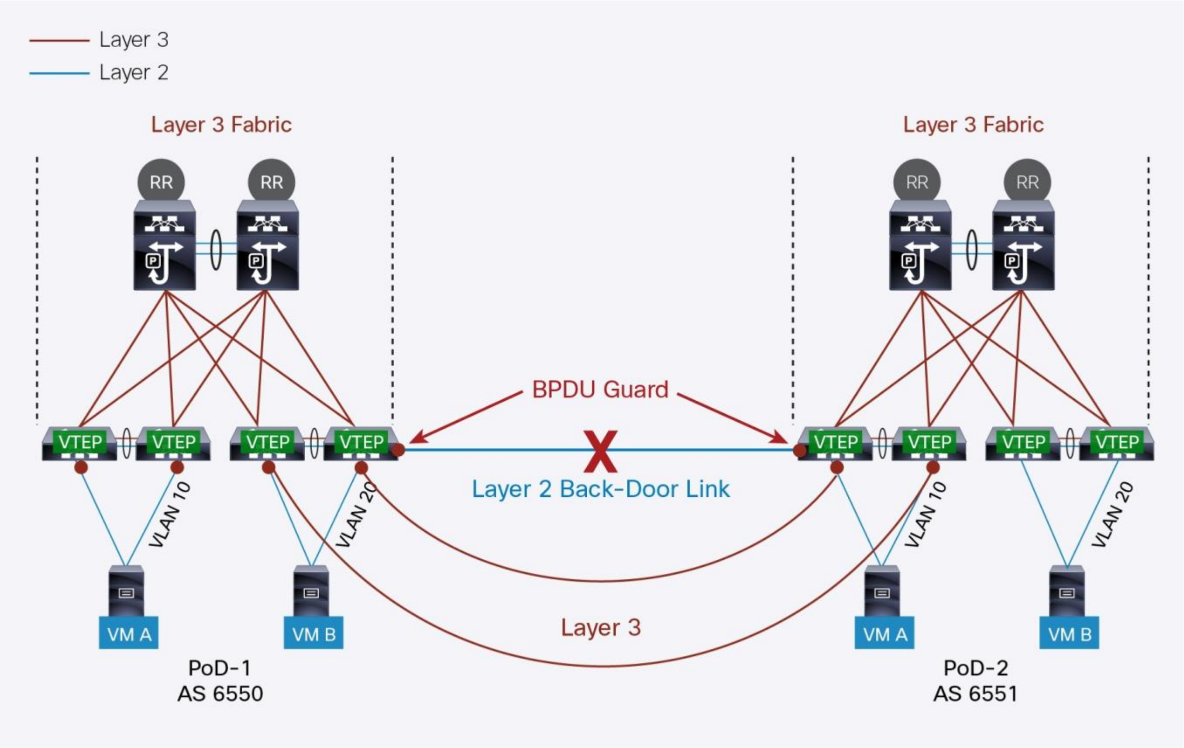
```
interface port-channel36
   switchport mode trunk
   spanning-tree port type edge trunk
```

BPDU Guard is not effective on interfaces of different types, such as the vPC peer link (**type network**):

```
interface port-channel3334
   switchport mode trunk
   spanning-tree port type network
   vpc peer-link
```

In Figure 29, which shows a VXLAN multipod deployment, BPDU Guard is globally enabled on all leaf switches. Consequently, if any Layer 2 link is established between two leaf switches (local or remote), BPDUs will be received, triggering the shutdown of the corresponding interface.

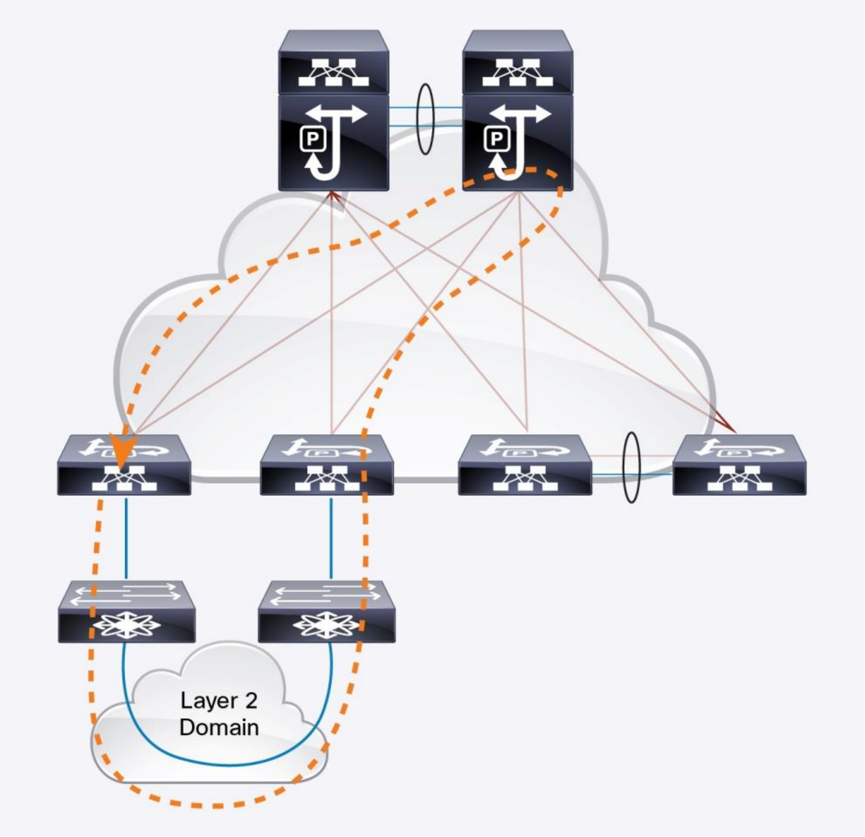
Figure 29. Using BPDU Guard to Break End-to-End Layer 2 Loops



In some specific scenarios, BPDU Guard does not prevent the creation of end-to-end Layer 2 loops. One simple example is the misconfiguration of a firewall deployed in bridge mode, because BPDUs are not bridged between the firewall interfaces.

Another possible scenario is a case in which the VXLAN fabric is connected to an external Layer 2 domain through a redundant Layer 2 path, as shown in Figure 30.

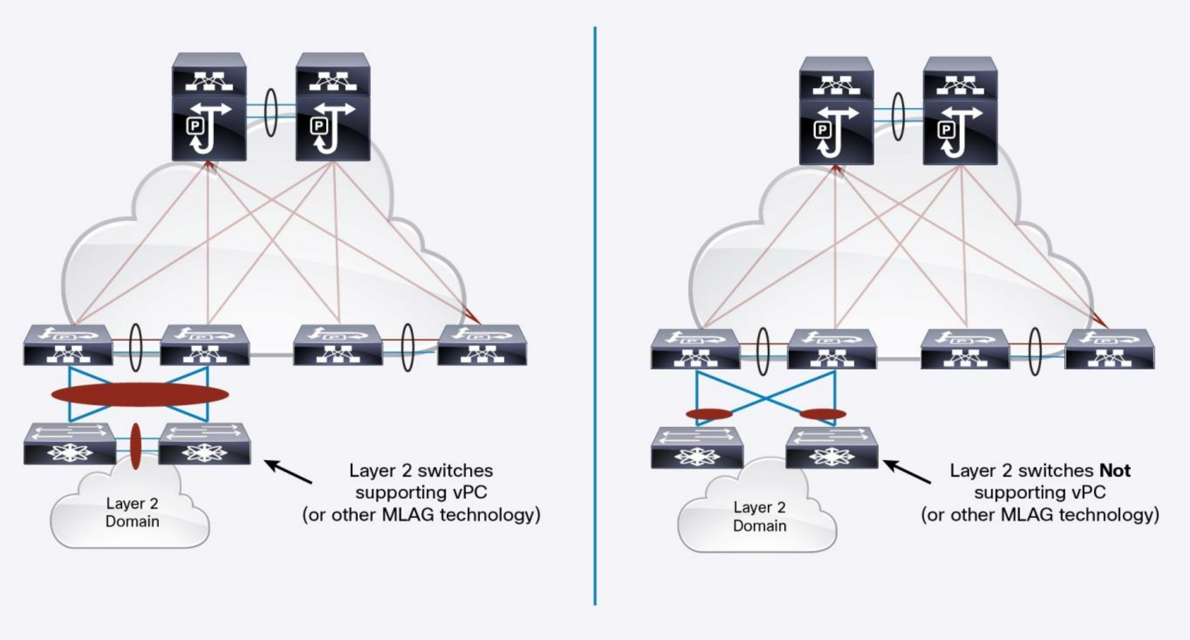
Figure 30. Creating a Layer 2 Loop with an External Layer 2 Network



In the case in Figure 30, you cannot enable BPDU Guard on the interfaces connected to the external Layer 2 network. Therefore, this mechanism cannot be used to break the potential end-to-end Layer 2 loop.

To avoid the creation of loops, be certain that Layer 2 connectivity to an external network domain is established only using Cisco vPC technology on a single pair of VTEP leaf nodes. Depending on the support for vPC (or any equivalent multichassis link aggregation [MLAG] technology) on the external Layer 2 switches, the two topologies shown in Figure 31 are possible.

Figure 31. Using vPC on a Pair of VTEPs for External Layer 2 Connectivity

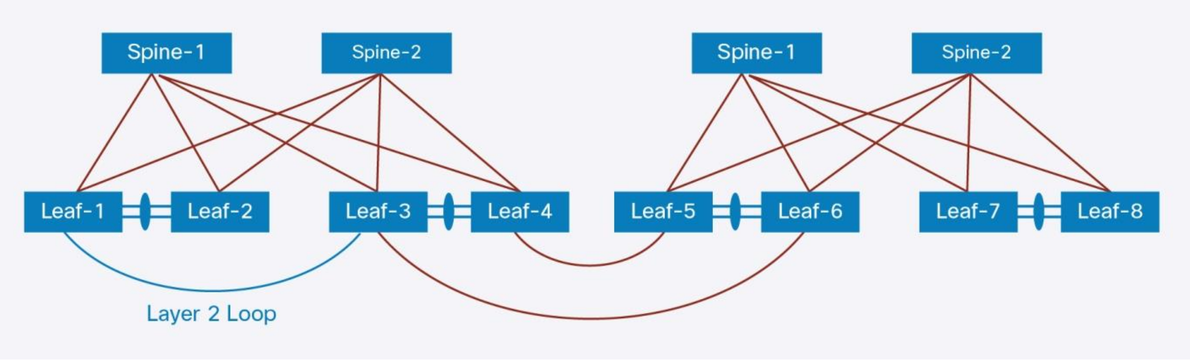

Layer 2 Domain
Layer 2 switches supporting vPC (or other MLAG technology)
Layer 2 Domain
Layer 2 switches **Not** supporting vPC (or other MLAG technology)

Mitigation of End-to-End Layer 2 Loops

A VXLAN multipod fabric functionally represents a single VXLAN fabric (a single availability zone). Thus, the inadvertent creation of a Layer 2 loop in a given pod would by default affect all the other pods. It is therefore highly recommended that you put in place configurations to mitigate the effects of a Layer 2 loop, helping ensure that remote pods remain fully functional.

To validate a possible mitigation solution based on the use of the storm-control function, a Layer 2 loop was artificially created by directly connecting Leaf-1 and Leaf-3 inside Pod-1, as shown in Figure 32 (for this purpose, the default BPDU Guard configuration was removed from those interfaces).

Figure 32. Creating a Layer 2 Loop Using a Back-Door Cable

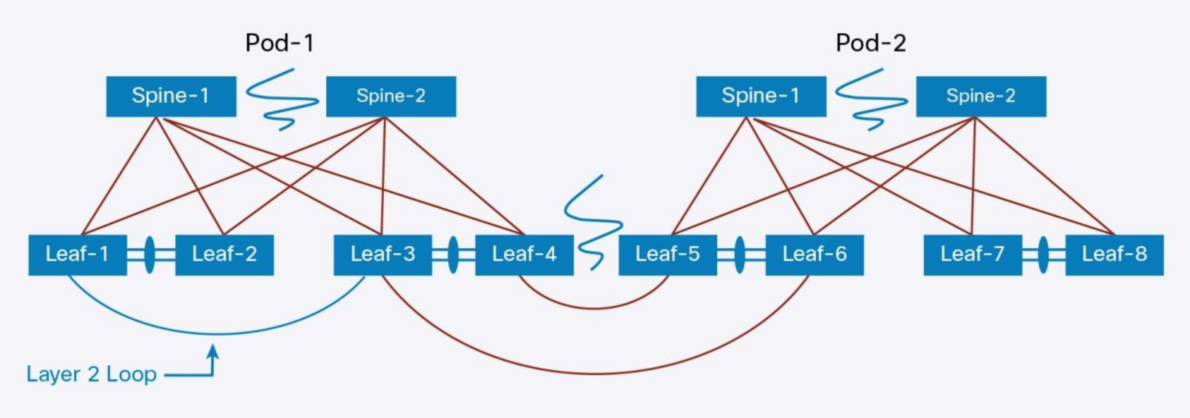


Subsequent sections discuss the use of storm control for loop mitigation. The following section discusses the impact that loops have on network components (data plane and control plane) and on the CPU of the endpoints connected across all the pods.

Impact of Loops on Network Links and Servers CPU Utilization

As a result of the creation of a Layer 2 loop, a single Layer 2 broadcast Address Resolution Protocol (ARP) request begins being infinitely replicated in the network, and all access links to hosts become overloaded (including the interpod links), as shown in Figure 33.

Figure 33. Physical Links Overloaded

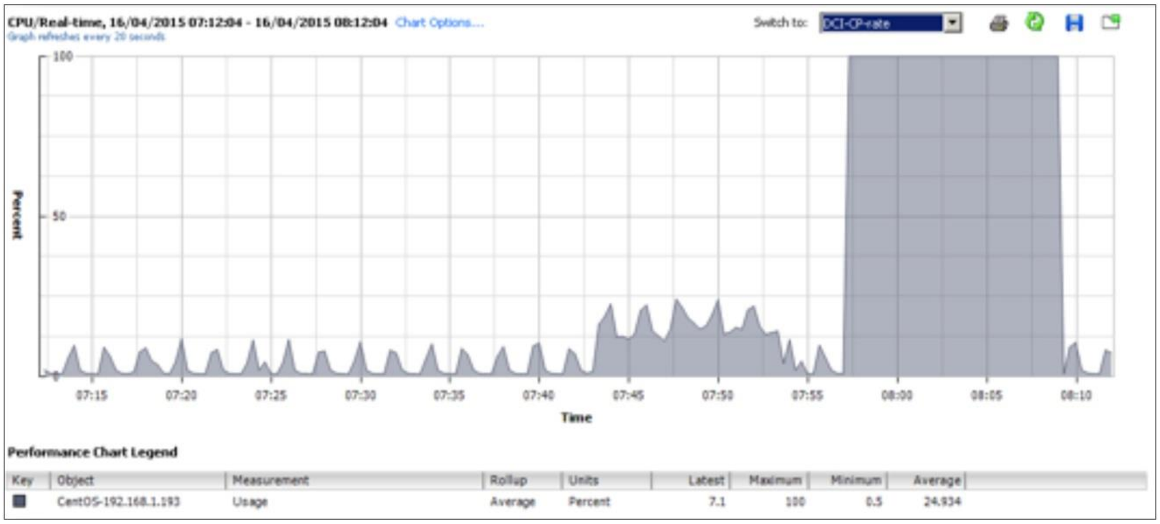


The intrapod 40 Gigabit Ethernet uplinks connecting leaf and node spines are also used to send almost 10 Gbps of traffic, as shown in the following output. This behavior occurs because the Layer 2 storm was created by looping two 10 Gigabit Ethernet interfaces, hence limiting the whole loop to a maximum of 10 Gbps of broadcast traffic.

```
Leaf6-Pod2# sh int e1/37 | i rate ← Interpod link of Leaf-6 in Pod-2
  30 seconds input rate 9201302624 bits/sec, 5000700 packets/sec
  30 seconds output rate 500056 bits/sec, 541 packets/sec
Spine1-Pod2# sh int e1/20 | i rate ← Spine-1 in Pod-2: Uplink to Leaf-6
  30 seconds input rate 9243077976 bits/sec, 4693065 packets/sec
  30 seconds output rate 296 bits/sec, 0 packets/sec
    input rate 9.28 Gbps, 4.69 Mpps; output rate 48 bps, 0 pps
Spine1-Pod2# sh int e1/21 | i rate ← Spine-1 in Pod-2: Uplink to Leaf-7
  30 seconds input rate 1560 bits/sec, 1 packets/sec
  30 seconds output rate 9247549192 bits/sec, 4695374 packets/sec
    input rate 1.17 Kbps, 1 pps; output rate 9.25 Gbps, 4.68 Mpps
Spine1-Pod2# sh int e1/22 | i rate ← Spine-1 in Pod-2: Uplink to Leaf-8
  30 seconds input rate 1800 bits/sec, 1 packets/sec
  30 seconds output rate 9262357912 bits/sec, 4687381 packets/sec
    input rate 1.23 Kbps, 1 pps; output rate 9.25 Gbps, 4.68 Mpps
```

In the context of the multipod solution, because VXLAN is used to extend Layer 2 connectivity across pods, the broadcast storm traffic also reaches the remote transit leaf nodes through the interpod links. The storm is therefore also propagated to all remote leaf nodes through their local spine layers. The VXLAN header is finally removed by the VTEPs, and the traffic is then flooded to all the endpoints that are part of the same Layer 2 broadcast domain affected by the storm. The result is increased use of the host ingress interface and increased server CPU utilization rate, as shown in Figure 34.

Figure 34. Virtual Machine CPU Saturation to Manage an ARP Storm Without Storm Control



CPU/Real-time, 16/04/2015 07:12:04 - 16/04/2015 08:12:04 Chart Options...

Graph refreshes every 20 seconds

Switch to: DCI-CP-rate

Performance Chart Legend

Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
■	CentOS-192.168.1.193	Usage	Average	Percent	7.1	100	0.5	24.934

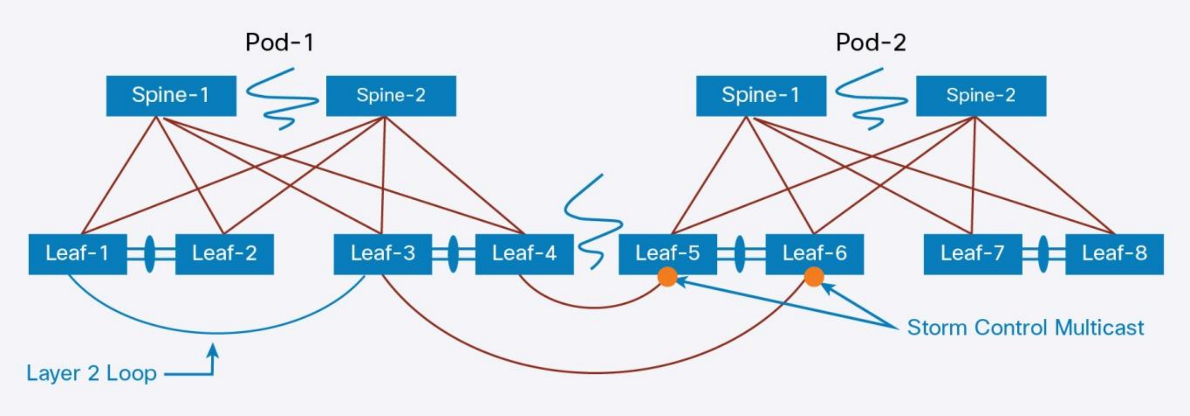
The net result is that not only is the network itself oversubscribed, but as shown in Figure 34, servers in both the local and remote pods are directly affected by the increased amount of broadcast traffic that they need to handle.

Using Storm Control to Protect Remote Pods

When multicast replication is used to handle the distribution of Layer broadcast, unknown unicast, and multicast traffic, you can use the storm-control function to rate-limit the amount of multicast traffic received on the ingress interfaces of the transit leaf nodes (interpod links), hence protecting all networks behind it (Figure 35).

Note: If ingress replication (unicast) is used for broadcast, unknown unicast, and multicast traffic, the approach described here does not apply.

Figure 35. Applying Storm-Control Multicast on Interpod Links



To determine the specific value to configure for this rate-limiting function, you must consider the following two conditions:

- Consider the existence of known multicast-based applications (for example, applications used for media distribution) that spread across pods.
- Then consider the network utilization of data multicast traffic in the production environment. More precisely, consider the percentage of resources used by broadcast, unknown unicast, and multicast traffic consumed by the applications under normal circumstances. This value is important to enable you to determine the minimum threshold value that can be used to rate-limit the Layer 2 multicast streams without affecting the applications when a broadcast storm is not present.

Selecting the Threshold Value for Storm Control in Interpod Links

As a general rule, the rate-limit value for broadcast, unknown unicast, and multicast traffic should be greater than the application data multicast traffic (use the highest value as a reference) plus the percentage of permitted broadcast traffic. If the multicast traffic spawned by an application consumes N percent, and the normal percentage of broadcast traffic is X percent, the storm-control multicast threshold should be equal to N + X.

The threshold value selected for the rate limiter must be well dimensioned for the CPU utilization of each server. To better understand the impact on the CPU utilization, different successive multicast storm-control threshold values have been applied at ingress on the transit leaf nodes in Pod-2, and in each scenario the direct impact on the server CPU has been recorded.

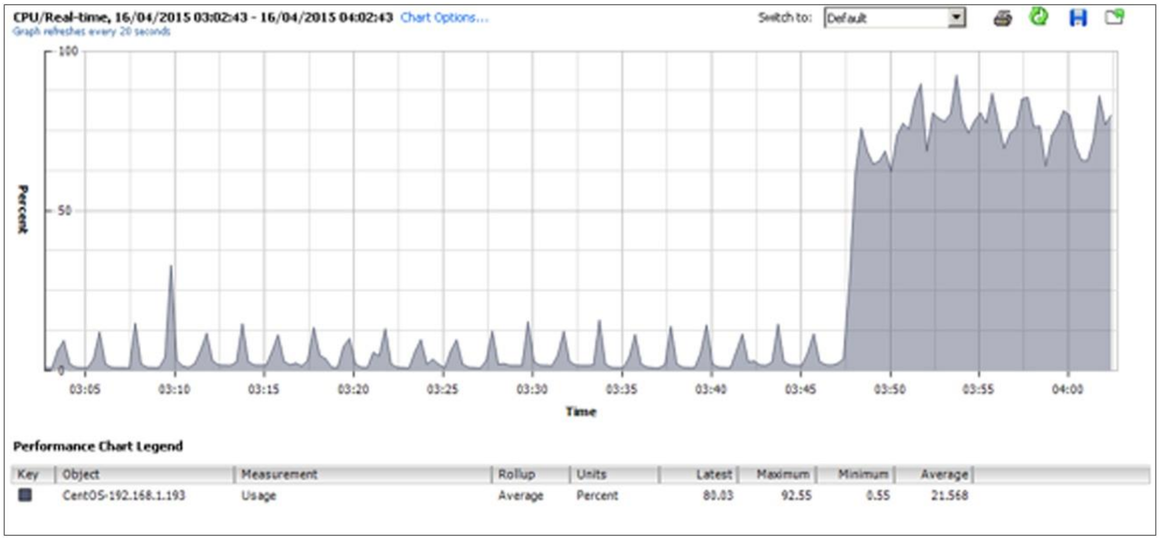
In the first scenario, the storm-control limiter is set to 10 percent for the 10-Gbps interpod ingress interfaces, using the configuration sample shown here:

```
interface Ethernet1/37
  description "Inter-pod Link"
  no switchport
  storm-control multicast level 10.00
  ip address 192.168.36.13/24
  ip pim sparse-mode
  no shutdown
```

As a result, the inbound 10 Gbps of multicast traffic received from the interpod link is rate-limited to 1 Gbps before it is sent to Pod-2. With this setting, the average amount of broadcast traffic that reaches the destination endpoints (virtual machines) belonging to the same Layer 2 broadcast domain represents almost 10 percent of the 10 Gigabit Ethernet interface capacity (note that the VXLAN header used to carry the multicast traffic is stripped off when the traffic is forwarded to the endpoints).

As shown in Figure 36, this amount of traffic was still sufficient to consume almost all the CPU resources of the receiving virtual machine.

Figure 36. CPU Use for a Virtual Machine in a VLAN Disrupted by a Broadcast Storm



CPU/Real-time, 16/04/2015 03:02:43 - 16/04/2015 04:02:43

Graph refreshes every 20 seconds

Switch to: Default

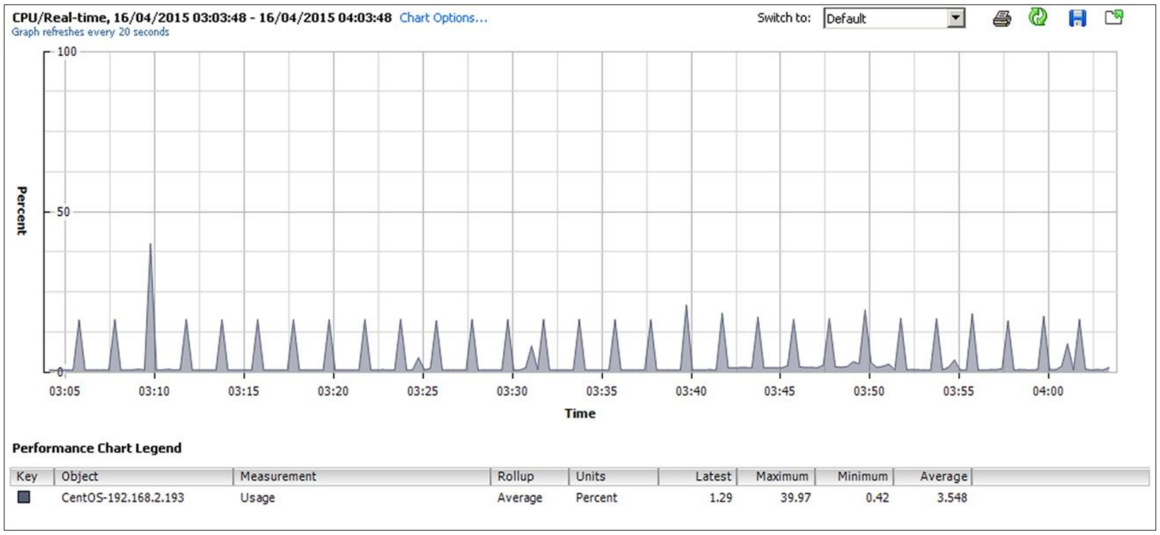
Performance Chart Legend

Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
■	CentOS-192.168.1.193	Usage	Average	Percent	80.03	92.55	0.55	21.568

Application processes were not responding (for example, pings were not acknowledged). As a consequence, rate limiting the storm-control multicast traffic at 10 percent was not sufficient to protect the virtual machines in the tested environment.

Note that the machines belonging to different broadcast domains (different VLANs not affected by the broadcast storm), even though they share the same multicast group for broadcast, unknown unicast, and multicast traffic, are not affected by the residual 10 percent of the broadcast storm. The reason is that the egress VTEP that receives the broadcast traffic from the VXLAN fabric floods the traffic to the local interfaces that are part of the VLAN mapped to the affected VNI, without affecting the interfaces mapped to other VLANs. This behavior is demonstrated in Figure 37, which shows the CPU utilization of a virtual machine located on the same ESXi host but connected to a different Layer 2 VNI segment not affected by the broadcast storm.

Figure 37. Virtual Machine on a Different Layer 2 VNI Not Affected by the Broadcast Storm



Performance Chart Legend

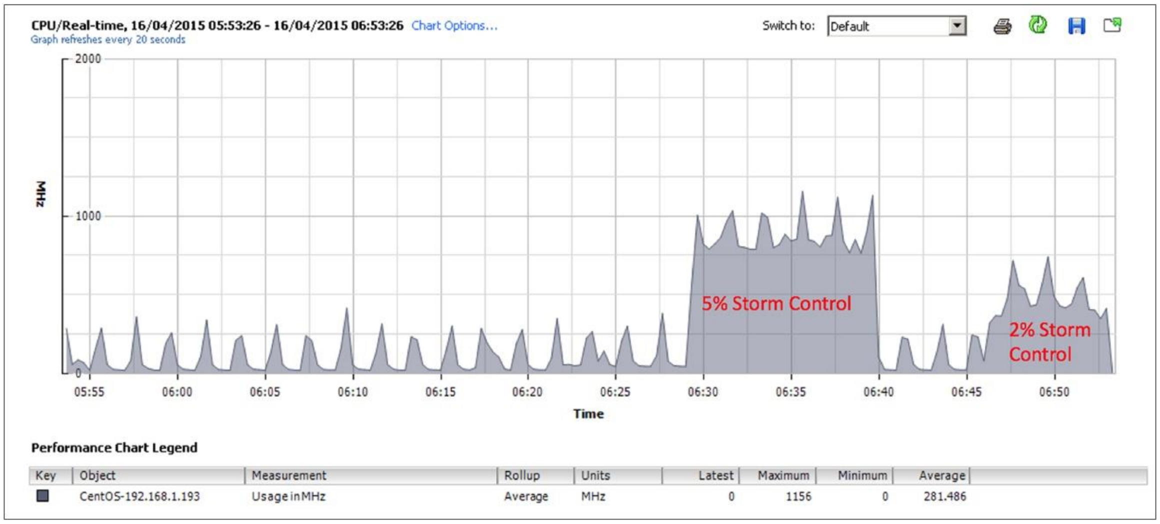
Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
■	CentOS-192.168.2.193	Usage	Average	Percent	1.29	39.97	0.42	3.548

Additionally, as discussed in the “Underlay Multicast Configuration” section, each Layer 2 VNI can rely on an underlay IP multicast group to carry the Layer 2 broadcast, unknown unicast, and multicast traffic. As a consequence, in the specific scenario of a Layer 2 broadcast storm discussed in this section, this traffic propagates across the IP multicast tree to all destination egress VTEPs that share the same multicast group (according to the local Layer 2 VNI configuration).

For example, if all Layer 2 VNIs use the same IP multicast group, then the multicast tree will reach all the leaf nodes. However, if each Layer 2 VNI uses a unique multicast group, the multicast tree will reach only the VTEPs (leaf nodes) in which the Layer 2 VNI of interest is configured. This consideration is important to keep in mind when deciding how to associate multicast groups with Layer 2 VNI segments. You need to help ensure that a Layer 2 loop generated in the VXLAN segment of a given tenant does not affect the other tenants supported in the fabric. Therefore, you should use a different and unique multicast group for each defined tenant.

With a rate-limiting threshold of 5 percent for multicast traffic, tests demonstrated that hosts belonging to the same broadcast domain (VLAN 100) consumed almost 50 percent of the CPU resources, as shown in Figure 38.

Figure 38. Impact on Virtual Machine CPU Utilization with Different Storm Control Thresholds



CPU/Real-time, 16/04/2015 05:53:26 - 16/04/2015 06:53:26 Chart Options...

Graph refreshes every 20 seconds

Switch to: Default

Performance Chart Legend

Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
■	CentOS-192.168.1.193	Usage in MHz	Average	MHz	0	1156	0	281.486

Figure 38 also shows that with a rate-limiting setting of almost 2 percent, hosts belonging to the affected Layer 2 broadcast domain consumed an average of 25 percent of CPU resources to handle the broadcast storm.

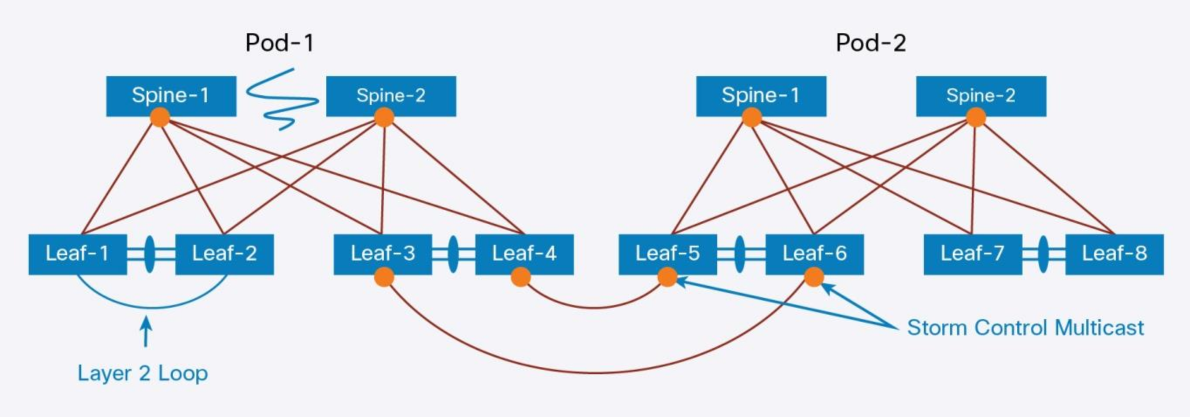
Note: CPU resources are allocated by the hypervisor manager, and consumption also depends on the hardware, memory, and other characteristics of the physical server hosting the virtual machines. Therefore, no single numbers can be recommended for the storm-control setting. Results provided in this section are examples only to show the process that you can follow to find the right values to be applied to a specific deployment.

Applying Storm Control for Loop Mitigation Within a Pod

Another way to mitigate the effects of a Layer 2 loop inside a given pod is to also enable storm-control multicast on the spine interfaces to the leaf nodes, as shown in Figure 39.

Note: If ingress replication (unicast) is used for broadcast, unknown unicast, and multicast traffic, the approach described here does not apply.

Figure 39. Storm Control on the Spine Node Interfaces



As in the interfabric case, you must understand the normal application multicast utilization rate within each pod before you provide a rate-limiting threshold value. This value is not necessarily the same locally and between the fabrics.

Main Points About Broadcast, Unknown Unicast, and Multicast Storm Control for Layer 2 Loop Mitigation

Note the following main points about Layer 2 loop mitigation:

- Multicast storm control aims to rate-limit the use of the network links to a functional bandwidth for user data. However, the CPU resources may be affected by the amount of broadcast traffic that is needs to be handled by the server.
- Hosts that locally belong to the same disrupted broadcast domain are affected by the CPU resource consumption.
- To mitigate network disruption from a broadcast storm, you should allocate a different multicast group for each tenant.
- In modern data centers (with 10 Gigabit Ethernet hosts), you can consider 1 to 2 percent a reasonable ratio for broadcast, unknown unicast, and multicast traffic control. However, it is important to tune the multicast rate limiter. Before setting the rate-limiter value, each enterprise should consider the following:
  - Consider the amount of multicast traffic consumed by some applications (for example, video streaming).
  - In the case of traditional hosts, the rate-limited value for broadcast, unknown unicast, and multicast should not exceed 1 percent based on the vCPU allocation, the physical-to-virtual ratio, and the remaining memory resources available.
- You cannot set a rate limiter per multicast group. It must be set globally at the interface level for all multicast groups.

Conclusion

The deployment of a VXLAN multipod fabric allows you to use the VXLAN technology with an MP-BGP EVPN control plane to extend Layer 2 and Layer 3 connectivity across multiple pods. Depending on the specific use case and requirements, those pods may represent different rooms in the same physical data center location, or separate data center sites (usually deployed at metropolitan-area distances from each other).

The deployment of the multipod design is a logical choice to extend connectivity between fabrics that are managed and operated as a single administrative domain. However, you need to consider that it functions like a single VXLAN fabric. This characteristic has important implications for the overall scalability and resiliency of the design.

A true multisite solution is still the best option for providing separate availability zones across data center locations. The use of specific functions (such as storm control) associated with the multipod design allows you to mitigate as much as possible the propagation of problems across separate pods, making the design a viable multisite option.

For More Information

For additional information, see the following documents:

- VXLAN Network with MP-BGP EVPN Control Plane Design Guide: http://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/guide-c07-734107.html
- Deploy a VXLAN Network with an MP-BGP EVPN Control Plane (white paper): http://www.cisco.com/c/en/us/products/collateral/switches/nexus-7000-series-switches/white-paper-c11-735015.html
- Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide, Release 7.0: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/sw/7-x/vxlan/configuration/guide/b_Cisco_Nexus_9000_Series_NX-OS_VXLAN_Configuration_Guide_7x/b_Cisco_Nexus_9000_Series_NX-OS_VXLAN_Configuration_Guide_7x_chapter_0100.html

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C11-737201-01 06/16