VXLAN Technical Brief

A standard based Data Center Interconnection solution

Dell EMC Networking – Data Center Technical Marketing

February 2017

Table of contents

Introduction

In the information technology field often when a new technology is created to address a need, an enhancement or complementary technology is created.

The best example of this is *virtualization* (network and resources) in the data center. Prior to virtualization, traditional network segmentation was provided through VLANs (Virtual Local Area Networks) where a set of hosts or a single host would be assigned to a different VLAN ID in order to segment them from each other. If these hosts wanted to communicate with each other; inter-vlan routing would be required.

There are inherent shortcomings when using VLANs, such as inefficient network inter-links, spanning tree limitations, physical location of devices, limited number of VLANs (4,094), multi-tenant environments, and ToR (Top of Rack) switch scalability. VLANs have become a limiting factor for IT departments and providers as they look to build efficient and highly scalable multitenant data centers where *server virtualization* is a key critical component.

Spanning Tree and VLAN Range Limitations

A typical data center uses a Layer 2 design to facilitate inter-VM communication. This usually results in the presence of the spanning tree protocol to avoid network loops due to duplicate network paths. Spanning tree renders half of the data center network fabric useless since it blocks half of the links to avoid network traffic replication that could cause looping of frames. This increases TCO (Total Cost Ownership), effectively paying for ports that will not be used.

In addition to spanning tree, the typical Layer 2 network uses VLANs to provide basic broadcast isolation. The traditional VLAN implementation uses a 12-bit field ID used to separate a large layer 2 domain into separate broadcast domains. This allows for a VLAN ID range of 1 – 4,094. For a small data center environment, this range is usually not an issue; however, with the increased virtualization adoption rate in today's data center and multi-tenant requirements the traditional VLAN range simply cannot scale.

Multi-tenancy

It wasn't long ago when the word cloud computing was just a "buzz" word or a "concept" found in niche areas around service providers willing to try out new bleeding edge technologies. Today, cloud computing is as normal of a concept to most enterprise organizations as the internet is to the average individual. It provides a list of benefits that when properly leveraged becomes a valued asset.

One of the primary drivers of cloud computing is the concept of "***elasticitic***y". This on-demand elastic provisioning - addition and or removal - of resources for multi-tenant applications makes cloud computing an opportunity for organizations to offload their IT infrastructure to a cloud service provider. Unfortunately, the use of traditional VLANs creates a scalability problem due to the typically large number of tenants supported within the same data center. Since VLANs are used to isolate traffic between the different tenants, limitation of 4,094 VLAN IDs often inadequate.

Although Layer 3 networks can sometimes be leveraged to address the VLAN range limitation, it is not atypical to find two different tenants requiring the same Layer 3 ip addressing scheme due to legacy applications, network design requirements, or inherited requirements.

The IEEE has proposed potential initiatives such as TRILL (Transparent Interconnection of Lots of Links), or SPB (Shortest Path Bridging) to address traditional spanning tree shortcomings. However these initiatives have not been widely adopted.

What can be done to resolve these issues? Enter VXLAN (Virtual Extensible LAN), a standard officially documented by the IETF as an open standards solution to resolve these shortcomings.

Dell EMC with its data center networking product portfolio switches, such as the fixed-configuration Dell EMC S6000 Series, S4048-ON, Z9100-ON, and S6100-ON have been designed for the next-generation data center with hardware-based VXLAN functionality.

HW-based VXLAN extend Layer 2 connectivity across a Layer 3 boundary while keeping seamless integration between VXLAN and non-VXLAN environments. Together, these switches form the physical network underlay building blocks of a scalable virtualized and multitenant data center.

Objective

The objective of this short technical brief document is to provide an overview of VXLAN, its components, and Dell EMC's architecture and solutions.

VXLAN Overview RFC 7348 – Why?

If VXLAN is yet another important technology introduced that extends and enhances the traditional 802.1Q VLAN feature set, how does VXLAN add more value or becomes a powerful tool besides enhancing what 802.1Q VLAN provides?

The answers to the questions lay in the following problem statement: "How can I as a network administrator or engineer stretch my Layer 2 reach across multiple data centers and not worry about running out of vlan IDs or using duplicate vlan IDs?"

VXLAN answers the questions through the use of what is referred as "encapsulation" running over an existing networking infrastructure (Layer 3) to provide a means of stretching a Layer 2 network across different data centers. The RFC defines VXLAN "…***as a Layer 2 overlay scheme on a Layer 3 network***..." where each overlay is referred as a VXLAN segment and each segment is identified through a 24-bit segment ID referred to as a VNI (VXLAN Network Identifier). This allows potentially up to 16 Million VXLAN segment IDs far greater than the traditional 4,094 VLAN IDs.

There are five key elements used within VXLAN as part of its implementation:

- VNI (VXLAN Network Identifier)
- VTEP (VXLAN Tunnel End Point)
- Multicast support – IGMP and PIM
- Layer 3 routing protocol – OSPF, BGP, IS-IS
- QoS

VNI – VXLAN Network Identifier

The VNI is part of the outer header that encapsulates the inner MAC frame sourced by the VM. The VNI is a unique combination of host or VM MAC SOURCE ADDRESS and VXLAN segment ID that identifies a single VM or physical host. It is so unique that it allows for duplicate source MAC addresses to coexist in the same administrative domain. For example, a VM with source MAC_A and VXLAN segment ID 5000 is completely different from a VM or physical host with source MAC_A and VXLAN segment ID 5001.

VTEP – VXLAN Tunnel End Point

VXLAN as an "encapsulation" scheme is also referred as a "tunneling" scheme and like any tunneling implementation there has to be a starting and end point to the tunnel. To this end VXLAN implements a VTEP in software or hardware. The VTEP uses information from the outer IP header to route or switch traffic from the VMs or physical host. (See Figure 1)

Multicast

Multicast is needed whenever a VM or host attempts to communicate with another VM or host over Layer 3 (IP) or Layer 2. Typically, if the destination VM or host is in the same subnet the source VM or host will send out an ARP broadcast packet. In a non-VXLAN environment, this frame is simply broadcast across all devices carrying that VLAN.

In a VXLAN environment, a broadcast ARP packet is sent to a multicast group address. A VXLAN VNI is mapped to this multicast address group. This mapping is then sent to each VTEP. Once the VTEP has this mapping it is used to provide IGMP membership reports to the upstream switch/router to join/leave the specific multicast group of interest by the VMs or hosts.

The infrastructure should be able to meet the multicast requirements, whether it is Layer 3 such as Protocol Independent Multicast – Sparse Mode (PIM-SM) or Layer 2 such as IGMP Snooping.

Layer 3 routing protocol

As an encapsulation and tunneling scheme, VXLAN is a layer 2 scheme over a layer 3 infrastructure. As a result basic layer 3 routing protocols such as BGP, OSPF, or IS-IS are part of any VXLAN deployment.

Quality of Service (QoS)

As VXLAN packets traverse the Layer 3 domain carrying with them specific types of traffic/applications and specific level of service, it is imperative that these level of services are not only honored but translated onto the VXLAN IP outer header.

This level of service translate as DSCP values where the external physical network infrastructure uses to prioritize the traffic based on the DSCP setting on the IP outer header.

VXLAN Encapsulation and Packet Format

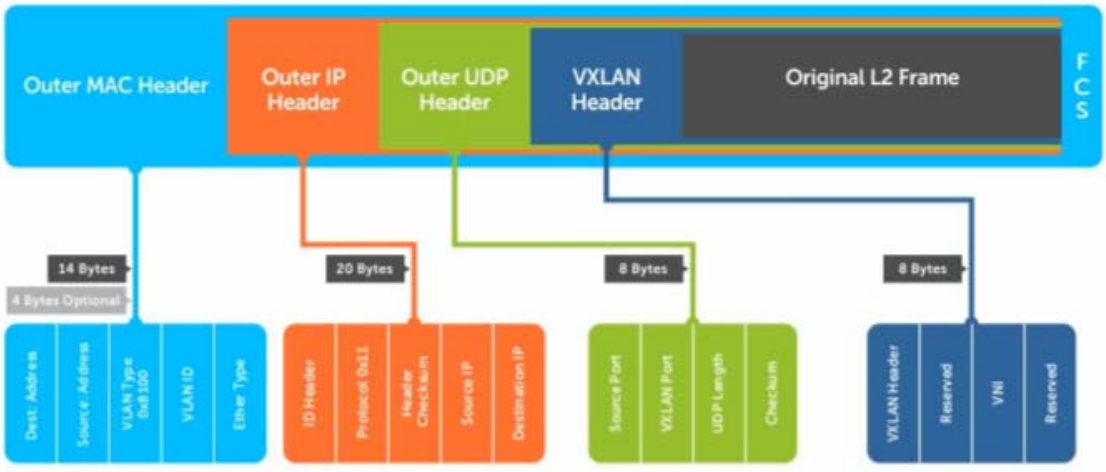
VXLAN packets are encapsulated over a UDP packet through the use of MAC Address-in-User Datagram Protocol (MAC-in-UDP) and tunneled across different data centers using virtual end point tunnels which will be discussed later on in the document.

The transport protocol used is IP plus UDP.

As noted before, VXLAN defines a MAC-in-UDP encapsulation format where the original Layer 2 frame from the source VM has a VXLAN header added and is then encapsulated again in a UDP-IP packet. Figure 1, shows the entire VXLAN Frame format and the related encapsulations over the original Layer 2 frame. Notice how each header is used by the different components as the packet traverses from source to destination.

This section breaks down the individual headers that make up the entire VXLAN packet and their respective components.

Figure 1 VXLAN Frame Format



Components of VXLAN Frame Format

Some of the important fields of the VXLAN frame format are:

Outer Ethernet Header/Outer MAC Header: The Outer Ethernet Header consists of the following:

- **Destination Address:** Generally, it is a first hop router's MAC address when the VTEP is on a different address.
- **Source Address:** It is the source MAC address of the router that routes the packet.
- **VLAN Type and ID:** It is optional in a VXLAN implementation and will be designated by an ethertype of 0x8100 and has an associated VLAN ID tag.
- **Ethertype:** It is set to 0x8100 because the payload packet is an IPv4 packet. The initial VXLAN draft does not include an IPv6 implementation, but it is planned for the next draft.

Outer IP Header: The Outer IP Header consists of the following components:

- **Protocol:** It is set to 0x11 to indicate that the frame contains a UDP packet.
- **Source IP:** It is the IP address of originating VTEP over which the communication VM is running.
- **Destination IP:** It is the IP address of the target VTEP. This address can be unicast or multicast. A unicast address represents the destination VTEP. A multicast address

represents the VXLAN VNI and the IP multicast group mapping for broadcast communication between VMs or hosts in different domains.

Outer UDP Header: The Outer UDP Header consists of the following components:

- **Source Port:** Entropy of the inner frame. The entropy or variability scheme could be based on the inner L2 header or inner L3 header. This value or source port number is to enable a level of uncertainty when it comes to load balancing of VM-to-VM traffic across the VXLAN overlay.
- **VXLAN Port:** IANA-assigned VXLAN Port (4789).
- **UDP Checksum:** The UDP checksum field is transmitted as zero. When a packet is received with a UDP checksum of zero, it is accepted for de-capsulation.

VXLAN Header: The VXLAN Header consists of the following components:

- **VXLAN Flags:** Reserved bits (8bits), set to zero except bit 3, the first bit is set to 1 for a valid VNI.
- **VNI:** The 24-bit field that is the VXLAN Network Identifier.
- **Reserved:** A set of fields, 24 bits and 8 bits that are reserved and set to zero.

Frame Check Sequence (FCS): The original Ethernet frame's FCS is not included, but new FCS is generated on the outer Ethernet frame.

Feature Comparison – 802.1Q VLAN vs. VXLAN

Table 1, shows the enhancements between the typical VLAN and VXLAN feature set. Each of the entries should be taken into consideration whenever deploying VXLAN in the network.

Table 1 802.1 Q VLAN and VXLAN Comparison

Feature and Scaling	802.1Q VLAN	VXLAN
Max ID Range	4,000 limited by spanning-tree scaling	16M, limited by the number of multicast groups supported by network devices
Packet Size	1.5K or 9K, some devices support up to 12K	50 additional bytes for VXLAN header
Multicast Requirements	None	PIM, SM, DM, or Bi-dir. Many VNIs to a single multicast group mapping supported
Routing Support	Any Layer 2 or Layer 3 capable device	Any Layer 3 or Layer 2 compatible with VMware vShield, vEdge, and any VTEP capable networking device such as the Dell EMC S6000, S4048-ON, and S6100-ON
ARP Cache	Limits the VMs supported per VLAN	Cache size on VMware or VTEP devices limits VMs supported per VNI
Duplicate IP across different logical segments	N/A	Yes

Duplicate MAC address across logical segments	N/A	Yes
Duplicate VLAN IDs across different logical segments	N/A	Yes

VTEP (VXLAN Tunnel Endpoint) Overview

So far, we have answered the question of why VXLAN is needed but we have not answered why it is also considered a tunneling scheme?

VXLAN as noted before is an encapsulation and tunneling scheme and as such an entity to encapsulate/decapsulate and originate/terminate the tunnel is needed. The encapsulation requirement has been discussed in the VXLAN overview section and now we need to discuss how this encapsulated traffic is handled.

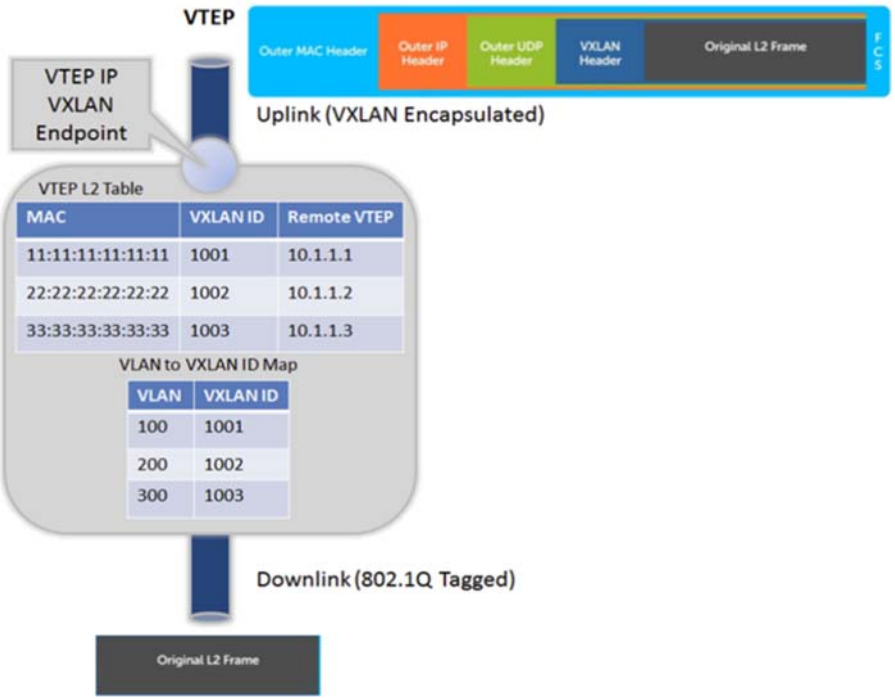
Most of today's data center consists of a mixture of virtualized and non-virtualized compute resources. We also know that these virtualized environments MUST be able to communicate with non-virtualized environments. It is not possible to have a homogeneous data center from the applications, resources, or infrastructure point of view.

In a typical data center, a virtualized server has several VMs and they communicate with each other through what is called a vSwitch (virtual Switch). This vSwitch is the first hop for all the VMs and implements the network virtualization part of a virtual environment. Unfortunately, this vSwitch construct does not exist on a non-virtualized server or bare-metal server and therefore establishing any sort of communication between a virtualized and non-virtualized compute resource is not possible unless some type of appliance or gateway device can serve as a bridge between these two different environments.

To bridge this gap, the VXLAN RFC introduced the concept of a VXLAN Tunnel Endpoint or VTEP. This entity has two logical interfaces: an uplink and downlink. The uplink interface receives VXLAN frames and acts as the tunnel endpoint with an IP address used for routing VXLAN encapsulated frames. The IP address assigned to this uplink interface is part of the network infrastructure and completely separate from the VMs or tenants IP addressing using the VXLAN fabric.

Packets received on the uplink interface are mapped from the VXLAN ID to a VLAN and the Ethernet payload is then sent as a typical 802.1Q Ethernet frame on the downlink to the final destination. As this packet passes through the VTEP a local table is created with the inner MAC SOURCE ADDRESS and VXLAN ID. Packets that arrive at the downlink interface are used to create a VXLAN ID to regular VLAN map.

Figure 2 VTEP Function Diagram



There are two types of VTEPs that have been introduced in the industry, a software and hardware based VTEP Gateway:

Software Based VTEP Gateway

The software based VTEP runs as a separate appliance and it typically runs on a standard x86 hardware with an instance of Open vSwitch. This VTEP is under a controller and it maps physical ports and VLANs on those ports to logical networks. Through this map, VMs that are part of the same logical network can now communicate with the physical device that belongs to the same logical network. This is the approach taken by several overlay architecture such as VMware NSX-v, Midokura, and others.

Software based VTEPs are a great solution for fairly moderate amounts of traffic between VMs and physical devices.

Hardware Based VTEP Gateway

When a full rack of physical servers running database applications need to connect to logical networks containing VMs, ideally these bare metal servers will need a high-density and high-performance switch that could bridge/switch these traffic patterns between the physical servers and logical segments. This is where hardware based VTEPs make sense – migration and interoperability of legacy applications. The hardware based VTEP is no different than the software based VTEP in terms of functionality and controller interaction. A controller which has visibility into the virtualized environment is used to integrate the hardware VTEP and thus creates the gateway functionality required between the virtual and non-virtual environments.

With hardware-based VTEP, the VTEP functionality is implemented in the hardware ASIC resulting in a performance and scalability advantage.

There are two types of VTEP functionalities: Layer 2 and Layer 3. As a Layer 2 VTEP gateway, it provides both encapsulation and de-capsulation of legacy or traditional Ethernet and VXLAN packets respectively and allows the stretch of a Layer 2 domain across a Layer 3 domain. Tagged packets configured with a VLAN ID are mapped to a respective VNI entry encapsulated with a VXLAN header.

As a Layer 2 VTEP gateway, VMs or hosts from VXLAN segment X cannot communicate with a different VM or host in VXLAN segment Y in order for this communication to take place, a router or Layer 3 device is needed and all routing functions will be performed based on the outer most IP header (see VXLAN Packet Format).

As a Layer 3 VTEP gateway, it performs straight VXLAN segment X to VXLAN segment Y routing similar to the traditional inter-VLAN routing providing communication between different VXLAN segments. Contrary to how a Layer 2 VTEP gateway functions (VXLAN to VLAN mapping), a Layer 3 VTEP gateway, performs routing using the VXLAN ID, there is no mapping dependency.

VXLAN routing is currently not widely supported with the current silicon. Currently, the most common type of a VTEP GW deployment is as a Layer 2 VTEP gateway with VXLAN routing support to take place with later software releases and newer silicon ASICs.

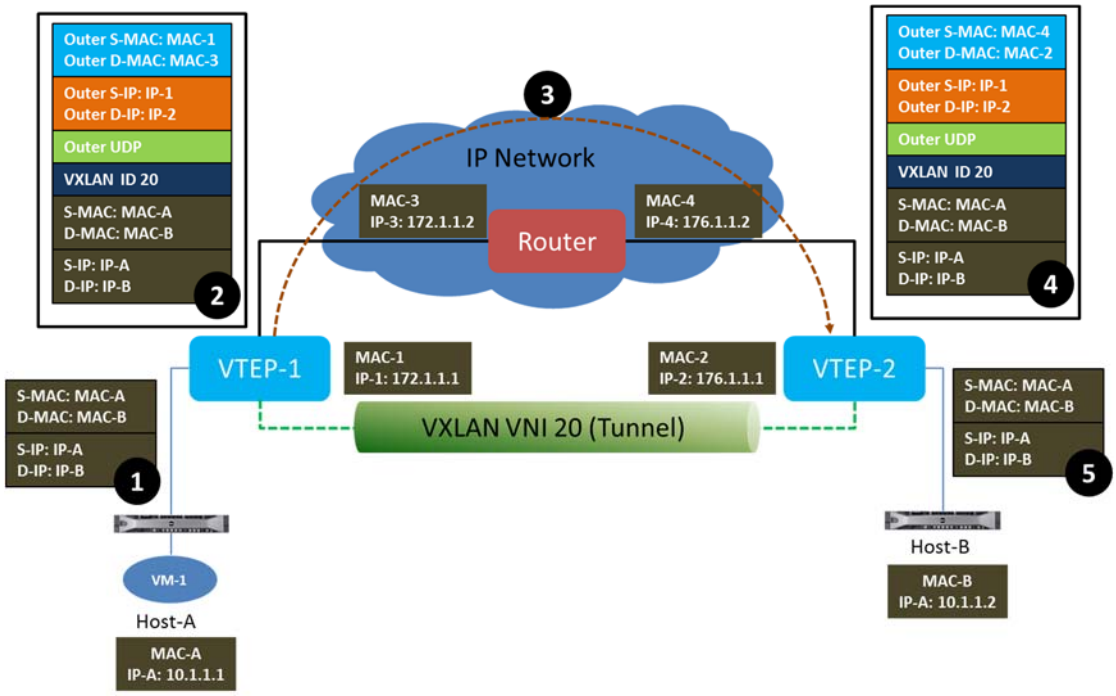
Table 2 Software vs. Hardware VTEP comparison

Key Points	Software VTEP	Hardware VTEP
Virtual, physical, or both	Virtual	Physical
Scalability	Moderate traffic between VMs	Moderate to heavy traffic between VMs and physical servers
Orchestration	Controller based	Controller and non-controller based
Performance	CPU driven	ASIC driven (Line rate)

VTEP Packet Forwarding Flow

Figure 3, shows an example of a VXLAN packet forwarding flow. Notice how the outer headers are used at the respective phases as the packet flows from VTEP 1 to VTEP 2.

Figure 3 VXLAN Unicast Packet Flow



Host-A and Host-B belong to the same VXLAN segment ID 20 and communicate with each other through the VXLAN tunnel created between VTEP-1 and VTEP-2.

Step 1 – Host-A sends traffic to Host-B, it creates an Ethernet frame with Host-B's destination MAC and IP address and sends it towards VTEP-1.

Step 2 – VTEP-1 upon receiving this Ethernet frame has a map of Host-B's MAC address to VTEP-2. VTEP-1 performs the VXLAN encapsulations by adding the respective headers such as VXLAN, UDP, and outer IP headers.

Step 3 – Using the information in the outer IP headers, VTEP-1 performs an IP address lookup for VTEP-2's address to resolve the next-hop in the transit or IP network. Using the outer MAC header information, VTEP-1 sees the router's MAC address as the next-hop device.

Step 4 – The Ethernet frame is routed towards VTEP-2 based on the outer IP header which has VTEP-2's IP address as the destination address.

Step 5 – After VTEP-2 receives the Ethernet frame, it decapsulates all the headers and forwards the packet to Host-B using the original destination MAC address.

Dell EMC VXLAN Architecture

Dell EMC networking's data center product portfolio provides a solid feature set for the data center. With the Dell EMC S and Z-series product families, Dell EMC supports the hardware-based VXLAN gateway function allowing for the extension of Layer 2 connectivity across a Layer 3 transport network providing predictable high-performance gateway between VXLAN and traditional VLAN environments.

Dell EMC's VXLAN implementation is based on RFC 7348, it uses the existing Layer 2 mechanisms such as flooding and dynamic MAC address learning to create the necessary underlay between VXLAN and traditional VLAN environments.

The mechanisms are used to perform the following key functions which are very similar to other vendor's implementations:

- BUM (Broadcast, unknown unicast, and multicast) traffic handling
- Discovery/learn remote host/VM MAC addresses and MAC-to-VTEP mappings per VXLAN segment

For each VXLAN segment, a map is created between the VXLAN segment and an IP multicast group. This multicast group address is known by all the VTEPs that have knowledge of the VXLAN segment and it is used by all BUM traffic from SA (Source Address) MACs to reach a Multicast Replicator Source (MRS) or service node. It is the service node's job to direct or service the join request received at the VTEP by the end host or VM.

The NVP (Network Virtualization Platform) controller creates a multicast tunnel pointing to the service node or multicast replicator. Each VTEP is programmed by the NVP controller to direct any incoming BUM traffic to this service node through the multicast tunnel. A single packet copy is sent to each VTEP as long as there is interest in joining a particular multicast group from hosts or VMs connected to these VTEPs.

Dell EMC's BUM traffic handling is further enhanced through the use of Bidirectional Forwarding Detection (BFD) to ensure reachability towards the service nodes is guaranteed. In a typical overlay deployment, a cluster of service nodes or MRSs is deployed in order to provide high-redundancy. Once the Dell EMC hardware VTEP establishes a tunnel towards the service node, a BFD session is created and communicated towards the NVP controller.

When it comes to VTEP discoveries, there are two approaches: **Dynamic and Static**. Dell EMC's implementation uses static configuration of the VTEPs (remote and local) via the NVP controller or manually - via CLI on the device - configuring the remote VTEP when deploying static vxlan tunnels, i.e. no NVP controller used (See *Option 2: Dell EMC Layer 2 HW-Based VTEP GW w/out controller (Static VXLAN Tunnels)* OS 9.11) .

Through the use of a static approach the implementation avoids the need for yet another protocol to discover the VTEPs.

As soon as the VTEPs have been configured on the NVP controller or statically configured via CLI on the device (no NVP controller used) and IP reachability has been confirmed, local and remote MAC learning takes place. During the learning phase, the SOURCE ADDRESS MAC address (local and remote) is

mapped to a single VTEP tunnel IP address and logical network/VXLAN segment using a slightly different method.

For a remote SOURCE ADDRESS MAC address, when using a controller, the NVP controller creates a remote SOURCE ADDRESS MAC to remote VTEP binding. If there is no controller, and static vxlan is used, remote MAC learning on tunnels is derived from the inner SOURCE ADDRESS MAC of the packet(s) received on the tunnel.

For a local SOURCE ADDRESS MAC address, when using a controller, the Dell EMC VTEP switch informs the NVP controller through OVSDB protocol of the SOURCE ADDRESS MAC learned on a specific port or VLAN and if no controller is used, local SOURCE ADDRESS MAC is done when a data packet is received on the access port of the local VTEP device just like any other switch

The bindings of the SOURCE ADDRESS MAC(s) to VTEP are propagated by the NVP controller to the VTEPs building a full mesh connectivity infrastructure. See Figure 4, on the specific steps about how the Dell EMC VTEP GW implements this communication.

Dell EMC VTEP Architecture on the S and Z series Data Center Switches

Dell EMC's initial VTEP implementation is based on a client to server communication relationship between an NVP (Network Virtualization Platform) and the VTEP. Later on, a different architecture non-controller based is discussed.

In Dell EMC's initial implementation, the NVP entity is the network orchestrator provided by VMware's NSX-v Platform.

The functions provided by the Dell EMC VTEP are the following:

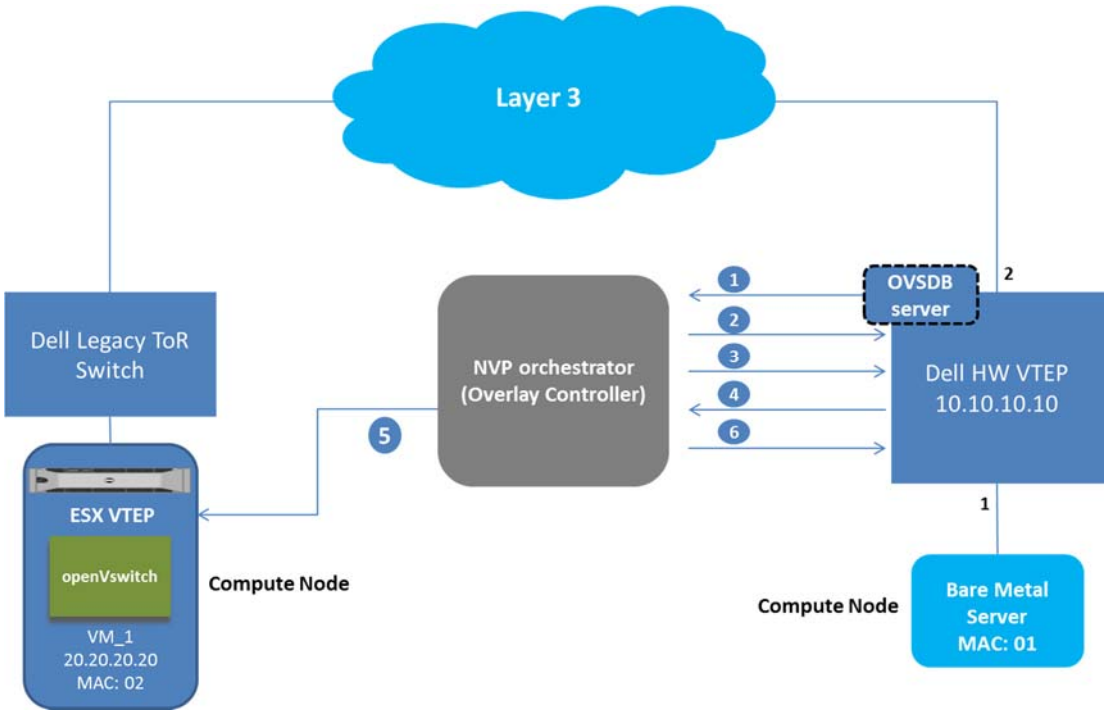
1. Create all the necessary logical networks between the virtualized and non-virtualized environments.
2. Identify and bind <Port, VLAN> to a logical network
3. Maintain the remote and local host MAC bindings to the respective VTEP
4. Establish communication and be managed by the NVP
5. Support standard communication protocol known as Open vSwitch Database (OVSDB) between NVP and itself.

The functions provided by the NVP orchestrator are the following:

1. It connects to the NVP gateway/VTEP through a TCP connection which must be configured by the user.
2. It orchestrates a Layer 2 network between the two VTEPs
    a. Binds Port and VLAN
    b. Installs VTEP tunnels between the hosts
    c. Installs both the remote and local VM/host SOURCE ADDRESS MACs on the VTEPs
3. Monitors network database from the VTEPs
4. Pushes the BFD configuration for VXLAN tunnels created towards the service nodes

Figure 4, shows the implementation steps on the VTEP as two hosts initiate to communicate.

Figure 4 Dell EMC EMC VXLAN Tunnel End-Point (VTEP) Implementation



Step 1
User based VXLAN configuration is passed on to the NVP controller through the OVSDB protocol. The configurations include, <port, vlan, VTEP information with respect to VXLAN configuration>. Dell EMC's VTEP implementation uses SSL certification between the Dell EMC VTEP and NVP prior to exchanging any information.

Step 2
Based on the information received in Step 1, the NVP controller creates a logical network consisting of a logical netwok ID, and VXLAN network ID or segment. The logical network and VXLAN ID/segment is created by the NVP controller administrator, this configuration is not created automatically.

Step 3
Bind the port, VLAN, and VTEP information from Step 1, to the logical network(s) created in Step 2. In other words, port X or VLAN Y participating on VXLAN instance Q from VTEP A bind to logical network M.

Step 4
Learn local MAC address on VTEP and create tunnel mapped to learned MAC address on the logical network M. All locally learned MACs are advertised to the NVP controller with the VTEP IP address/tunnel as the destination tunnel.
Step 5
NVP controller advertises the mapped information to the remote VTEP and the remote VTEP installs this in its forwarding table.

Step 6
NVP controller repeats step 4 for remote VTEP's host or VM and advertises this information to the Dell EMC HW VTEP where the Dell EMC EMC VTEP installs this entry. The entry information contains the remote SOURCE ADDRESS MAC, remote VTEP IP on which the remote SOURCE ADDRESS MAC has been learned and the logical network.

As far as multicast, broadcast, unknown unicast traffic originated at the SOURCE ADDRESS MAC, the NVP controller creates a multicast tunnel pointing to the service node or multicast replicator. Each VTEP is programmed by the NVP controller to direct any incoming BUM traffic to this service node through the multicast tunnel. A single packet copy is sent to each VTEP as long as there is any interest from hosts or VMs connected to these VTEPs.

Deployment Considerations/Guidelines

Whenever deploying a Dell EMC HW based VTEP Gateway the following considerations or guidelines should be taken:

1. **Ensure Dell EMC S6000-ON or S4048-ON are running at least release OS 9.10 or above**
2. **Ensure IP connectivity is up between the NVP controller and the Dell EMC switch**
3. **Ensure VMware NSX build is 6.2.2 or later. NSX 6.2.2 supports HW based VTEP GW**
4. **Configure jumbo frames (Dell EMC MTU 9216) or elephant flow (1600 MTU) frame on all devices creating the underlay. Most networking devices support up to 9216 byte MTU size.**

Dell EMC HW-Based VXLAN Gateway/VTEP Deployment Options

There is no denying that virtualization (compute and network) play a key role in today's data centers. It has transformed the data center from a sunken expense into a revenue generating asset. It has provided enterprises business agility, competitive advantage, and most importantly efficient usage of resources.

Dell EMC's networking data center product portfolio is designed to deliver on these benefits. The Dell EMC S and Z series family support and function as a hardware-based VXLAN gateway. They seamlessly bridge and connect VXLAN and VLAN segments as a single Layer 2 domain across a Layer 3 infrastructure without incurring any performance degradation. The current hardware supports up to 8,000 unique VNIDs and 511 tunnels. Each tunnel can carry multiple VNIDs.

Note: Although VXLAN allows up to 16M unique VNIDs or VXLAN segments, the current silicon supports up to 8,000 unique VNIDs with 16M logical IDs available. The current number of VNIDs supported covers the majority of today's VXLAN to non-VXLAN environments deployments.

The key VXLAN encapsulation and de-capsulation function is hardware-based providing line-rate performance regardless of frame size. Figures 5 and 6 show two key deployments of the hardware-based VTEP configurations from the Dell EMC data center switches.

There are two supported HW-based VTEP GW solutions.

- Dynamic using an NVP (Network Virtualization Platform) controller such as VMware's NSX-v to establish the VTEP tunnels between the different data centers.

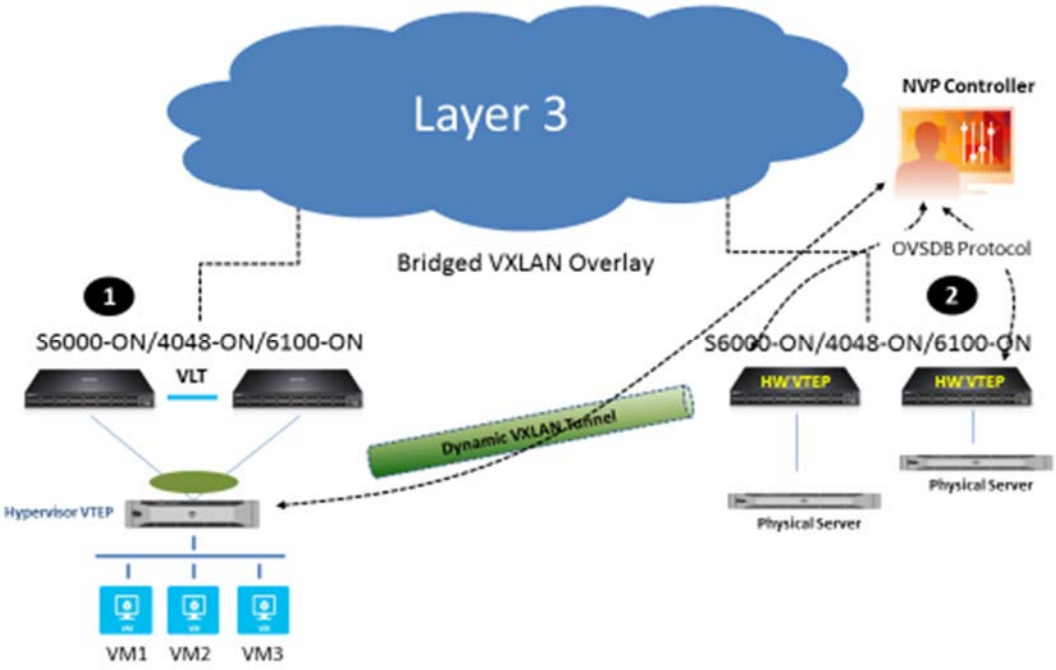
- Static using manually configured VTEP tunnels between the different data centers.

Option 1: Dell EMC Layer 2 HW-Based VTEP GW with Controller OS 9.10

Figure 5 shows the Dell EMC S and Z- series switches working in conjunction with the NVP controller (VMware NSX) to setup dynamic VTEP tunnels between the Dell EMC hardware-based VTEP and the software-based VTEP hypervisor allowing communication between the VMs and the physical server. The dashed lines show the communication between the VTEPs (software or hardware-based) and the NVP controller.

On the left side of the topology, the hypervisor VTEP or software based VTEP has dual links to the Dell EMC S6000s. The Dell EMC S6000s are configured in a VLT (Virtual Link Trunk) domain creating a virtual switch towards the software based VTEP hypervisor. This configuration provides redundancy at the ToR, in this case the Dell EMC S6000s. Layer 3 is configured from the hypervisor to the core consisting of a pair of Z9100s or S6000s.

Figure 5 Current Deployment - Dell EMC Hardware-Based L2 VTEP with NVP controller



Option 2: Dell EMC Layer 2 HW-Based VTEP GW w/out controller (Static VXLAN Tunnels) OS 9.11

The second deployment type (see Figure 8) shows the Dell EMC S and Z switches as a point-to-point static VXLAN tunnel. In this configuration, there is no controller present and all VTEP tunnels are created manually/statically between the local and remote VTEPs. Although only S6000-ON and S40489-ON are illustrated in each diagram, the entire S and Z series data center switch family supports static vxlan as well.

Host or VM SOURCE ADDRESS_MAC learning is identical to a controller based deployment where the SOURCE ADDRESS_MAC is learned as soon as data packet is received at the access port of the VTEP switch. With static VXLAN, address learning is enabled by default.

Dell EMC EMC's static vxlan implementation is efficient and uses simple logical network ID mapping with a specific VTEP tunnel.

BUM Data Packet Handling

When a packet arrives on an access port with a destination MAC address being either a broadcast, unknown unicast, or multicast (BUM) for which there is no knowledge on the VTEP switch which remote VTEP tunnel to be used for the destination MAC address, the following steps are taken by the local VTEP switch:

Step 1
Capture the logical network ID configured on this port or vlan interface on which the packet has been received.

Step 2
Learn the SOURCE ADDRESS MAC address on this port or vlan interface.

Step 3
Send a unicast copy of the packet to each remote VTEP over the vxlan tunnel which is configured to be part of this logical network. Also send a copy to any local access port that are also part of this logical network.

Unicast Data Packet Handling

When a data packet arrives on an access port with a unicast destination MAC address, the following steps are taken by the VTEP.

Step 1
The logical network ID is derived from the VLAN or port on which the packet has been received
Step 2
A lookup is performed in the Layer 2 entry table with the VFI and a unicast destination MAC address.

- If the lookup is successful and if the packet is destined to a local access port(s), the native packet will be sent on the access port(s), however, if the packet is destined to a remote VTEP tunnel, the packet is encapsulated with a VXLAN header with a VNID based VFI and the destination IP

address based on the remote VTEP tunnel and is sent out to the next hop along the path to the remote VTEP tunnel

- If the lookup is unsuccessful, the same steps that apply to a BUM traffic is performed.

MAC learning

Local and remote SOURCE ADDRESS MAC learning happen at different places during data forwarding. Local learning takes place when a data packet is received on the access ports. Whereas remote learning takes place at the VTEP tunnel.

Learning is enabled by default and it cannot be disabled. The same applies with remote MAC learning which is enabled on the VTEP tunnels.

For example, when a data packet is received on an access port, this SOURCE ADDRESS MAC address is mapped to a specific VNID and ingress VTEP tunnel ip address. This piece of information (VNID, SOURCE ADDRESS MAC, and VTEP tunnel ip) is programmed in the switch's Layer 2 entry or forwarding table.

Figures 7 & 8 show an output of both local and remote address learning per vxlan-instance. Note the tunnel ID or ip address of the tunnel. This is the vtep tunnel on which the remote MAC address is learned. It is also the static tunnel used to reach the MAC address.

Figure 6 Local MAC address learning

```
S4048-ON_1#sh vxlan  vxlan-instance 1 unicast-mac-lo
Total Local Mac Count:   1
VNI          MAC                 PORT       VLAN
5000         d4:ae:52:9e:0c:f8   Te 1/41     10
```

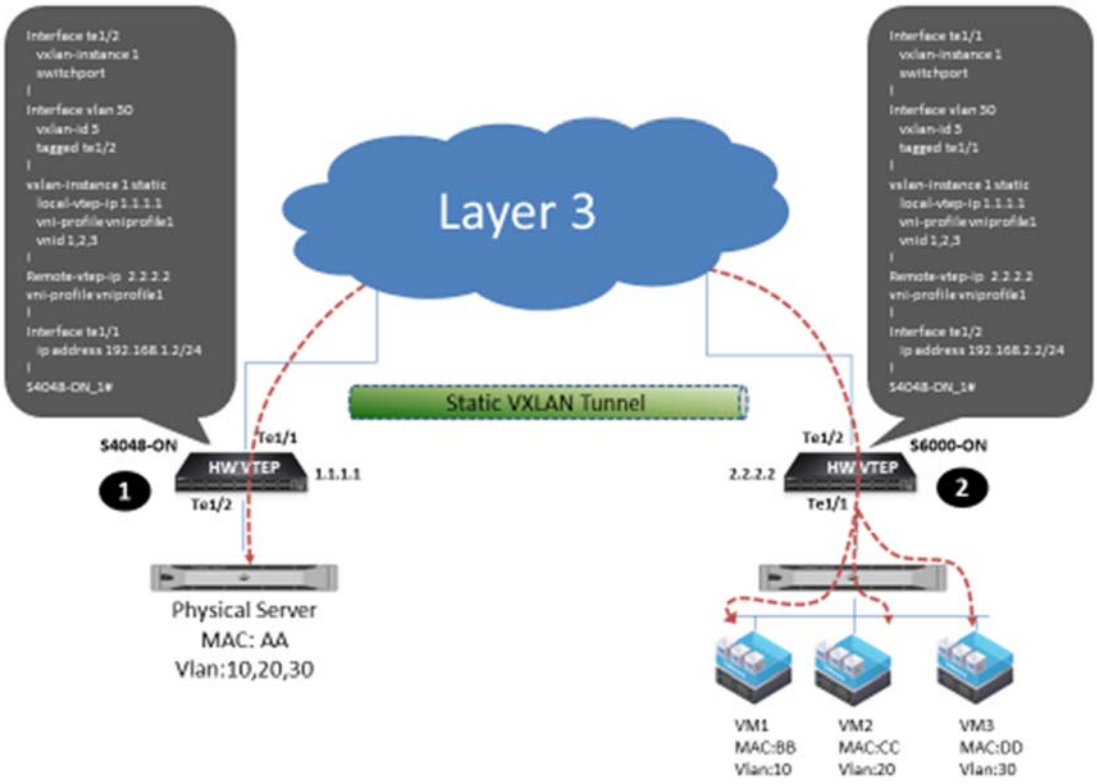
Figure 7 Remote MAC address learning

```
S4048-ON_1#sh vxlan  vxlan-instance 1 unicast-mac-rem
Total Remote Mac Count:   3
VNI          MAC                       TUNNEL
5000         00:50:56:be:34:e8         10.10.10.10
5000         00:50:56:be:49:2a         30.30.30.3
5000         00:50:56:be:e1:c3          10.10.10.9
```

There are two deployment options. Option 1 supported from day one of the release and Option 2 as a future choice.

Figure 8 shows the static vxlan configuration of the Dell EMC data center switches. The sample configuration captures only one vlan (vlan 30) but it shows a single tunnel with a single vni profile mapping to all three VNIs (10,20,3). The configuration would still call for vlans 10 and 20 to be configured on both switches.

Figure 8 Current Dell EMC Hardware-Based VTEP.



Dell EMC's static VXLAN configuration is divided into 5 short steps:

Step 1 – Create a static VXLAN instance
Step 2 – Create a VNI profile
Step 3 – Associate a VNID to the VNI profile
Step 4 – Associate a remote VTEP to the VNID
Step 5 – Associate the VNID to a VLAN

Caveats

There are some guidelines to follow when deploying static vxlan. Table 3 describes these guidelines.

Table 3 Dell EMC Static VXLAN current guidelines

Limitations/Guidelines
• No stacking or VLT supported • One vxlan instance supported • No Layer 3 over VXLAN • No SNMP or REST API supported • 4,000 VNIs • No Load balancing • 802.1p QoS marking preservation not supported over VXLAN tunnel

Each deployment has its own set of advantages and disadvantages (see Table 4).

Table 4 Dell EMC Hardware-Based VTEP with Controller vs. without Controller comparison

Dell EMC HW VTEP	Advantage	Dis-advantage
w/ controller	• Dynamic VTEP tunnels configurations • Minimal manual configuration needed • Feature maturity • Large implementation	• Requires additional components, i.e nvp controller, software, protocol • Orchestration vendors proprietary implementations
Static VXLAN	• Straight forward, no additional components needed such as a controller, manager, or communication protocol. • Simple tunnel implementation • Small/medium size implementation	• Requires manual configuration for each VTEP tunnel • Potential configuration errors • Potential for excessive flooding. • Feature maturity

Conclusion

VXLAN builds on the traditional and well understood 802.1Q VLAN technology to address the challenges found in today's data center.

Before VXLAN, organizations were limited to building isolated/dedicated infrastructures where virtual environments and physical environments could only communicate with their respective counterparts. Unfortunately the typical organization's business infrastructure runs on a mixed set of applications - virtualized and physical - and therefore a solution was needed.

Being able to bridge the connectivity gap between the ubiquitous virtual and physical environments and provide data center interconnect (DCI) functionality has become an operational necessity.

Dell EMC Networking with its data center product portfolio supports VXLAN in hardware so customers can leverage the predictable high-performance and density expected from any large enterprise solution provider.

With Dell EMC Networking, VXLAN segments whether virtualized or physical can seamlessly connect with a traditional VLAN segment allowing for multi-tenants to reside in either domain while keeping flexibility, scalability, and familiarity as part of its solution to our customers.