Learn About VXLAN in Virtualized Data Center Networks

Data centers have rapidly increased their server virtualization over the past decade, resulting in dramatic increases in agility. This, in turn, has created a demand for data center networks that are equally flexible and agile. Virtualization of the network is the next obvious step – decoupling the virtual network from the physical network makes it easier to manage, automate, and orchestrate.

One common method being used today to virtualize data center networks is the use of *overlay* networks. An overlay network sits on top of the physical network, enabling the use of a virtual network of switches, routers, firewalls, load balancers, and so on. This decoupling of the virtual from the physical enables fast programmatic provisioning of the network for any application. You no longer have to orchestrate changes across a set of physical devices.

Network overlays are created by encapsulating traffic and tunneling it over the physical network. Although relatively new to data center networks, overlay networks have been used in campus networks for years to carry wireless LAN traffic over the wired network.

Creating a virtual overlay network also benefits the physical, or *underlay*, network, which can now be a simple IP network that is concerned solely with delivering packets to destinations. An overlay network adds simplicity, resiliency, and scale to the physical network, another reason overlay networks are gaining popularity.

Figure 1 illustrates an overlay network. From the perspectives of virtual machine 1 and virtual machine 2 (VM1 and VM2), traffic between them is taking the route shown by the dotted line, going through traditional networking devices such as switches, routers, and firewalls, which are instantiated in the hosts. However, the traffic is actually taking the path shown at the bottom of the figure.

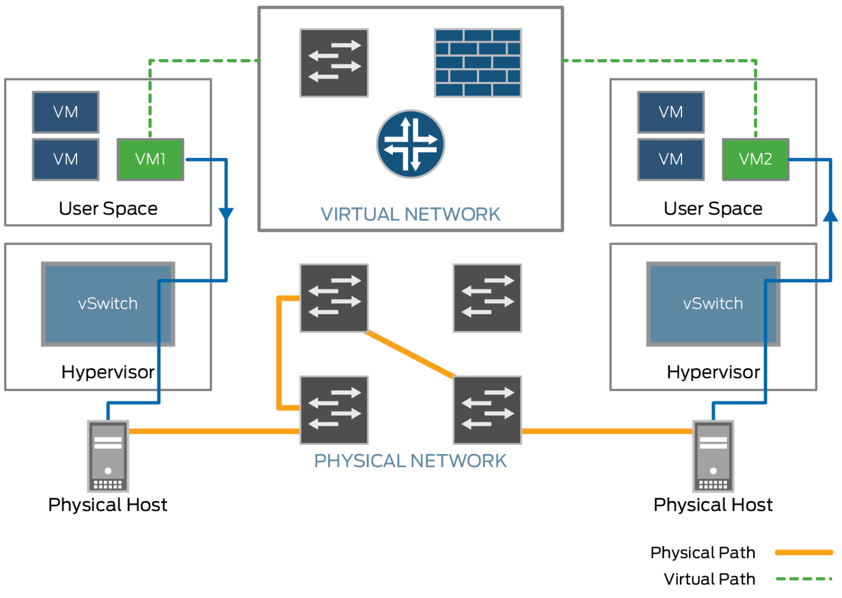

Figure 1 An Overlay Network Traffic Path

A number of tunneling protocols can be used to create overlays in the data center – for example, Network Virtualization using Generic Routing Encapsulation (NVGRE), Stateless Transport Tunneling (STT), and Virtual Extensible LAN (VXLAN). Because VXLAN is the most commonly used, this Learn About focuses on it.

The VXLAN protocol is documented in Internet Engineering Task Force (IETF) RFC 7348. In addition to supporting the virtualization of the data center network, the VXLAN protocol is also designed to address the needs of multi-tenant data centers by providing the necessary segmentation on a large scale.

VXLAN Basics

VXLAN is a tunneling protocol that encapsulates Layer 2 Ethernet frames in Layer 3 UDP packets, enabling you to create virtualized Layer 2 subnets, or segments, that span physical Layer 3 networks. Before we examine the encapsulated packet format, let's explore two key VXLAN concepts:

- VXLAN Network Identifier (VNI)
- VXLAN Tunnel Endpoint (VTEP)

VXLAN Network Identifier

Each Layer 2 subnet or segment is uniquely identified by a VXLAN network identifier (VNI) that segments traffic the same way that an IEEE 802.1Q VLAN ID segments traffic. As is the case with a VLAN, VMs on the same VNI can communicate directly with each other, whereas VMs on different VNIs need a router to communicate with each other.

Although the VNI allows for more than 16 million VXLAN segments, limitations in the networking devices might restrict the total number of segments supported.

The VNI was designed to address the growing needs of multi-tenant data centers. It provides:

- Increased scale – Although the VNI performs a function similar to the VLAN ID, the VNI has one very large advantage over the VLAN ID: the VNI is 24 bits in length, potentially allowing more than 16 million VXLAN segments. The 12-bit VLAN ID provides for only 4094 usable segments. Thus, the VXLAN protocol can support network segmentation at the scale required by cloud builders with large numbers of tenants.
- Increased ease of administration – In a VXLAN deployment, a VM is uniquely identified by the combination of its MAC address and its VNI. Two or more VMs can therefore have the same MAC address as long as they are in different VNIs, which helps simplify the administration of multi-tenant networks.

Virtual Tunnel Endpoint

The entity that performs the encapsulation and decapsulation of packets is called a *VXLAN tunnel endpoint* (VTEP).

VTEPs typically reside in hypervisor hosts, such as VMWare vSphere ESXi hypervisor hosts, or kernel-based virtual machine (KVM) hosts, as shown in Figure 2. Each VTEP has two interfaces. One is a switching interface that faces the VMs in the host and provides communication between VMs on the local LAN segment. The other is an IP interface that faces the Layer 3 network. Each VTEP has a unique IP address that is used for routing the UDP packets between VTEPs.

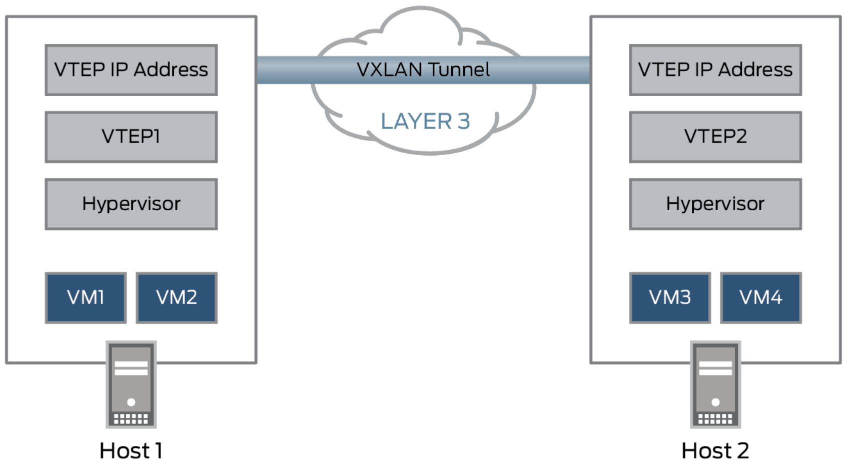

Figure 2 *VTEPs in the Data Center*

As shown in Figure 2, when VTEP1 receives an Ethernet frame from VM1 addressed to VM3, it uses the VNI and the destination MAC to look up in its forwarding table which VTEP to send the packet to. VTEP1 then adds a VXLAN header that contains the VNI to the Ethernet frame, encapsulates the frame in a Layer 3 UDP packet, and routes the packet to VTEP2 over the Layer 3 network.

VTEP2 decapsulates the original Ethernet frame and forwards it to VM3. VM1 and VM3 are completely unaware of the VXLAN tunnel and the Layer 3 network between them.

VXLAN Packet Format

Let's look at the VXLAN packet itself. Figure 3 illustrates the general VXLAN packet format.

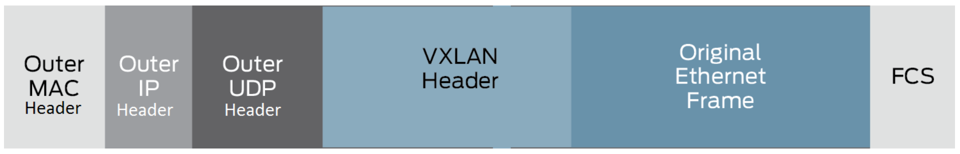

Figure 3 *VXLAN Packet Format*

The key fields for the VXLAN packet in each of the protocol headers are:

- Outer MAC header – Contains the MAC address of the source VTEP and the MAC address of the next-hop router. Each router along the packet's path rewrites this header so that the source address is the router's MAC address and the destination address is the next-hop router's MAC address.
- Outer IP header – Contains the IP addresses of the source and destination VTEPs.
- Outer UDP header – Contains source and destination UDP ports:

The VXLAN repurposing of the source UDP port field enables equal-cost multipath (ECMP) to perform hashes on the field for load-balancing flows across multiple paths. The devices in your underlay network that do ECMP calculations should be configured to use the Layer 4 headers for their calculations.

  - Source UDP port – The VXLAN protocol repurposes this standard field in a UDP packet header. Instead of using this field for the source UDP port, the protocol uses it as a numeric identifier for the particular flow between VTEPs. The VXLAN standard does not define how this number is derived, but the source VTEP usually calculates it from a hash of some combination of fields from the inner Layer 2 packet and the Layer 3 or Layer 4 headers.
  - Destination UDP port – The VXLAN UDP port. The Internet Assigned Numbers Authority (IANA) allocates port 4789 to VXLAN.
- VXLAN header – Contains the 24-bit VNI.
- Original Ethernet Frame – Contains the original Layer 2 Ethernet frame.

In total, VXLAN encapsulation adds between 50 and 54 bytes of additional header information to the original Ethernet frame. Because this can result in Ethernet frames that exceed the default 1514 byte MTU, best practice is to implement jumbo frames throughout the network.

Connecting to the Nonvirtual World

So far we've looked at how virtual machines running in a host with a virtual switch can communicate over a VXLAN overlay. In the real world, however, very few data centers have all of their servers virtualized. Nonvirtualized, or *bare-metal, servers* still exist – for example, non-x86 servers (UNIX and mainframe devices), storage (NAS, iSCSI SANs), and certain database and high-performance compute instances. These devices typically do not support VXLAN and need to continue to reside on VLAN segments. *So how can nonvirtualized devices be connected to the virtualized network?*

One way is to use gateways at the network edge that act as VTEPs, as shown in Figure 4. VTEP gateways map VLANs to VXLANs and handle the VXLAN encapsulation and decapsulation so that the nonvirtualized resources do not need to support the VXLAN protocol. This permits the VXLAN and VLAN segments to act as one forwarding domain across the Layer 3 boundary. For example, as far as Physical Server 1 in Figure 4 is concerned, VM1 and VM2 appear to be residing in the same VLAN segment in which it resides, VLAN 10.

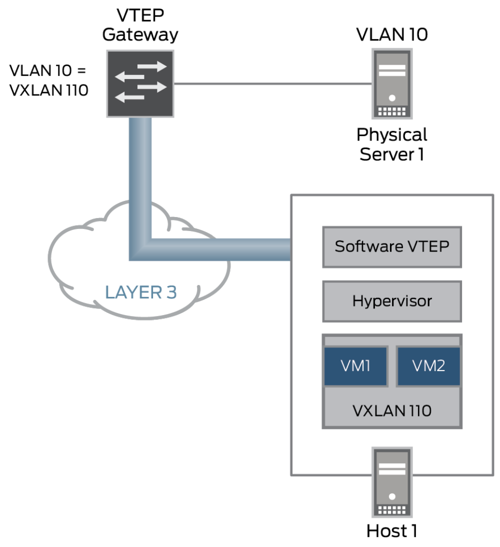

Figure 4 Hardware VTEP Gateway

A software appliance – for example, a virtual switch instance running on standard x86 hardware – can act as a VTEP gateway. Your data center, however, might require more predictability and throughput than can be achieved through software appliances. To meet this requirement, switches that function as VTEP gateways, often referred to as *hardware VTEP gateways*, are available.

For example, Juniper Networks MX Series routers, QFX5100 switches, and EX9200 switches can all act as VTEP gateways, encapsulating/decapsulating VXLAN packets on behalf of bare-metal servers. In addition, the MX Series routers and EX9200 switches can route between different VXLANs. You can configure bridge domains on these devices, associate them with VNIs, and then configure integrated routing and bridging (IRB) interfaces on bridge domains.

MAC Learning – Data Plane Versus Control Plane

To encapsulate and forward traffic on behalf of its local endpoints, a VTEP must maintain a table that maps the MAC addresses of VMs and bare-metal servers to specific VTEPs in the network. To create and maintain the mappings, VTEPs must learn about the MAC and VTEP addresses in the network. VXLAN deployments support two ways of learning about addresses: MAC learning in the data plane and MAC learning in the control plane.

MAC Learning in the Data Plane

The current VXLAN specification does not include a control plane to provide a mechanism for VTEPs to share what they have discovered about addresses in the network. The specification does, however, describe a mechanism for each VTEP to discover on its own what addresses are on the network through monitoring the packets on the data plane.

This mechanism is similar to the way Ethernet switches learn about MAC addresses: whenever a VTEP receives a VXLAN packet, it records the IP address of the source VTEP, the MAC address of the VM, and the VNI in its VXLAN forwarding table. In the future, when the VTEP receives an Ethernet frame to forward to that MAC address on that VNI segment, it knows to encapsulate that frame in a VXLAN packet addressed to that VTEP.

What happens if a VTEP receives an Ethernet frame destined for a VM for which it doesn't know the MAC address? Again, the VXLAN process used for handling unknown unicast traffic is similar to the way Ethernet switches handle unknown unicast traffic – the VTEP floods the traffic. To control unnecessary flooding, each VNI is assigned to a multicast group. When a VTEP receives an unknown unicast packet on a VNI, it floods the packet to all VTEPs in the VNI's multicast group.

As an example, suppose VM1 in Figure 5 wants to send a packet to VM2 at IP address 192.168.0.11 but does not have VM2's MAC address. Both virtual machines reside in VNI 100, which has been assigned to multicast group 239.1.1.100.

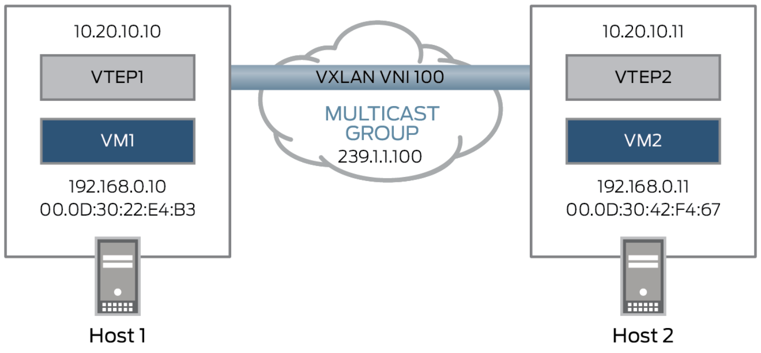

Figure 5 Unknown Multicast Handling Example

The process for handing this unknown unicast packet is as follows:

1. VM1 sends an ARP packet requesting the MAC address associated with 192.168.0.11.
2. VTEP1 receives this ARP packet. It encapsulates this ARP request into a multicast packet and addresses the packet to the multicast group 239.1.1.100.
3. All VTEPs in multicast group 239.1.1.100 receive the packet. They decapsulate it and check the VNI in the VXLAN header. If the VNI for a local VXLAN segment is 100, the VTEPs forward the original ARP packet to that VXLAN segment. Otherwise, they drop the packet.

   The VTEPs also add the mapping of the IP address of VTEP1 to the MAC address of VM1 to their local VXLAN tables:

VNI	MAC Address	VTEP Address
100	00:0D:30:22:E4:B3	10.20.10.10

4. When VM2 receives the ARP packet from VTEP2, it responds with its MAC address.
5. VTEP2 encapsulates the response in a unicast IP packet and sends it to VTEP1.
6. VTEP1 receives the ARP packet, decapsulates it, and passes it on to VM1. VTEP1 now stores the mapping of the VTEP2 IP address and the VM2 MAC address in its VXLAN mapping table:

VNI	MAC Address	VTEP Address
100	00:0D:30:42:F4:67	10.20.10.11

At this point, all relevant MAC addresses have been learned, and VM1 and VM2 can communicate directly via unicast in the future.

Typically, to reduce unnecessary flooding of packets, administrators assign each VNI to its own multicast group. However, there is no requirement that each VNI have its own multicast group; you can assign multiple VNIs to a single multicast group. Isolation of the VXLAN segments is maintained in this case because the VTEP always checks the VNI before it forwards decapsulated packets to a VXLAN segment.

MAC Learning in the Control Plane

Although the multicast flooding mechanism for learning MAC addresses works, it isn't efficient and doesn't scale well. There are more possible VNIs than there are multicast groups, so at some point it can become necessary to assign multiple VNIs to the same multicast group, which increases inefficiency. In addition, the use of multicast requires running PIM and IGMP in the underlay network and creates administrative overhead in managing IGMP groups.

To build a more robust and scalable overlay network, many enterprises have turned to implementing control planes for VXLAN. Two of the most commonly implemented control planes are:

- Open vSwitch Database Management Protocol (OVSDB)
- Ethernet VPN (EVPN)

OVSDB

OVSDB is an OpenFlow configuration protocol designed to manage Open vSwitch implementations. OpenFlow provides the control plane, managing flows and determining how packets are forwarded between source and destination VMs. OVSDB also provides a management plane that allows for a standard method to configure and manage switches, even those from different vendors.

OVSDB is usually coupled with an overlay controller, such as VMware NSX. An overlay controller uses OVSDB to provision the VTEPs and to handle MAC address learning. The controller provides the VTEPs with MAC addresses and the VXLAN tunnels over which the addresses can be reached. In return, the VTEPs inform the controller of any MAC addresses they learn. This centralized replication of MAC addresses is much more efficient than MAC learning through multicast.

OVSDB is documented in RFC 7047, Open vSwitch Database Management Protocol, which is an IETF informational RFC.

OVSDB support is not limited to virtual switches. A hardware VTEP gateway that supports OVSDB can participate in a controller-based overlay network such as VMware NSX for Multi-Hypervisor (NSX-MH). Juniper Networks MX Series routers, QFX5100 switches, and EX9200 switches support OVSDB. Because they support OVSDB, you can use the NSX controller to automate the creation of VXLAN tunnels on these devices.

EVPN

EVPN, a recently approved IETF standard, is a control-plane technology that has been decoupled from the data plane, allowing it to be used with different data planes. EVPN uses Multiprotocol BGP (MP-BGP) for MAC and IP address distribution, with MAC addresses being treated as routes in the BGP table. Route entries can contain just a MAC address or a MAC address plus an IP address (ARP entry). With EVPN, VTEPs interested in specific VNIs can exchange MAC addresses with other VTEPs.

An example of an overlay controller that uses EVPN as the control plane is Juniper Networks Contrail. Contrail vRouters reside in hypervisors and use EVPN to exchange learned MAC addresses with each other and the controller. To support bare-metal servers, Contrail also uses OVSDB to exchange learned MAC address information with VTEP gateways, which might not support EVPN.

Juniper Networks pioneered the development of EVPN. It also participated in the IETF committee that published the first EVPN standard in February 2015, RFC 7432, BGP MPLS-Based Ethernet VPN.

It is also possible to have a controller-less implementation that uses EVPN as a control plane. To do this, the VTEPs – both software and hardware – must support the use of EVPN.

A draft IETF standard exists for using for using EVPN with VXLAN (and other virtualization overlay protocols). Juniper Networks is actively involved in the committee developing the standard, A Network Virtualization Overlay Solution using EVPN. Vendors, including Juniper Networks and Cisco Systems, are in the process of building support for EVPN in VTEPs.

Physical Network Requirements

In a VXLAN overlay implementation, the core requirement for the underlying physical network is that it be a routed, Layer 3 network. Building a data center network on top of a pure Layer 3 network has these benefits:

- It has to support a single routing protocol only, rendering the network more stable.
- You do not need to rely on STP to converge the topology – the routing protocol takes care of that instead. Without STP, no links are blocked and every link can be used to carry traffic.
- You can take advantage of ECMP to load-balance traffic across all active links.

When building your Layer 3 underlay network, you want to use an architecture that provides predictable performance and that scales linearly. A Clos (spine-leaf) architecture provides a flat and easily scalable architecture.

Juniper Networks offers the following fabric architectures for the underlay network:

- Virtual Chassis Fabric (VCF) – VCF is a low-latency, high-performance fabric architecture that you manage as a single switch. You can create a VCF using a spine-and-leaf topology of up to 32 QFX Series switches, with a maximum of 4 spine switches. It allows you to build a fabric for VXLAN tunnel overlays without having to worry about BGP and PIM. Because the VCF behaves like a single logical switch, you can create integrated routing and bridging (IRB) interfaces and all traffic is automatically routed between all networks, because each network appears as a directly connected network. To enable multicast on a VCF, you enable Internet Group Management Protocol (IGMP) on any interfaces requiring multicast. There is no need to design and configure PIM or any other multicast protocols.

Juniper Networks has published a network configuration example based on an architecture that uses VCF with VMware NSX. See MetaFabric Architecture 2.0: Configuring Virtual Chassis Fabric and VMware NSX.

- IP fabric – For larger data centers whose scaling requirements exceed VCF, you can build a spine-and-leaf IP fabric using QFX5100 switches, with a maximum of eight spine switches.
- Junos Fusion – A new fabric technology, Junos Fusion is a single-tier architecture that allows a data center network to be managed as a single system. Junos Fusion comprises two main components – aggregation devices and satellite devices – that work together as a single switching system, flattening the network without compromising resiliency. You manage a Junos Fusion fabric through the aggregation devices, which can control up to 128 satellite devices.

For more information on these architectures, see http://www.juniper.net/us/en/products-services/switching/data-center-switching-architectures/.

Stretching VXLAN between Data Centers

Traditionally, enterprises with multiple data centers have used Layer 3 solutions to connect the data centers. However, enterprises might want to stretch VXLAN tunnels between data centers – for example, to support live VM mobility between data centers.

VXLAN was not designed to be a Data Center Interconnect (DCI) solution by itself. However, there are a number of Layer 2 DCI solutions available that could enable stretching VXLAN between data centers – for example, Ethernet over MPLS (EoMPLS), Ethernet over MPLS over GRE (EoMPLSoGRE), or virtual private LAN service (VPLS).

Finally, there is the newest Layer 2 VPN – EVPN. As described earlier, EVPN can be used with different data planes, including MPLS and VXLAN. As a DCI solution, EPVN capitalizes on the industry's experience with VPLS and provides these advantages over other Layer 2 VPNs:

- MAC address learning in the control plane. Instead of flooding unknown unicast, EVPN uses MP-BGP for MAC address learning. PE routers exchange Layer 2 MAC address and Layer 3 IP address information using MP-BGP.
- Advanced multihoming capabilities with active-active redundancy.
- Integration of Layer 3 routing on the same interface or VLAN.
- Support for VM migration. EVPN's MP-BGP control plane supports MAC mobility, which enables the PE routers to track the movement of a VM's MAC address. Thus, the PE routers always have current reachability information for the MAC address.

  In addition, EVPN incorporates mechanisms to prevent traffic being "tromboned" to a VM's previous default gateway when the VM moves across data centers.

Juniper Networks has integrated VXLAN and EVPN on the MX Series routers, providing the ability to use EVPN to stitch together VXLAN tunnels from one data center to another, as shown in Figure 6.

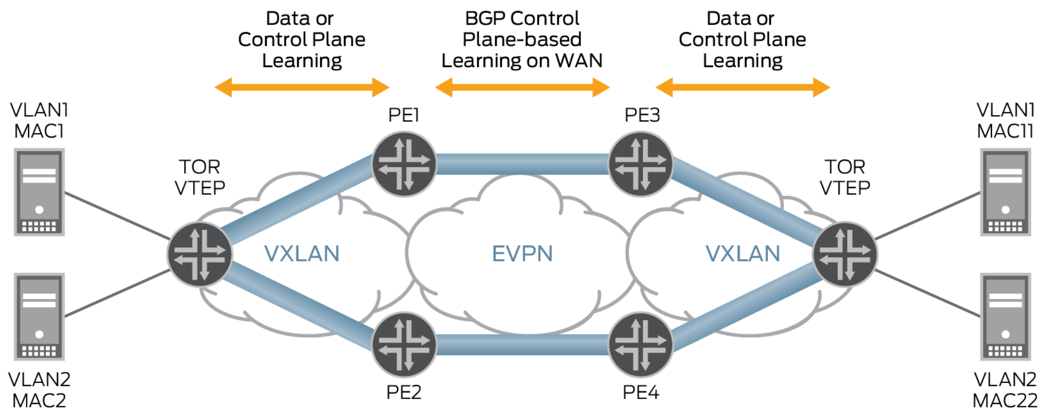
Data or Control Plane Learning
BGP Control Plane-based Learning on WAN
Data or Control Plane Learning
VLAN1 MAC1
VLAN2 MAC2
TOR VTEP
PE1
PE2
PE3
PE4
VXLAN
EVPN
VXLAN
TOR VTEP
VLAN1 MAC11
VLAN2 MAC22

Figure 6 Stretching VXLAN Across Data Centers with EVPN

Correlating the Virtual with the Physical

Although virtualizing your data center network through VXLAN overlays provides great benefits, it also presents some challenges, particularly when it comes to troubleshooting application performance issues. Server administrators, who in most data centers are in charge of the overlay network, can have limited visibility into what is happening in the underlying physical network. And network administrators, for their part, often have limited visibility into the application flows on the network. Yet, being able to correlate how events in the physical network affect particular application flows in the virtual network is critical to identifying and resolving problems in a speedy and cost-effective manner.

Juniper Networks provides the following solutions for visualization, analysis, and correlation of the network physical and virtual layers:

- Cloud Analytics Engine – An integrated analytics solution designed for Juniper-based networks. It provides visibility into all devices within a network's physical and virtual layers. It includes data collection, analysis, and correlation to help you better understand the behavior of workloads and applications across the physical and virtual infrastructure. It is an open solution that supports RESTful APIs for third-party integration with any standards-based database used to process real-time network performance data.
  See http://www.juniper.net/us/en/products-services/network-management/cloud-analytics-engine/.
- Junos Space Network Director – A network management solution that enables administrators to visualize, analyze, and control the entire enterprise network, from the data center to the campus, physical to virtual, and wired to wireless, all through a single management screen.
  See http://www.juniper.net/us/en/products-services/network-management/junos-space-applications/network-director/.
- In release 2.0, Network Director integrates with OpenStack with VMware NSX-MH to provide full visibility into the overlay and underlay networks. You can view the hypervisor servers, bare-metal servers, VMs, and network devices and their connectivity, as well as any associated VTEPs and VNIs. Network Director also integrates with the Cloud Analytics Engine to provide application flow analysis, allowing you to view the path an application flow takes through the physical network and to examine information such as the latency at each hop, the traffic statistics for devices in the path, and alarm status for devices in the path.
  See http://www.juniper.net/techpubs/en_US/junos-space-apps/network-director2.0/information-products/pathway-pages/index.html.

Further Reading

OVSDB and VXLAN Feature Guide
http://www.juniper.net/techpubs/en_US/junos14.1/information-products/pathway-pages/junos-sdn/ovsdb-vxlan.html

Contrail Documentation
http://www.juniper.net/techpubs/en_US/release-independent/contrail/information-products/pathway-pages/index.html

Day One: Using Ethernet VPNs for Data Center Interconnect
http://www.juniper.net/us/en/training/jnbooks/day-one/proof-concept-labs/using-ethernet-vpns/

Virtual Chassis Fabric
http://www.juniper.net/techpubs/en_US/junos14.1/information-products/pathway-pages/qfx-series/virtual-chassis-fabric.html

White Paper: Clos IP Fabrics with QFX5100 Switches
http://www.juniper.net/us/en/local/pdf/whitepapers/2000565-en.pdf

Cloud Analytics Engine
http://www.juniper.net/techpubs/en_US/junos14.1/information-products/pathway-pages/qfx-series/cloud-analytics-engine.html

Network Director documentation
http://www.juniper.net/techpubs/en_US/junos-space-apps/network-director2.0/information-products/pathway-pages/index.html

The *Juniper QFX5100 Series*, by Doug Hanks. Published by O'Reilly Media
http://www.juniper.net/us/en/training/jnbooks/qfx5100.page

Learn About VXLAN in Virtualized Data Center Networks

by Brooke Doverspike

Data center networks are being challenged to become more flexible and agile in order to increase cost effectiveness and gain competitive advantage. Enterprises are turning to network virtualization in the data center in response to this challenge. Learn about VXLAN and the role it plays in virtualizing data center networks.

Brooke Doverspike is a Staff Engineering Technical Writer at Juniper Networks with more than 30 years of experience developing networking, security, programming, and systems documentation.

For more information see:
juniper.net/documentation

ISBN: 978-1-941441-13-8 Version History: First Edition, April 2015 2 3 4 5 6 7 8 9

ISBN 978-1941441138

50200

9 781941 441138