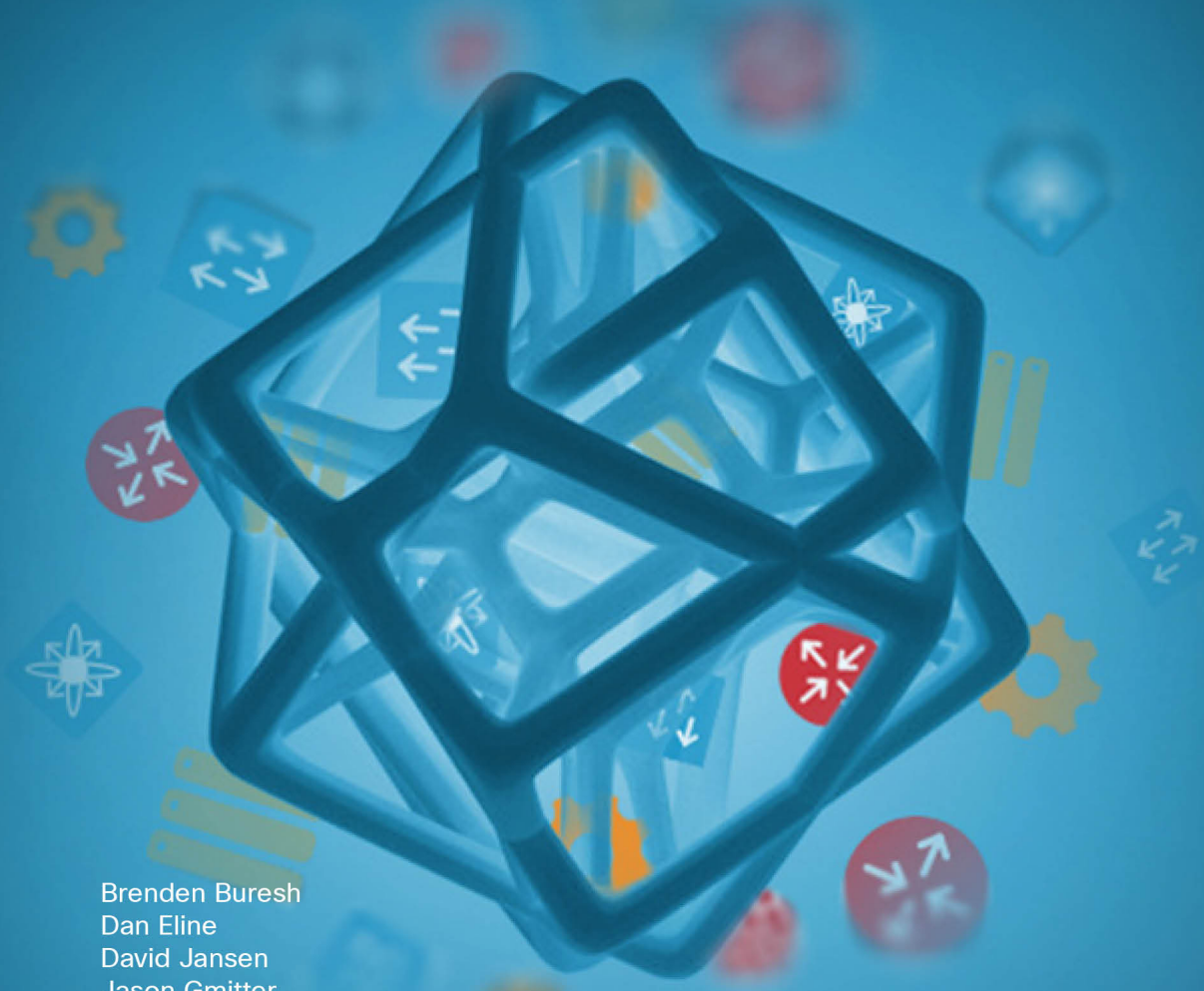
A Modern, Open and Scalable Fabric

VXLAN EVPN



Brenden Buresh
Dan Eline
David Jansen
Jason Gmitter
Jeff Ostermiller
Jose Moreno
Kenny Lei
Lilian Quan
Lukas Krattiger
Max Ardica
Rahul Parameswaran
Rob Tappenden
Satish Kondalam

cisco

Preface

Introduction

VXLAN EVPN

For many years now, VLANs have been the de-facto method for providing network segmentation in data center networks. Standardized as IEEE 802.1Q, VLANs leverage traditional loop prevention techniques such as Spanning Tree Protocol which not only imposes restrictions on network design and resiliency, but it also results in an inefficient use of available network links due to the blocking of redundant paths, required to ensure a loop free network topology.

VLANs also use a 12-bit VLAN identifier to address Layer 2 segments, allowing for the addressing of up to a practical limit of ~4,000 VLANs. In modern data center network deployments, VLANs have become a limiting factor to IT departments and cloud providers as they build increasingly large and complex, multi-tenant data centers.

Modern data centers require an evolution from the restraints of traditional Layer 2 networks. Cisco, in partnership with other leading vendors, proposed the Virtual Extensible LAN (VXLAN) standard to the IETF as a solution to the data center network challenges posed by traditional VLAN technology and the Spanning Tree Protocol. At its core, VXLAN provides benefits of elastic workload placement, higher scalability of Layer 2 segmentation, and connectivity extension across the Layer 3 network boundary. However, without an intelligent control plane, VXLAN has its limits due to its flood and learn behavior.

Multi-Protocol Border Gateway Protocol (MP-BGP) introduced new Network Layer Reachability Information (NLRI) to carry both Layer 2 MAC and Layer 3 IP information at the same time. By having the combined set of MAC and IP information available for forwarding decisions, optimized routing and switching within a network becomes feasible and the need for flood and learn behavior which limits its ability to scale. The extension that allows BGP to transport Layer 2 MAC and Layer 3 IP information is called EVPN – Ethernet Virtual Private Network.

In summary, the advantages provided by a VXLAN EVPN solution are as follows:

- Standards-based Overlay (VXLAN) with standards-based control plane (BGP)
- Layer 2 MAC and Layer 3 IP information distribution by control plane (BGP)
- Forwarding decision based on scalable control plane (minimizes flooding)
- Integrated Routing/Bridging (IRB) for Optimized Forwarding in the Overlay
- Leverages Layer 3 ECMP – all links forwarding – in the underlay
- Significantly larger namespace in the overlay (16M segments)
- Integration of physical and virtual networks with hybrid overlays
- Facilitation of Software-Defined-Networking (SDN)

This book explores VXLAN EVPN, beginning with the introductory stages, gaining an understanding of terms and concepts and evolving through deployments within a single data center to multiple data centers. The book also addresses design and integration of L4-L7 network services, co-existence with brownfield environments, and the tools needed to build, operate, and maintain a VXLAN EVPN Fabric. At the conclusion of this book, readers will have a solid foundation of VXLAN EVPN and a comprehension of real-world use cases that can be immediately utilized to assist in development of a plan to successfully transition to a next generation data center Fabric.

VXLAN BGP EVPN features and functionality discussed within are available on the following Cisco Nexus Series Switches:

- Cisco Nexus 9000 Series starting NX-OS 7.0
- Cisco Nexus 7000 Series and Nexus 5600 starting NX-OS 7.3

Authors

This book represents a collaborative effort between Technical Marketing and Sales Engineers during a week-long intensive session at Cisco Headquarters in San Jose, CA.

- Brenden Buresh - Systems Engineering
- Dan Eline - Systems Engineering
- David Jansen - Systems Engineering
- Jason Gmitter - Systems Engineering
- Jeff Ostermiller - Systems Engineering
- Jose Moreno - Systems Engineering
- Kenny Lei - Technical Marketing
- Lilian Quan - Technical Marketing
- Lukas Krattiger - Technical Marketing
- Max Ardica - Technical Marketing
- Rahul Parameswaran - Technical Marketing
- Rob Tappenden - Systems Engineering
- Satish Kondalam - Technical Marketing

Acknowledgements

A special thanks to Cisco's Insieme and EISG BU Executives, Technical Marketing and Engineering teams, who supported the realization of this book. Thanks to Carl Solder, James Christopher, Joe Onisick, Matt Smorto, Victor Moreno and Yousuf Khan for supporting this effort. Thanks to Cisco Sales Leadership for supporting the group of individual contributors who have dedicated their time in authoring this book.

We would also like to thank Cynthia Broderick for her exceptional resource organization and support throughout our journey, and Shilpa Grandhi for making sure that everything worked smoothly and that all thirteen writers made it to the end.

We are also genuinely appreciative to our Book Sprint (www.booksprints.net) team:

- Adam Hyde (Founder)
- Henrik van Leeuwen (Illustrator)
- Juan Carlos Gutiérrez Barquero (Technical Support)
- Julien Taquet (Book Producer)
- Laia Ros (Facilitator)
- Raewyn Whyte (Proof Reader)

Laia and the team created an enabling environment that allowed us to exercise our collaborative and technical skills to produce this technical publication to meet a growing demand.

Organization of this book

Readers can read this book sequentially, or go directly to individual chapters. Generally, chapters should be self-contained, but in order not to duplicate information, pointers to other parts of the book may exist. Where applicable, hyperlinks will take the reader to Internet pages that provide additional levels of details.

Why a New Approach

The motivation for the development of this new technology is explored, as well as the benefits organizations can extract from it. A brief overview explains how the data and control planes of VXLAN EVPN contribute to solving business challenges that many organizations are confronted with.

Fundamental Concepts

This chapter shifts the focus from the "Why" to the "What". Essential concepts for understanding the technology are laid out, to set the necessary foundation for understanding the rest of the book. The basics of VXLAN technology are articulated, as well as the fundamentals of networking in a VXLAN Fabric.

Software Overlays

The intersection of virtual and physical networking is discussed in order to help the reader gain the required perspective to decide how to best implement VXLAN technology to support these virtualized environments.

Single-POD VXLAN Design

A deep dive into the inner workings of the VXLAN protocol, including best practices, design recommendations and lessons learned about both the underlay and overlay elements of a VXLAN Fabric with MP-BGP EVPN. Even though this publication should not be considered as a configuration guide, command examples are included so the reader interested in the actual deployment can understand the required configuration for each of the individual components of the technology.

External Connectivity for VXLAN Fabrics

After a VXLAN Fabric is up and running, the next step is connecting it to the rest of the world. This chapter discusses the details of connecting the VXLAN Fabric over Layer 2 and Layer 3 with non-VXLAN networks. These techniques are used in the last section of the chapter to demonstrate a procedure for migrating a brownfield legacy network into a VXLAN Fabric.

Layer4-Layer7 Services

Ethernet routers and switches are not the only elements providing network services in a data center. Layer 4-Layer 7 devices like firewalls or application delivery controllers are often indispensable for secure and efficient application delivery. This chapter addresses how to connect these network appliances to the VXLAN Fabric so the data center network offers the best performance and availability end-to-end.

Multi-POD and Multi-Site Designs

Most organizations today have business continuity requirements that determine how network infrastructure is deployed in multiple geographical locations. Whether networks are deployed across two rooms in the same building or across sites thousands of miles apart, this chapter illustrates how to resolve the distributed network problem, achieving at the same time workload mobility and fault containment.

Operations and Management

Efficient management practices can optimize the way computer networks are monitored and deployed, and VXLAN Fabrics are no exception to this. This chapter describes how to use both traditional and modern network techniques to manage a VXLAN Fabric. Off-the-shelf network management software will be discussed as well as open source approaches or DevOps-inspired tools such as Puppet, Chef and Ansible.

Intended Audience

The intended audience for this book is network professionals with a general need to understand how to deploy VXLAN networks in their organizations to unleash the full potential of modern networking. While interested network administrators will reap the most benefits from this content, the information included within this book may be of use to every IT professional interested in networking technologies. Elements in this book explore how VXLAN and EVPN solve network challenges that have daunted the industry for years, as well as how to deploy constructs that are typically seen in traditional networks with this new technology.

Book Writing Methodology

How many engineers do you need to write a book? Thirteen! Thirteen highly-skilled professionals got together in Building 31 in Cisco headquarters in San Jose, California. Thirteen In, One Out: Thirteen individually-selected highly-skilled professionals from diverse backgrounds accepted the challenge to duel thoughts over the course of five days. Figuring out how to harness the brain power and collaborate effectively at first seemed to be nearly impossible, however, opposites attracted and the team persisted through the hurdles. The Book Sprints (www. booksprints.net) methodology captured each of our strengths, fostered a team-oriented environment, and accelerated the overall time to completion. The assembled group leveraged their near two hundred years of experience and a thousand hours of diligent authorship which resulted in this publication. Representing four continents and seven nationalities, after five long days, one book was produced. Fueled by Chinese, Indian, Japanese, Mexican and Italian food, but first and foremost by American coffee, together with their facilitator from Book Sprints, the writers poured their experience and knowledge into this publication.

Why a New Approach

Introduction

IT is evolving toward a cloud consumption model. This transition affects the way applications are being architected and implemented, driving an evolution in data center infrastructure design to meet these changing requirements. As the foundation of the modern data center, the network must also take part in this evolution while also meeting the increasing demands of server virtualization and new microservices-based architectures. This demands a new paradigm that must deliver on the following areas:

- **Flexibility** to allow workload mobility across any floor tile in any site
- **Resiliency** to maintain service levels even in failure conditions (better fault isolation)
- **Multi-tenancy** capabilities and better workload segmentation
- **Performance** to provide for adequate bandwidth and predictable latency, independent of scale for demanding workloads
- **Scalability** from small environments to cloud scale while maintaining the above characteristics

As a result, modern data center networks are evolving from traditional hierarchical designs to horizontally-oriented spine-leaf architectures with hosts and services distributed throughout the network. These networks are capable of supporting the increasingly common east-west traffic flows experienced in modern applications. In addition, there are clustering technologies and virtualization techniques that require Layer 2 adjacency.

Evolving user demands and application requirements suggest a different approach that is simple, and more agile. Ease of provisioning and speed are now critical performance metrics for data center network infrastructure that supports physical, virtual, and cloud environments - without compromising scalability or security. These are the main drivers for the industry to look at Software Defined Network (SDN) solutions.

Cisco Application Centric Infrastructure (ACI) is an innovative data center architecture that simplifies, optimizes and accelerates the entire application lifecycle through a common policy management framework. ACI provides a turnkey solution to build and operate an automated cloud infrastructure. An alternative option is a VXLAN Fabric with BGP EVPN control plane that provides a scalable, flexible and manageable solution to support growing demands of cloud environments.

This chapter introduces the concepts of VXLAN EVPN and the problem it has been designed to solve.

Why VXLAN Overlay

Network overlays are a technique used in state-of-the-art data centers to create a flexible infrastructure over an inherently static network by virtualizing the network. Before going into the details of how overlays work, the challenges they face, and the solutions to overlay problems, it's worth spending some time to understand why traditional networks are so static.

When networks were first developed, there was no such thing as an application moving from one place to another while it was in use. As a result, the original architects of TCP/IP used the IP address as both the identity of a device and its location on the network. This was a perfectly reasonable thing to do as computers and their applications did not move, or at least they did not move very fast or very often.

Today in the modern data center, applications are often deployed on virtual machines (VMs) or containers. The virtualized application workload can be stretched across multiple locations. The application endpoints (VMs, containers) can also be mobile among different hosts. Their identities (IP addresses) no longer indicate their location. Due to the tight coupling of an endpoint's location with its identity in the traditional network model, the endpoint may need to change its IP address to indicate the new location when it moves. This breaks the seamless mobility model required by the virtualized applications. Therefore, the network needs to evolve from the static model to a flexible one in order to continuously support communications among application endpoints regardless of where they are. One approach is to separate the identity of an endpoint from its physical location on the network so the locations can be changed at will without breaking the communications to the endpoint. This is where overlays come into the picture.

An overlay takes the original message sent by an application and encapsulates it with the location it needs to be delivered to before sending it through the network. Once the message arrives at its final destination, it is decapsulated and delivered as desired. The identities of the devices (applications) communicating are in the original message, and the locations are in the encapsulation, thus separating the location from the iden-

tity. This encapsulation and decapsulation is done on a per-packet basis and therefore must be done very quickly and efficiently.

Today, according to market research, approximately 60-70% of all application workloads are virtualized, however, more than 80% of the servers in use today are not running a hypervisor. Of course, every data center is unique and the mix of servers running virtualized workloads vs. non-virtualized workloads covers the entire spectrum. Any network solution for the data center must address this mix.

Cisco, in partnership with other leading vendors, proposed the Virtual Extensible LAN (VXLAN) standard to the IETF as a solution to the data center network challenges posed by traditional VLAN technology. The VXLAN standard provides for the elastic workload placement and higher scalability of Layer 2 segmentation that is required by today's application demands.

VXLAN is designed to provide the same Ethernet Layer 2 network services as VLANs do today, but with greater extensibility and flexibility. Implementing VXLAN technologies in the network will provide the following benefits to every workload in the data center:

- Flexible placement of any workload in any rack throughout and between data centers
- Decoupling between physical and virtual networks
- Large Layer 2 network to provide workload mobility
- Centralized Management, provisioning, and automation, from a controller
- Scale, performance, agility and streamlined operations
- Better utilization of available network paths in the underlying infrastructure

Why a Control Plane

When implementing an overlay, there are three major tasks that have to be accomplished. Firstly, there must be a mechanism to forward packets through the network. Traditional networking mechanisms are effective for this.

Secondly, there must be a control plane where the location of a device or application can be looked up and the result used to encapsulate the packet so that it may be forwarded to its destination.

Thirdly, there must be a way to update the control plane such that it is always accurate. Having the wrong information in the control plane could result in packets being sent to the wrong location and likely dropped.

The first task, forwarding the packet, is something that networking equipment has always delivered. Performance, cost, reliability, and supportability are fundamental considerations for the network which must equally apply to both the physical and overlay networks respectively.

The second task, control plane lookup and encapsulation, is really an issue of performance and capacity. If these functions were performed in software, they would consume valuable CPU resources and add latency when compared to hardware solutions.

The third component of an overlay is the means by which modifications to the control plane are updated across all network elements. This updating is a real challenge and a concern for any data center administrator due to the potential for application impact from packet loss if the control plane malfunctions.

VXLAN Control Plane

VXLAN as an overlay technology does not provide many of the mechanisms for scale and fault tolerance that other networking technologies have developed and are now taking for granted. In a VXLAN network, each switch builds a database with the locally connected hosts. A mechanism is required so that other switches learn about those hosts. In a traditional network, there is no mechanism to distribute this information. The only control plane previously available was a data plane-driven model called flood and learn. For a host to be reachable, its information has to be flooded across the network. Ethernet networks have operated with this deficiency for decades.

While the demand for scalable networks increases, the effects of flood and learn need to be mitigated. For a VXLAN overlay, a control plane is required that is capable of distributing the Layer 2 and Layer 3 host reachability information across the network. Early implementations of VXLAN lacked the ability to carry Layer 2 network reachability information, therefore, Ethernet VPN (EVPN) extensions were added to Multi-Protocol BGP (MP-BGP) to carry this information.

MP-BGP EVPN

MP-BGP EVPN for VXLAN provides a distributed control plane solution that significantly improves the ability to build and interconnect SDN overlay networks. MP-BGP EVPN control plane for VXLAN offers the following key benefits:

- Control plane learning for end host Layer 2 and Layer 3 reachability information.
- Ability to build a more robust and scalable VXLAN overlay network
- Supports multi-tenancy
- Provides integrated routing and bridging
- Minimizes network flooding through protocol-driven host MAC/IP route distribution
- ARP suppression to minimize unnecessary flooding
- Peer discovery and authentication to improve security
- Optimal east-west and north-south traffic forwarding

Looking Ahead

Even though VXLAN technology has attained a considerable degree of maturity in a very short time, the industry is already designing the next evolution of this technology.

Generic Protocol Encapsulation (VXLAN-GPE)

VXLAN is one of many data plane encapsulations available. Examples of other UDP-based encapsulations are LISP (Locator/ID Separation Protocol) and OTV (Overlay Transport Virtualization). These three encapsulations are very similar, the differences lying in the overlay shim header. While all three use the same size header, the field allocation and the naming are slightly different. Within the encapsulation, there are also variations. While VXLAN maintains an inner-MAC header, LISP only carries an inner-IP header. It becomes evident that an approach for header extensions is needed to avoid adding yet another UDP-based encapsulation.

VXLAN-GPE was invented to bring some consolidation in the UDP-based encapsulation family. A major part of VXLAN-GPE is the inclusion of a protocol-type field to define what is being encapsulated and set the meaning for the various flags and options in the overlay shim header. This protocol type describes the packet payload; currently defined types include IPv4, IPv6, Ethernet, and Network Service Header (NSH).

A prominent example for the need of this flexible protocol extension is Service Chaining and the related NSH approach.

NSH enables the possibility of dynamically specifying that certain network traffic is sent through a chain of one or more network services. The goal of NSH is to create a topology-independent way of specifying a service path. NSH also includes a number of mandatory, fixed-size context headers designed to capture network platform information. NSH even contains an optional variable length metadata field for additional extensibility and is designed to include all required information inside fixed-size fields.

Creation of yet another encapsulation protocol stands to add more confusion to the already crowded encapsulation protocol space. The extensibility of VXLAN-GPE and NSH promises to both reduce the amount of encapsulation in the industry and accommodate future network encapsulation requirements. Geneve, VXLAN-GPE, and NSH are all recent protocol drafts proposed to the IETF. The three protocols provide similar approaches to achieve flexible protocol mappings. While Geneve uses variable length options, VXLAN-GPE and NSH use fixed size options. Cisco supports open standards and will continuously reevaluate support for future encapsulations.

Evolution of the EVPN Control Plane

The current implementation of the EVPN control plane is focused on delivering scalable data center Fabrics with mobility and segmentation. As EVPN control plane implementations become more complete, the EVPN control plane may address additional use-cases such as DCI. The complete theoretical definition of the EVPN control plane is captured in a series of Internet drafts being worked on at the IETF. The general specification of EVPN accommodates use cases beyond the Data Center Fabric, including Layer 2 Data Center Interconnect.

In order to properly address the DCI requirements, the EVPN control plane implementation must be expanded to include the multi-homing functionality defined in the EVPN specification to deliver failure containment, loop protection, site-awareness, and optimized multicast replication.

Fundamental Concepts

Introduction

In the networking world, an overlay network is a virtual network running on top of a physical network infrastructure. The physical network provides an underlay function, offering the connectivity and services required to support the virtual network instances delivered in the overlay. The virtual network allows for an independent set of network services to be offered regardless of the underlay infrastructure, even though those services may be the same. As an example, it is possible to deliver Layer 2 connectivity services on top of a Layer 3 network infrastructure via an overlay network. A common example of this would be VPLS service offered over a carrier's MPLS infrastructure.

An overlay network typically provides transport of network traffic between tunnel endpoints on top of the underlay by encapsulating and decapsulating traffic between tunnel endpoints. The tunnel endpoint may be delivered through a physical network device, and perform tunnel encapsulation/decapsulation in hardware. It also may be virtual, with the tunnel endpoint process running in a hypervisor. A hardware tunnel endpoint provides greater performance leveraging hardware-based forwarding, but has less flexibility implementing new capabilities. In contrast, a software endpoint provides increased flexibility but at the cost of limited performance.

This chapter provides an overview of the concepts required to have a basic understanding of the technology and how it works.

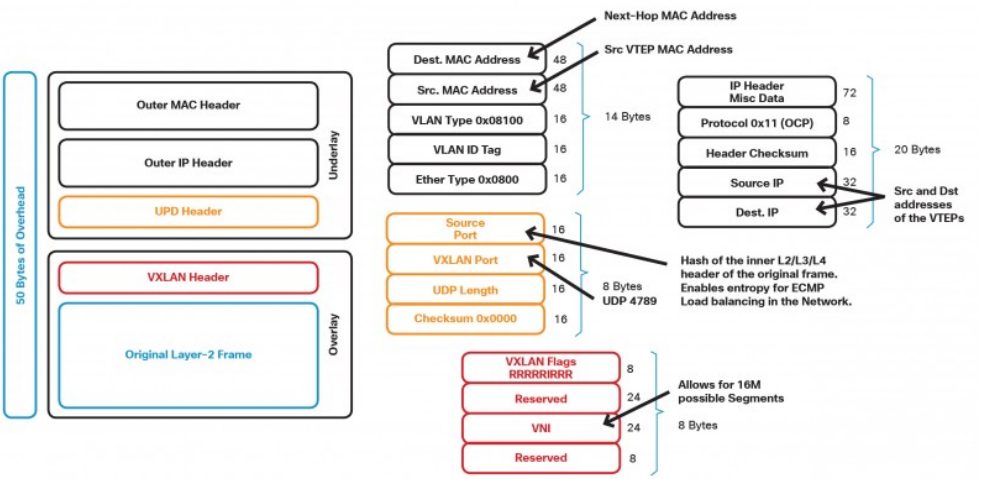
What is VXLAN?

Virtual Extensible LAN (VXLAN) as defined in RFC 7348 is an overlay technology designed to provide Layer 2 and Layer 3 connectivity services over a generic IP network. IP networks provide increased scalability, balanced performance and predictable failure recovery. VXLAN achieves this by tunneling Layer 2 frames inside of IP packets. VXLAN requires only IP reachability between the VXLAN edge devices, provided by an IP routing protocol.

There are pros and cons to consider when selecting the underlay routing protocol and these are discussed in more detail in the Single-POD VXLAN Design Chapter.

The VXLAN standard defines the packet format illustrated by the following diagram:

Figure: VXLAN Packet Format



VXLAN uses an 8-byte header that consists of a 24-bit identifier (VNID) and multiple reserved bits. The VXLAN header, along with the original Ethernet frame, is placed in the UDP payload. The 24-bit VNID is used to identify Layer 2 segments and to maintain Layer 2 isolation between the segments. With 24 bits allocated for the VNID, VXLAN can support up to 16 million logical segments.

The terminology used when describing the key components of a VXLAN Fabric include:

- VTEP – Virtual Tunnel Endpoint: The hardware or software element at the edge of the network responsible for instantiating the VXLAN tunnel and performing VXLAN encapsulation and decapsulation
- VNI – Virtual Network Instance: a logical network instance providing Layer 2 or Layer 3 services and defining a Layer 2 broadcast domain
- VNID – Virtual Network Identifier: a 24-bit segment ID that allows the addressing of up to 16 million logical networks to be present in the same administrative domain
- Bridge Domain: A set of logical or physical ports that share the same flooding or broadcast characteristics

The VXLAN tunnel endpoint function can be performed by a hardware device or by a software entity such as a hypervisor. The main advantage of using a hardware-based tunnel endpoint is the enhanced performance offered through the capabilities of the switch ASICs.

Alternatively, a software-based VTEP removes the dependency from the hardware switches, albeit at the expense of performance. Additionally, VXLAN deployments could adopt hybrid approaches, where the VXLAN tunnels are established between hardware and software VTEPs. More information on this can be found in the Software Overlays chapter.

As discussed in the introduction, the use of VXLAN technology brings several benefits to Data Center networking which include:

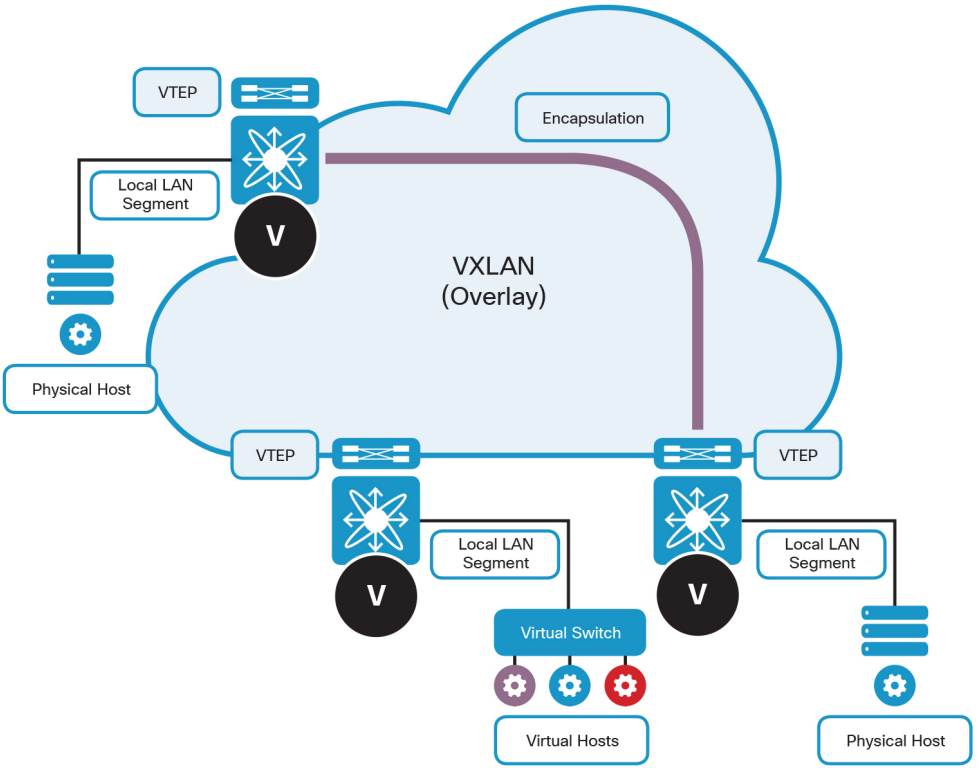
- Multi-tenancy: VXLAN Fabrics inherently support multi-tenancy both at Layer 2 (separate Layer 2 VNIs represent logically isolated bridging domains) and Layer 3 (by defining different VRFs for each supported tenant)
- Mobility: The overlay capability offered by VXLAN provides Layer 2 extension service across the data center to provide flexible deployment and mobility of physical and virtual endpoints
- Increased Layer 2 segment scale: VLAN-based designs are limited to a maximum of 4,096 Layer 2 segments due to the use of a 12 bit VLAN ID. VXLAN introduces a 24-bit VNID that theoretically supports up to 16 million distinct segments
- Multi-path Layer 2 support: Traditional Layer 2 networks support one active path because Spanning Tree (STP) expects and enforces a loop-free topology by blocking redundant paths. A VXLAN Fabric leverages a Layer 3 underlay network for the use of multiple active paths

How Does VXLAN Work?

Data Plane

VXLAN requires an underlying transport network that performs data plane forwarding. This data plane forwarding is required to provide unicast communication between end-points connected to the Fabric. The following diagram illustrates data plane forwarding in a VXLAN network.

Figure: VXLAN Overlay Network



At the same time, the underlay network can be used to deliver multi-destination traffic to endpoints connected to a common Layer 2 broadcast domain in the overlay network. Often this traffic is referred to as BUM, since it includes Broadcast, Unknown Unicast and Multicast traffic.

Two different approaches can be taken to allow transmission of BUM traffic across the VXLAN Fabric:

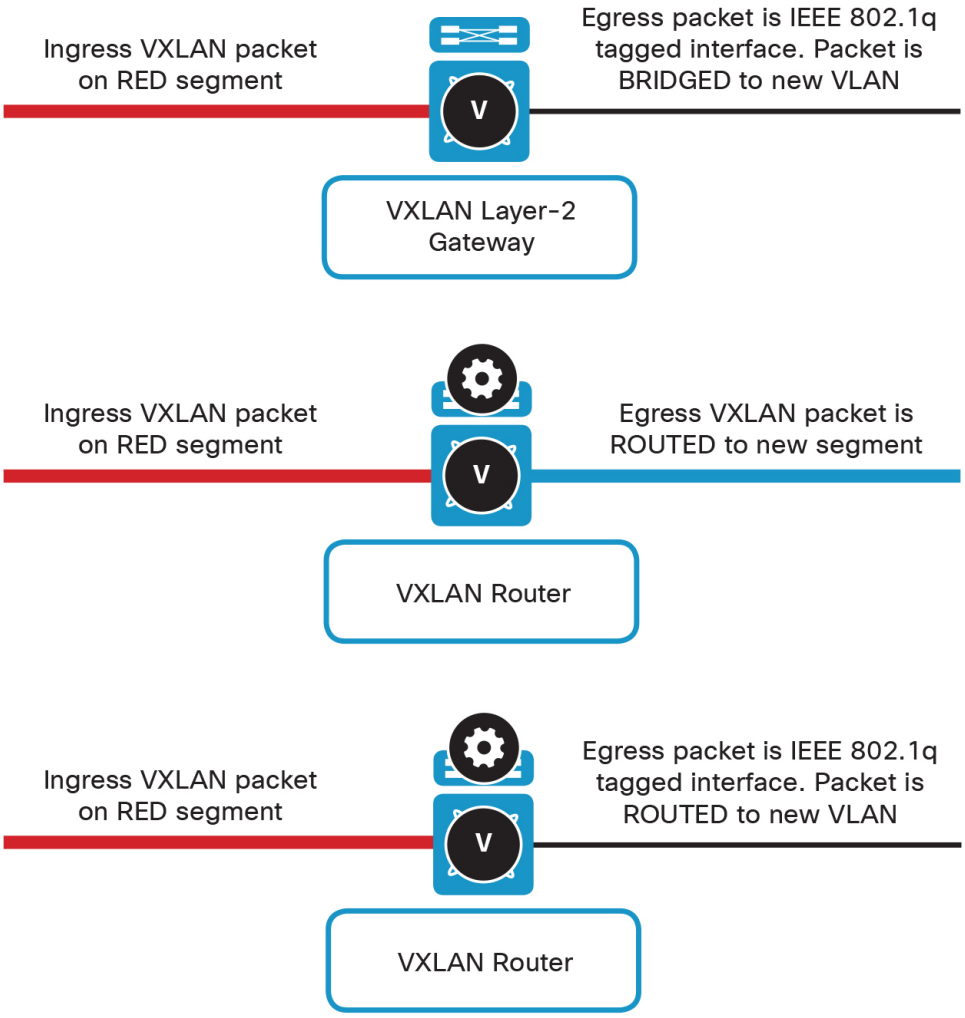
1. Leverage multicast technology in the underlay network (Protocol Independent Multicast or PIM), to make use of the native replication capabilities of the Fabric spines to deliver traffic to all the edge VTEP devices.

2. In scenarios where multicast cannot be deployed, it is possible to make use of the source-replication capabilities of the VTEP nodes that create multiple unicast copies of the BUM frames to be sent to each remote VTEP device. This approach is not as efficient as using multicast for BUM traffic replication.

VXLAN doesn't change the semantics of Layer 2 or Layer 3 forwarding and allows the VTEP to perform bridging and routing functions while leveraging the VXLAN tunnel for data plane forwarding. As such, the VTEP offers a set of different gateway functions as outlined in the following diagram.

- Layer 2 Gateway: VXLAN to VLAN bridging maps a VNI segment to a VLAN to create a common bridge domain
- Layer 3 Gateway (VXLAN Router): VXLAN to VXLAN routing provides Layer 3 connectivity between two VNIs natively so no decapsulation function is required
- Layer 3 Gateway (VXLAN Router): VXLAN to VLAN routing provides Layer 3 connectivity between a VNI and a VLAN

Figure: VXLAN Gateway Functions



Control Plane

The VXLAN RFC has to date only concerned itself with the transport (data plane) of traffic, ensuring connectivity to all hosts in a VXLAN domain. The control plane, or method by which VXLAN reachability and learning occurs, was achieved through what is known as flood and learn behavior. Simply speaking, flood and learn is a data-driven methodology wherein a VTEP that doesn't know the location of a given destination MAC floods the frame onto the VXLAN's associated multicast group. Multicast is typically used in order to provide a more manageable approach to multi-destination traffic. Instead of learning the source interface associated with a frame's source MAC address, the host learns the encapsulating source IP address of the remote VTEP. Flood and learn methodology is concerned with both the discovery (between peers) of VTEPs as well as remote endpoint location learning.

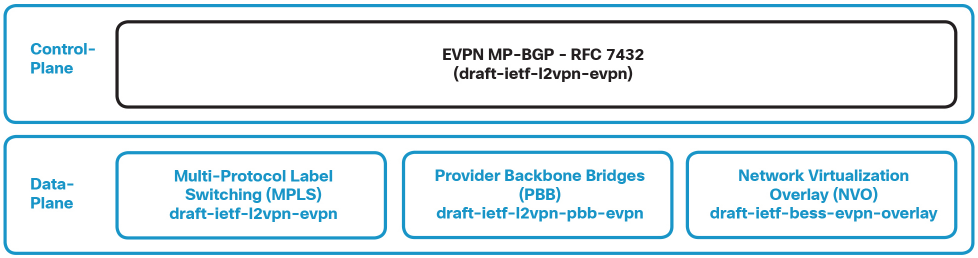
While flood and learn methodology presents a reasonably low barrier to entry for network vendors to implement a VXLAN stack, the drawback to flood and learn is first and foremost scalability. The amount of additional multicast traffic introduced into an environment can be difficult to predict and as such has been a barrier to adoption for some enterprise customers.

In order to address the concerns of scalability, the concept of a control plane to manage MAC learning and VTEP peer discovery is desirable, and preferably one that could be based on existing protocols that are generally well understood. Multi-Protocol Border Gateway Protocol (MP-BGP) with Ethernet Virtual Private Network (EVPN) extensions has been proposed as the IETF standard control plane for VXLAN. Based on the existing MP-BGP standard, the MP-BGP EVPN control plane provides protocol-based VTEP peer discovery and endpoint reachability information distribution that allows more scalable VXLAN overlay network designs. The MP-BGP EVPN control plane introduces a set of features that reduces the amount of traffic flooding in the overlay network and enables optimal forwarding for both east-west and north-south traffic. Relevant to the data center use case, EVPN provides reachability information for both L2 and L3 endpoints. Extending this level of reachability, and adding the capability for ARP suppression, reduces the required amount of flooding in the network. One additional benefit of the EVPN control plane is that it provides VTEP peer discovery and authentication, mitigating the risk of rogue VTEPs in the VXLAN overlay network.

In order to understand MP-BGP EVPN functionality, it is helpful to have a background understanding of MP-BGP as it is commonly used in MPLS networks. A traditional MPLS network has a full mesh of BGP routers or route reflectors for scaling that exchange reachability and profile information for L3VPNs (or L2VPNs in the case of VPLS for example). The combination of route distinguishers (RD) and VPNv4 addresses ensure the ability to uniquely identify a target, and routes can be selectively learned using route target (RT) filtering.

In the EVPN control plane, there are technically three data plane options: Multi-Protocol Label Switching (MPLS, draft-ietf-l2vpn-evpn), Provider Backbone Bridging (PBB, draft-ietf-l2vpn-pbb-evpn), and Network Virtualization Overlay (NVO, draft-ietf-bess-evpn-overlay).

Figure: EVPN IETF Draft



For the purposes of this book, NVO will be assumed when discussing EVPN.

Networking in a VXLAN Fabric

In traditional Layer 2 access networks, the Layer 3 default gateway is most commonly placed at the aggregation layer. Generally, the pair of aggregation switches leverage a first-hop redundancy protocol such as HSRP, VRRP or GLBP to provide a redundant default gateway IP address. Depending on configuration and protocol, these may be configured for active/standby or active/active redundancy.

With the rise of virtualization in the data center, the physical design of the network and its logical representation are increasingly different. Virtualization encourages workload mobility and this introduces inefficiencies, given that default gateway placement was predicated upon the physical location of network resources. Traffic forwarding continues to function, however, the inherent inefficiency created by traffic hair-pinning is suboptimal.

Distributed Anycast Gateway

The use of the MP-BGP EVPN control plane introduces Distributed Anycast Gateway functionality. In this model, the default gateway function is fully distributed across all leaf nodes within the VXLAN Fabric. Leveraging the Distributed Anycast Gateway function provides improved efficiency and higher cross-sectional bandwidth while eliminating the need to run a First Hop Redundancy Protocol (FHRP). Furthermore, routed traffic between workloads connected to the same leaf is locally forwarded without having to be sent to the spine layer. By decreasing hop count, the Distributed Anycast Gateway greatly reduces network latency.

Integrated Routing and Bridging

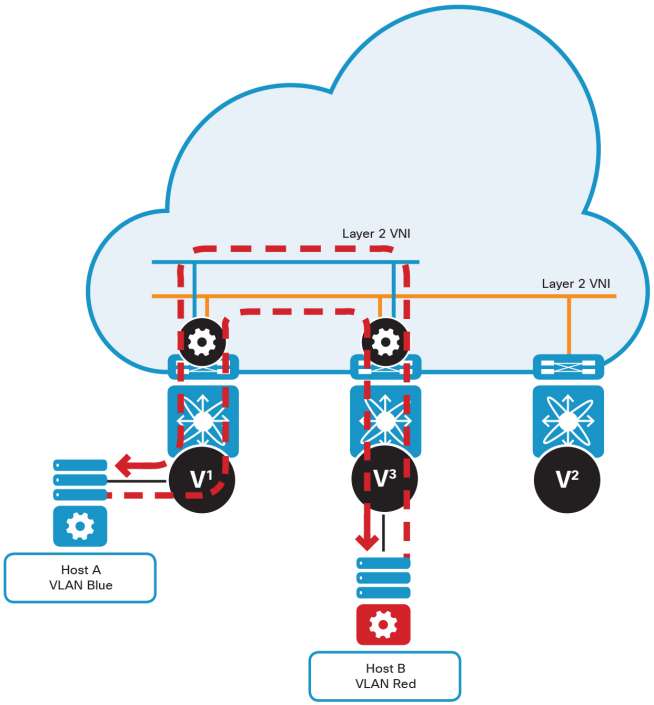
EVPN VXLAN Fabrics introduce Integrated Routing and Bridging (IRB) functionality, which offers the capability of both Layer 2 and Layer 3 forwarding directly at the leaf switch. This is fundamental to the Distributed Anycast Gateway function which provides a distributed default gateway capability closest to the endpoints.

Asymmetric vs Symmetric Forwarding

The EVPN draft defines two different methods for routing traffic between VXLAN overlays. The first method is referred to as asymmetric IRB and the second is known as symmetric IRB.

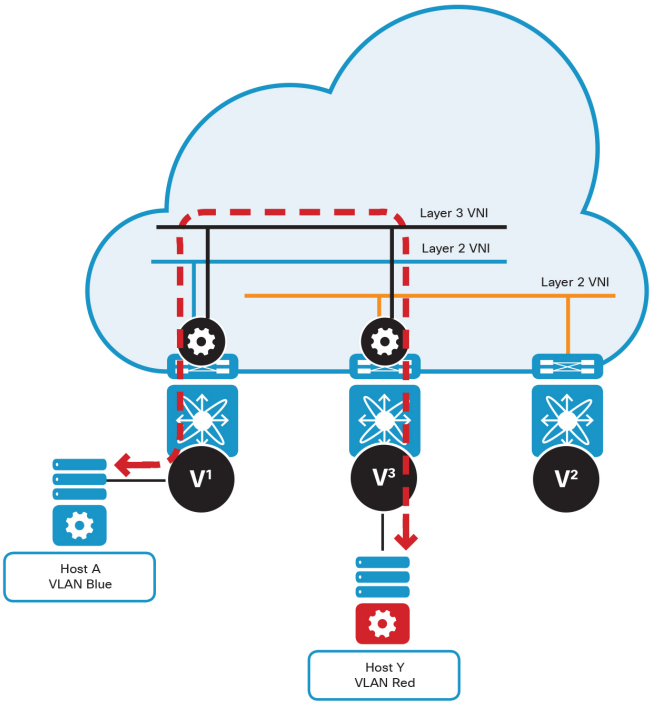
With asymmetric IRB, the ingress VTEP is performing both routing and bridging, whereas the egress VTEP is only performing bridging. As a result, the return traffic will take a different VNI than the source traffic. This necessitates that the source and destination VNIs reside on both the ingress and egress VTEPs. This leads to a more complex configuration as all switches need to be configured for all possible VNIs. Perhaps a more pressing consideration is the scaling implications of all devices potentially needing to learn a considerably larger number of endpoints.

Figure: Asymmetric IRB



In symmetric IRB, both the ingress and egress VTEP provide both L2 and L3 forwarding. This results in predictable forwarding behavior. As a result, only the VNIs of locally-attached endpoints need to be defined in a VTEP (plus the transit L3 VNI), which in turn simplifies configuration and reduces scale requirements through optimized use of ARP and the MAC address table. This results in better scale in terms of the total number of VNIs a VXLAN Fabric can support.

Figure: Symmetric IRB



It is important to keep in mind that as both methods are defined in the standard, consideration must be given to device selection and the implications for interoperability. For example, Cisco supports only symmetric IRB on the Nexus platforms as it offers better scalability.

Software Overlays

Introduction

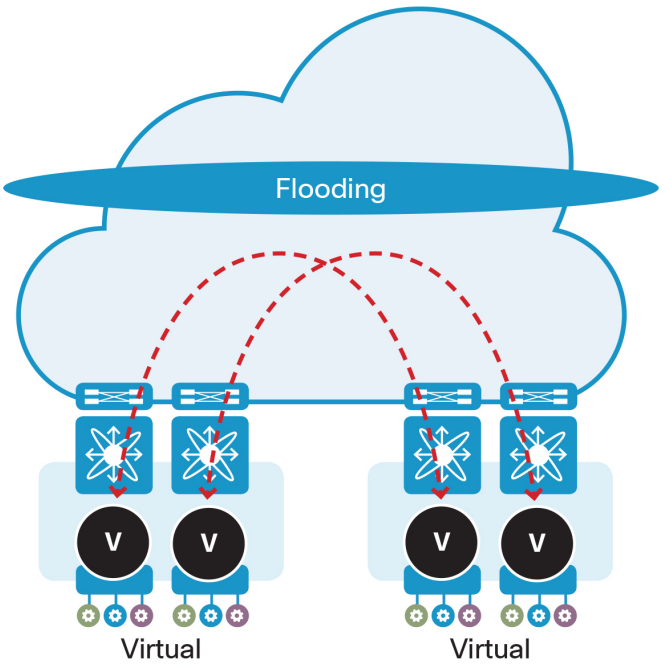
Server virtualization has transformed the way in which data centers are operated, and the vast majority of data centers today implement it to some degree. However, making the assumption that those data centers run exclusively virtualized workloads would be a mistake. Many organizations still make use of mainframes, for example. Moreover, new applications that do not require server virtualization are coming into the main stage, such as cloud-based software that makes use of Linux containers, or modern scale-out applications such as Big Data, that deliver operational benefits and scale without the need of a hypervisor.

Although VXLAN is a generic overlay concept that is commonly deployed in the network, it is sometimes associated with server virtualization and hypervisors. This chapter covers the advantages and disadvantages of implementing VXLAN on virtualized hosts, and how to realize the most benefit out of this technology, keeping in mind that one of the main reasons for interest in VXLAN is its openness, that avoids vendor lock-in (vendor or hypervisor).

Host-Based Overlay

Server virtualization offers significant benefits including flexibility and agility in delivering compute services in the data center. Traditionally, networking to the hypervisor is provided via VLAN transport, and there is a new trend to adopt host-based VXLAN overlays to improve agility and automation of the network layer.

Figure: Host-Based Overlay



The host-based overlay typically runs between host VTEPs over an IP transport and offers ease of deployment and automation capability, empowering the server team to deliver virtual networking services without needing to involve the network team. This ability to automate networking directly through the Virtual Machine Manager (VMM) as

a software only overlay often results in a sub-optimal network solution which does not take into account the broader aspects of operations, integration, and performance for the network as a whole.

In addition to the CPU impact introduced with host-based overlays, the network team has to provide extra efforts in troubleshooting due to the lack of correlation between the overlay and the underlay networks. In regards to CPU impact, the performance of a software VTEP is dependent on CPU and memory available on the hypervisor. Some implementations run the VTEP function in kernel space, others in user space. Both options must deliver the necessary packet processing required for efficient application delivery. These solutions typically struggle to deliver line-rate throughput even with hardware assistance at the server NIC.

Additionally, host-based overlay network solutions are primarily focused on networking for virtual servers without consideration for physical workloads or other existing services inside or outside the data center. Connectivity to both physical servers and resources beyond the virtual network typically require gateways, either in software or hardware which must be integrated with the physical network.

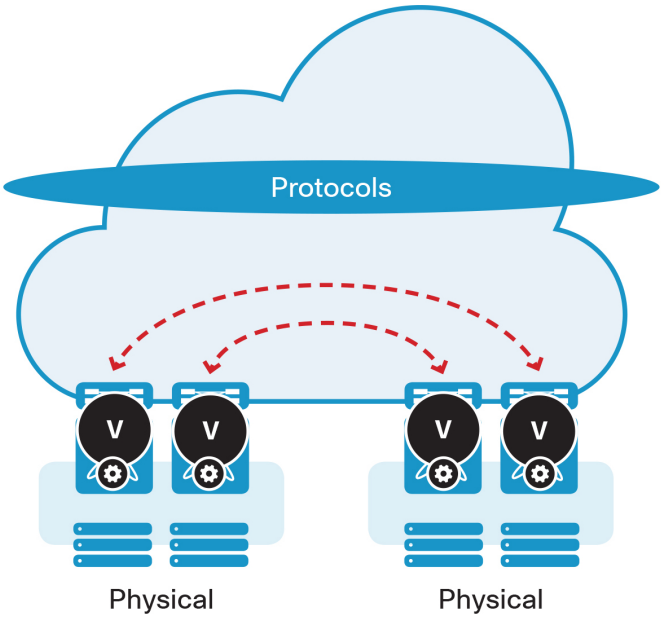
In summary, when evaluating host-based overlay solutions, it is critical to consider the broader business and technical implications for the data center including licensing cost, compute overhead, performance penalty, additional gateway infrastructure requirements, and the impact to network operations.

An Alternative to Host-Based Overlays

A network-based overlay deploys network switches as VXLAN tunnel endpoints (VTEPs). Compared to host-based overlay solutions, network overlays deliver hardware-accelerated encapsulation provided by ASICs, which is the core of a network switch. By offloading the VXLAN overlay functions to the network switch, a simple VLAN-based deployment can be used to connect physical or virtual workloads to the VXLAN Fabric. By removing the burden of data traffic forwarding and encapsulation from the hypervisor, CPU resources are freed. In other words, the hypervisor can allocate all available hardware resources to its key function, serving Virtual Machines (VMs) and applications.

With a VXLAN EVPN Fabric and the associated operation and management tools, it is possible to deliver the flexibility of automated network provisioning for virtual machines while overcoming some of the previously discussed limitations affecting the host-based overlay model. Deploying a network-based overlay does not have an impact on the overall network performance, visibility, and troubleshooting of network issues.

Figure: Network-Based Overlay



The VM Tracker (VMT) function on Nexus leaf switches provides visibility of the hypervisor hosts and the VMs connected to the VXLAN Fabric so the network can take decisions upon that information. For example, the VM Tracker auto-config feature enables automated provisioning of network resources to support virtual machines in a VMware vSphere environment. VM Tracker communicates with VMware vCenter Server to retrieve information relating to the virtual network configuration. The information includes:

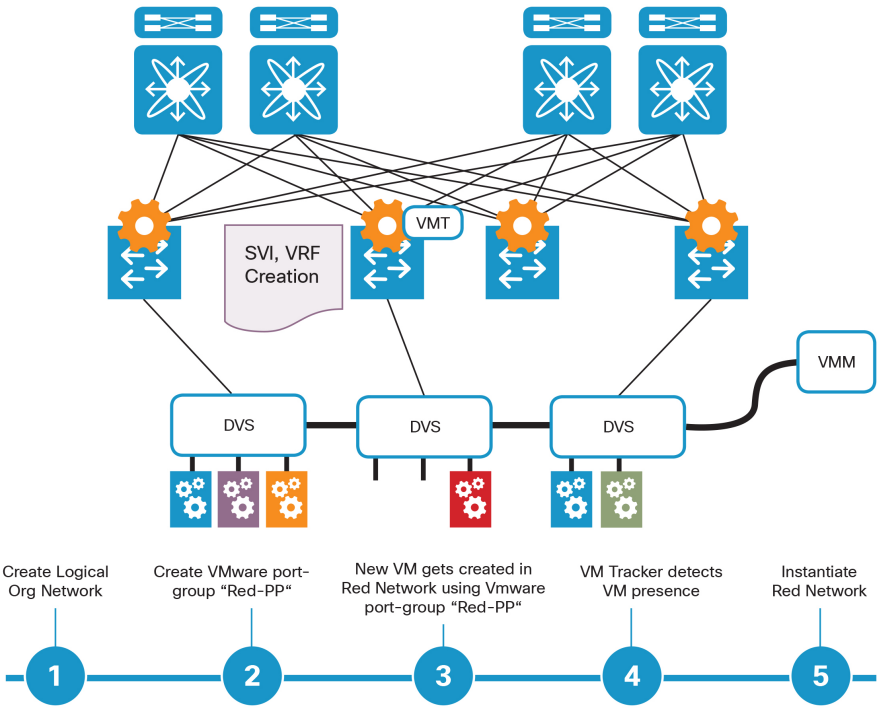
- vSphere ESX host to VM mappings
- VM state

- Physical port attachments
- VDS port groups assignment to VM

The network configuration that is dynamically provisioned, depending on the previous information, includes the following attributes:

- VLAN provisioning
- VNI allocation
- L3 gateway
- VRF provisioning

Figure: vCenter Integration for Dynamic Network Configuration



For example, proper VLAN and VNI configuration is deployed on a specific leaf when the first VM associated to that VNI is locally connected and removed when the last VM is migrated to a different server or powered off. With such functionality, the server administrator can achieve the agility required to quickly deploy VMs, and at the same time provision network resources in a dynamic manner.

This solution provides the performance of hardware-based encapsulation without having to upgrade the host physical NICs. This is especially relevant in the case of virtual network functions. For example, certain host-based overlays use a virtual gateway to interact with the rest of the world, as already mentioned in a previous section. The performance discussion is critical in this case, because that single virtual gateway might become a bottleneck for the whole virtual environment.

In conclusion, by leveraging network-based overlays it is possible to achieve line-rate throughput for both east-west and north-south traffic flows while eliminating the need for software gateways. At the same time, the network administrator gains visibility of the virtual infrastructure attached to the Fabric, assisting with the ongoing operations and troubleshooting of the environment. Finally, VM Tracker functionality ensures that the virtualization administrator gains network agility on top of the benefits that server virtualization already provides.

Hybrid Overlays with VXLAN EVPN

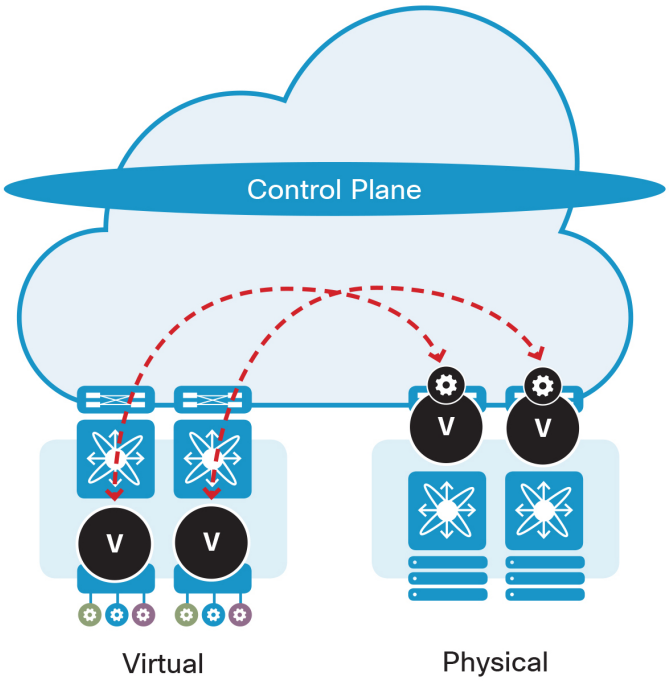
As previously discussed, pure host-based overlays bring little value to data centers, but there are situations where a hybrid approach might solve some challenges or use cases. Service Providers have very specific requirements regarding network management and operations including:

- Support for a mix of software and hardware VTEPs
- Integration with the hypervisor layer
- Support multi-vendor Fabrics
- Overlay and underlay are operated by different teams

Hybrid VXLAN overlays consist of both host-based software VTEPs and switch-based hardware VTEPs. A cohesive operational and management model is needed to integrate the two types of VTEP together. Cisco Virtual Topology System (VTS) is an example of such a solution.

Further details about the Cisco Virtual Topology System are provided in the Management And Operations chapter, but below is a brief summary of Cisco VTS architecture.

Figure: Hybrid Overlays



Cisco Virtual Topology Systems (VTS) provisions hardware and software VTEPs. The ability to integrate a VXLAN software-based VTEP allows the deployment of the VXLAN technology on top of legacy network hardware or to complement hardware-based VTEP deployments.

The Virtual Topology Controller (VTC) is the single point of management for hybrid overlays to configure, manage and operate a VXLAN Fabric with MP-BGP EVPN control plane. The management layer supports integration with hypervisors such as VMware vSphere or Openstack/KVM so that network constructs can be directly provisioned from the hypervisor User Interface. The northbound REST APIs enable integration with third party tools.

The control plane is represented by a virtualized IOS-XR router to provide integration with MP-BGP EVPN and advertise reachability information to the software VTEP itself over an API. The software VTEP named Virtual Topology Forwarder (VTF) provides VXLAN encapsulation capability in the hypervisor.

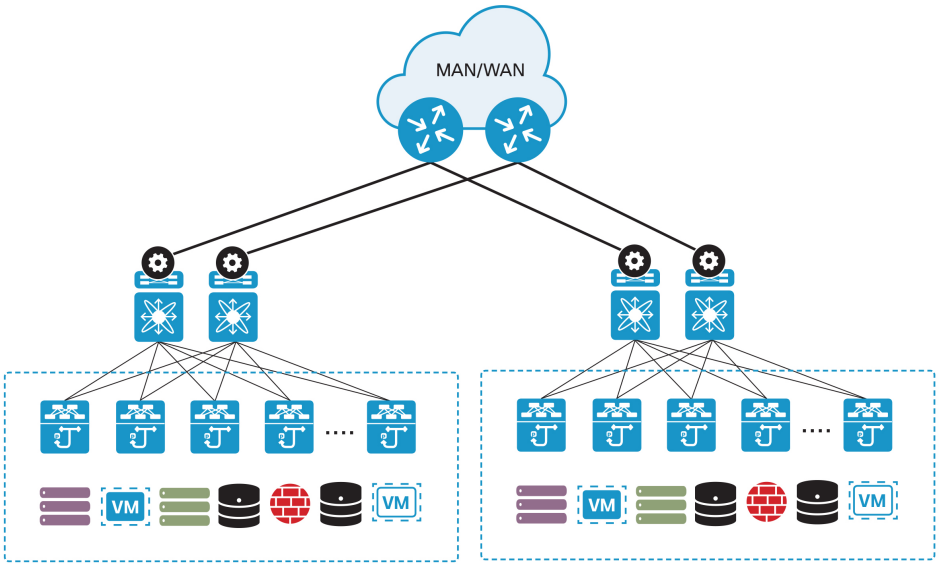
More details on Cisco Virtual Topology System architecture are available at https://www.cisco.com/go/vts

Single-POD VXLAN Design

Introduction

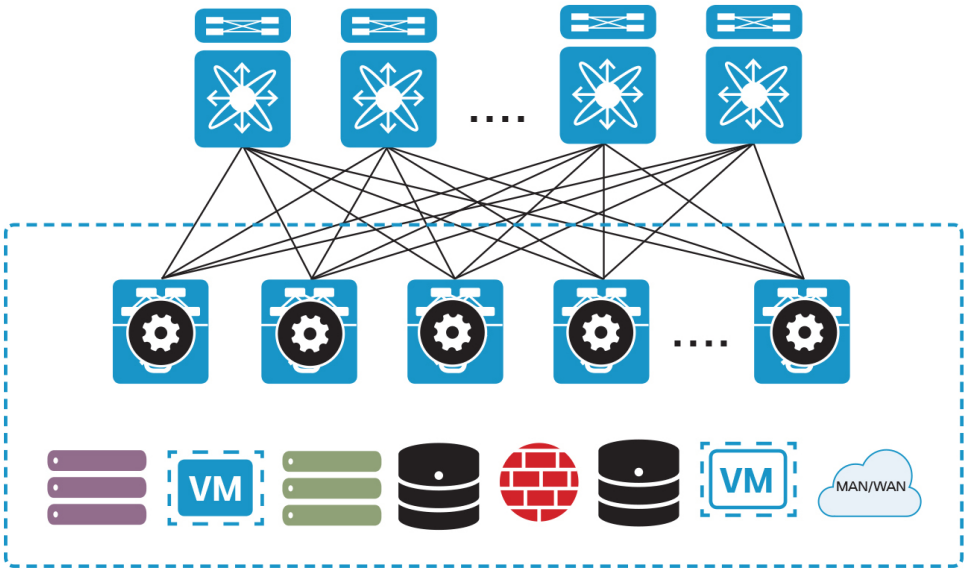
In classic hierarchical network designs, the access and aggregation layers together provide Layer 2 and Layer 3 functionality as a building block for data center connectivity. In smaller data center environments, this single building block would provide sufficient scale to meet the entire demands for connectivity and performance. As the environment scales to meet the increased demands of the larger data center, this building block is typically replicated with an additional core layer introduced to connect these together. These building blocks are commonly referred to as a Point of Delivery, or POD, and allow for consistent, modular scale as the environment grows.

Figure: Hierarchical Network Design



When designing a VXLAN Fabric, a single-POD also defines a single VXLAN Fabric based on a scalable spine-leaf architecture as shown in the diagram below.

Figure: VXLAN Fabric



A single VXLAN POD can scale to hundreds of switches and thousands of ports which will meet the demands of many enterprise data center environments; however, to meet more complex or larger scale requirements, the VXLAN POD may be replicated in the form of a multi-POD design. In a typical deployment with multiple data center locations, these VXLAN Fabrics, whether single or multi-POD- based, will be deployed together as a multi-site VXLAN design. Both the multi-POD and multi-site deployment types are described further in the multi-POD and multi-site Designs chapter. Additionally, the connectivity of Layer 2 and Layer 3 to the external network domain is covered in the External Connectivity for the VXLAN Fabric chapter.

This chapter explores the design considerations for building a single VXLAN POD comprising the underlay network foundation and the overlay network together with their associated data and control planes, as well as guidelines for endpoint connectivity to the Fabric.

Underlay

In building a VXLAN EVPN Fabric, it is essential to construct an appropriate underlay network as this will provide a scalable, available and functional foundation to support the overlay. This section includes important considerations for the underlay design.

Routed Interface Considerations

MTU

In order to improve the throughput and network performance, it is recommended to avoid fragmentation and reassembly on network devices performing VXLAN encapsulation and decapsulation. It is therefore required to increase the maximum transmission unit (MTU) in the transport network by at least 50 bytes (54 if an 802.1Q header is present in the encapsulated frame). If the overlay uses a 1500-byte MTU, the transport network needs to be configured to accommodate 1550 byte (1554 bytes if including the 802.1Q header) frames as a minimum. Jumbo frame support in the transport network is strongly recommended if the overlay applications use frame sizes larger than 1500 bytes.

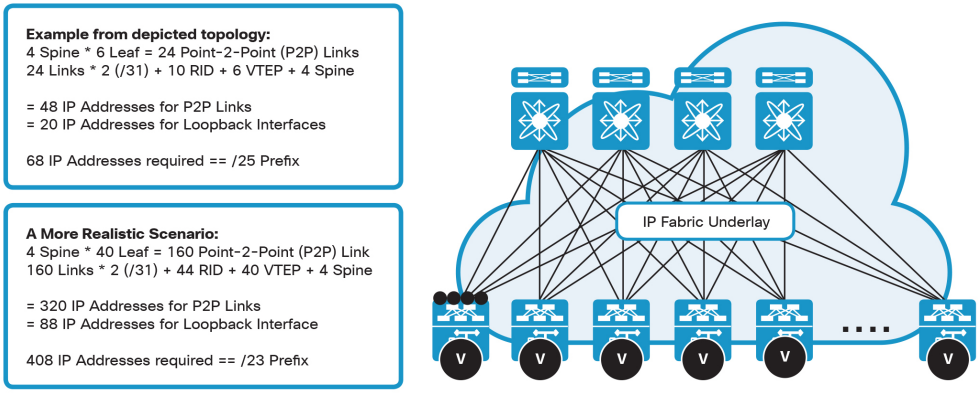
In order to ensure that VXLAN encapsulated packets can be successfully carried across the Fabric, the increase of MTU must be configured on all the Layer 3 interfaces connecting the Fabric nodes.

Routed Interface Addressing

The connectivity between network devices in a VXLAN Fabric typically leverage routed point-to-point interfaces which can be simply addressed with a /30 or even a /31 subnet mask. In a large data center Layer 3 underlay network, there will be many routed links, leading to high IP address consumption.

Following are the IP address requirement for a couple different scenarios.

Figure: IP Address Requirement



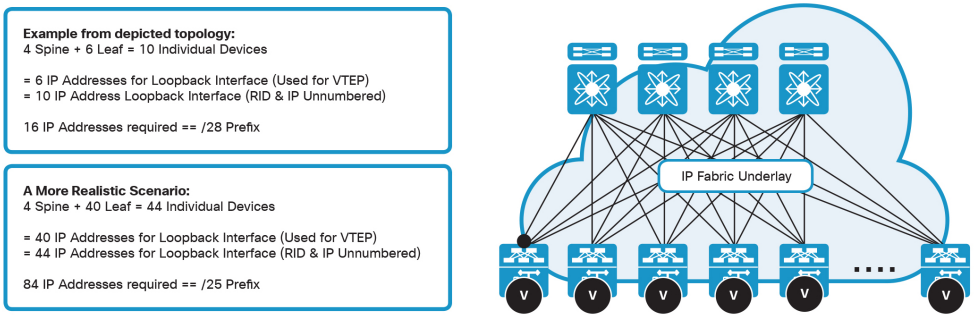
In the example above, in a small network with 4 spine and 6 leaf switches, there will be a minimum of 24 point-to-point links, requiring a total of 68 addresses for the Fabric underlay. This number will exponentially increase to 408 in a larger scale scenario of 4 spine and 40 leaf switches.

A recommended approach is to use IP unnumbered for the interface IP address configuration requiring only a single IP address per device, regardless of the number of links deployed. As shown below, in the smaller scale example the total number of IP addresses consumed would be reduced to 16 for the entire Fabric underlay. When expanding the network, the IP address requirement will increase linearly with the number of devices.

Loopback Interface Addressing

As highlighted in the examples below, each leaf switch with a VTEP should have a minimum of two loopback interfaces. The first loopback is used as Router-ID (RID) and for assigning an IP address to the unnumbered Layer 3 links. The second loopback represents the VTEP IP address used as source and destination for VXLAN encapsulated traffic.

Figure: IP Address Requirement with IP Unnumbered



Routing Protocol Considerations

The choice of routing protocol for the underlay network has numerous options, however, it is typically determined by what protocols are already in use and are familiar to the network administrator. In making this decision, it is important to consider the protocol convergence characteristics as this will determine the overall speed of convergence of the overlay network. Specifically, Open Shortest Path First (OSPF) and Intermediate System - Intermediate System (IS-IS) are two types of Interior Gateway Protocol (IGP) that are particularly suitable for multi-stage spine-leaf Fabrics. As the spine-leaf design inherently provides multiple paths between leaf switches via the spine, SPF-based protocols will compute a topology consisting of multiple equal cost paths through the network and provide rapid convergence around failures.

While BGP also has merit as an underlay routing protocol, it is a Path Vector Protocol and primarily considers Autonomous Systems (AS) to calculate paths. Despite this, an experienced network engineer can manipulate BGP to achieve comparable convergence outcomes to SPF-based routing protocols by leveraging the many attributes and options available. The main perceived advantage of using BGP in the underlay is having only one routing protocol in use across the entire network (underlay + overlay). While this offers simplification, potential disadvantages exist due to the additional configuration required. Since the overlay predominantly uses a single path from VTEP to VTEP, it is assumed that the underlay provides multi-path forwarding. This is not the default

forwarding behavior of BGP, therefore, specific attention is needed to achieve equivalent multi-pathing as would be achieved when using SPF-based IGPs in the underlay.

When selecting routing protocols for use in the underlay, it is imperative to consider how the overlay control plane protocol functions and should be configured. By using the same protocol for the underlay and overlay, a clear separation of these two domains can become blurred. Therefore, when designing an overlay network, it is a good practice to independently build a transport network as has been done in MPLS. The deployment of an IGP in the underlay offers this separation of underlay and overlay control protocol. This provides a very lean routing domain for the transport network that consists of only loopback and point-to-point interfaces. At the same time, MAC and IP reachability for the overlay exists in a different protocol, namely MP-BGP EVPN.

OSPF Deployment Recommendation

OSPF is a link-state routing protocol commonly used in enterprise environments. The OSPF default interface type used for Ethernet interfaces is "Broadcast," which inherently results in a Designated Router (DR) and/or Backup Designated Router (BDR) election thus reducing routing update traffic. While this is fine in a Multi-Access network (such as a shared Ethernet segment), it is unnecessary in a point-to-point network.

In a point-to-point network, the "Broadcast" interface type of OSPF adds a DR/BDR election process and an additional Type 2 Link State Advertisement (LSA). This results in unnecessary additional overhead, which can be avoided by changing the interface type to "point-to-point". In this way, the DR/BDR election process can be avoided, reducing the amount of time to bring up the OSPF adjacency between the leaf and spine switches. In addition, with the point-to-point interface mode, the need for Type-2 LSAs is removed with only Type-1 LSA needed since there is no Multi-Access (or Broadcast) segment present. As a result, the OSPF LSA database remains lean.

IS-IS Deployment Recommendation

Another standard based IGP routing protocol is Intermediate System – Intermediate System (IS-IS). This link state routing protocol is gaining popularity with fast convergence in a large-scale environment although has primarily been deployed in service provider environments. IS-IS uses Connectionless Network Protocol (CLNP) for communication between peers and doesn't depend on IP. There is no SPF calculation on link change and SPF calculation only happens when there is a topology change which helps with faster convergence and stability in the underlay. No significant tuning is required for IS-IS to achieve an efficient, fast converging underlay network.

IP Multicast Recommendation

IP multicast provides an efficient mechanism for the distribution of multi-destination traffic in the Fabric underlay.

To deploy IP multicast in the underlay, a Protocol Independent Multicast (PIM) routing protocol needs to be enabled and must be consistent across all the devices in the underlay network. The two common PIM protocols are Sparse-Mode (PIM-ASM) and Bidirectional (PIM-Bidir). This implies the requirement to deploy rendezvous Points (RPs).

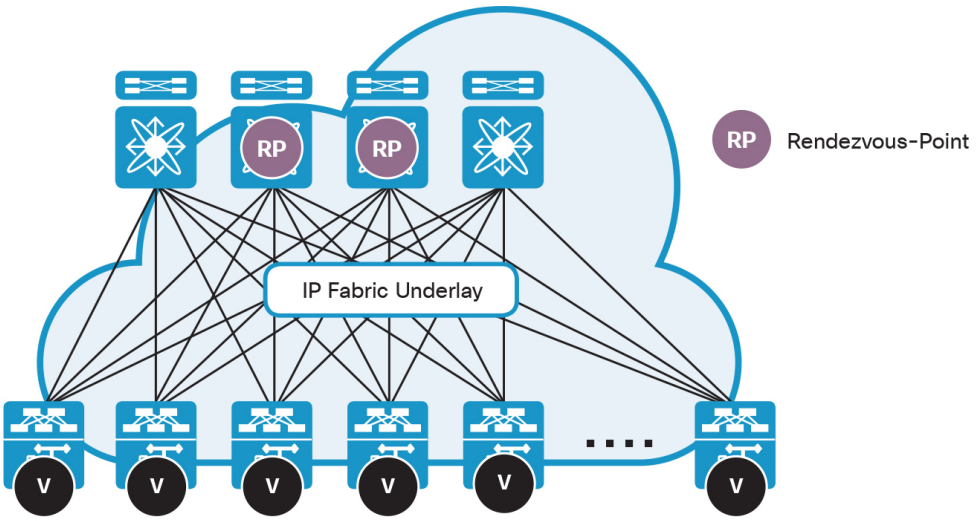
Multicast Rendezvous Point (RP) Consideration

Several methods are available to achieve a highly available RP deployment, including for example the use of protocols such as auto-RP and Bootstrap. However, to improve the convergence experience in the RP failure scenario, the recommendation is to deploy Anycast RP, which consists of using a common IP address on different devices to identify the RP. Simple static RP mapping configuration is then applied to each node in the Fabric to associate multicast groups to the RP, so that each source or receiver can then utilize the local RP that is the closest from a topological point of view.

It is important to remember that the VTEP nodes represent the sources and destinations of the multicast traffic used to carry BUM traffic between endpoints connected to those devices.

Normally, the RPs would be deployed on the spine nodes, given the central position those devices play in the Fabric.

Figure: Multicast RP Placement



When deploying Anycast RP, it is critical to synchronize information between the different RPs deployed in the network, as it may happen that sources and receivers join different RPs, depending where they are connected in the network. Two mechanisms are supported on Cisco Nexus platforms to synchronize state information between RPs:

- Multicast Source Discovery Protocol (MSDP): this option has been around for a long time and it is widely available across different switches and routers. MSDP sessions are established between RP devices to exchange information about source and receivers for each given multicast group
- PIM with Anycast RP: this option is currently supported only on Cisco Nexus platforms and leverages PIM as control plane to synchronize state between RPs

Ingress Replication

Ingress replication, also known as Head-End replication, may be used as an alternative to IP multicast to carry the BUM traffic inside the Fabric. One reason for using this alternate method is that IP multicast is not always an available option due to hardware and software constraints. IP multicast may also not be preferred due to perceived complexity by the network operations team.

When deploying ingress replication it is important to consider the overall scale of the Fabric and the amount of multi-destination traffic expected in the environment. This is because for VXLAN EVPN ingress replication, the VXLAN VTEP uses a list of IP addresses of other VTEPs in the network to send BUM traffic as unicast traffic, creating multiple copies of the same traffic type. It is worth noticing that the deployment of the MP-BGP control plane enables the list of VTEPs connected to the same VXLAN Fabric to be dynamically built. These IP addresses are exchanged between VTEPs through the BGP EVPN control plane.

```
interface nve1
    no shutdown
    source-interface loopback0
    host-reachability protocol bgp
    member vni 30000
      ingress-replication protocol bgp
    member vni 30001
      ingress-replication protocol bgp
```

As shown in the configuration sample above, the ingress replication mode is configurable on a per-L2VNI. It is not possible to mix multicast and ingress replication for the same L2VNI in the same VXLAN Fabric.

Overlay

After building a solid foundation for the VXLAN network with the underlay, the overlay concepts are equally important to provide the required functionality and flexibility.

VXLAN EVPN Control Plane

As an industry standard overlay technology, VXLAN has seen increasing adoption in the data center space. EVPN is the control plane for VXLAN and provides an efficient method for route learning and distribution in the VXLAN overlay network. The routing information includes Layer 2 MAC routes, Layer 3 Host IP routes, and Layer 3 subnet IP routes. EVPN control plane also introduces multi-tenancy support to the VXLAN overlay network, as well as a VTEP peer discovery, security, and authentication mechanism. This section is intended to provide a deeper understanding of the VXLAN EVPN control plane.

MP-BGP EVPN

EVPN uses MP-BGP as the routing protocol to distribute reachability information for the VXLAN overlay network, including endpoint MAC addresses, endpoint IP addresses, and subnet reachability information.

EVPN is another MP-BGP address family leveraging similar constructs as the VPNv4 address family traditionally deployed in MPLS VPN architectures. Those constructs include VRFs, Route Distinguishers (RD) and Route Targets (RT). The peculiarity of the EVPN control plane when compared to VPNv4 is the capability of exchanging not only IP but also MAC address information.

Virtual Routing and Forwarding (VRF)

Virtual Routing and Forwarding (VRF) defines the Layer 3 routing domain for each tenant supported in the VXLAN Fabric. In VXLAN EVPN networks, each tenant VRF has a Layer 3 VNI used as a virtual backbone for routing within the VRF.

Route Distinguisher (RD)

Route Distinguisher (RD) is the identifier of a VRF since each VRF has its own unique RD in the network. When an EVPN advertisement is sent out to the peers, the RD of the VRF to which this route belongs is prepended to the original route itself to render it unique within the network. This allows different VRFs to use overlapping IP addresses so that different tenants can have true autonomy for IP address management. The RD can be automatically defined to simplify configuration.

Route Target (RT)

Route Target (RT) is an extended attribute in EVPN route updates used to control route distribution in a multi-tenant network. EVPN VTEPs have an import RT setting and an export RT setting for each VRF and each L2VNI. When a VTEP advertises EVPN routes, it affixes its export RT in the route update. The routes will be received by other VTEPs in the network. These devices will compare the RT value carried with the route against their own local import RT setting. If the two values match, the route will be accepted and programmed in the routing table. Otherwise, the route will not be imported. The RT can be automatically defined to simplify configuration.

EVPN Route Types

The EVPN control plane advertises different types of routing information:

- Type-2 - Endpoint reachability information, including MAC and IP addresses of the endpoints
- Type-3 - Multicast route advertisement-announcing capability and intention to use Ingress Replication for specific VNIs
- Type-5 - IP prefix route used to advertise internal IP subnet and externally learned routes onto the VXLAN Fabric

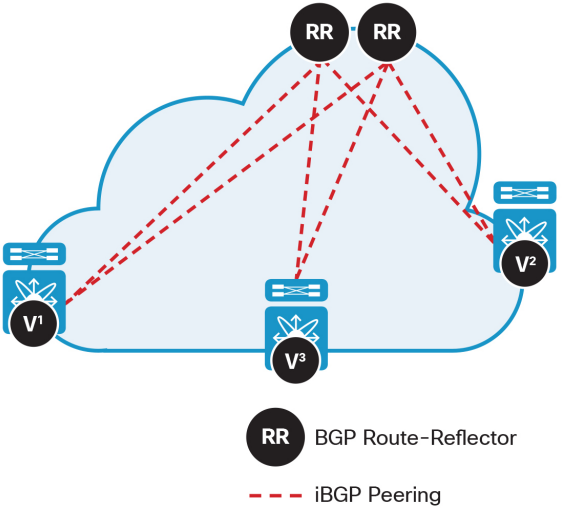
The EVPN route update also includes the following information:

- VNID for the L2VNI and VNID for the L3VNI for the tenant VRF
- BGP next-hop IP address identifying the originating VTEP device
- Router MAC address of the originating VTEP device

Route Reflector Placement

As discussed in the previous chapter, iBGP is the most common routing protocol deployed for the EVPN control plane in VXLAN Fabrics. With iBGP, there is a requirement to have a full mesh between all of the iBGP speakers. To help scale and simplify the iBGP configuration, it is recommended to implement iBGP Route Reflectors (RR). The placement of the iBGP route reflectors is recommended to be implemented on the spines as they are central to all of the leaf switches. In this case, two of the spines will have BGP route reflector configured and all of the leaf switches will be configured as the BGP route reflector clients. The route reflector will reflect EVPN routes for the VTEP leaf switches.

Figure: iBGP Route Reflector Placement



Endpoint Detection and Tracking

A VTEP in MP-BGP EVPN detects attached endpoints via local learning. MAC addresses are learned in the data plane from the incoming Ethernet frames whereas the IP address is learned via ARP or Gratuitous ARP (GARP) control plane packets sent by the endpoint. Alternately, the learning can be achieved by using a control plane or through management plane integration between the VTEP and the local hosts.

Once a VTEP detects its local endpoints, it will install a Host Mobility Manager (HMM) route to track it. The VTEP will also construct an EVPN Type-2 route to advertise the learned MAC and IP address of the endpoint to the rest of the VTEPs in the same Fabric.

The EVPN Type-2 route has an embedded sequence number used for endpoint movement tracking. When an endpoint moves from one VTEP to another VTEP, the new VTEP will detect it as a newly attached local host. It will send a new EVPN Type-2 routing update with the reachability information for this endpoint. When doing so, it will increment the sequence number by one. When the rest of VTEPs receive the new route with the higher sequence number they will update their routing information for the endpoint using the new VTEP as the next hop.

Layer 2 Logical Isolation (Layer 2 VNIs)

The creation of VXLAN overlay networks provides 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 Layer 2 Virtual Network Instance (L2VNI).

The VXLAN segments are independent of the underlying network topology; likewise, the underlying IP network between VTEPs is independent of the VXLAN overlay. The combination of locally defined VLANs and their mapping to associated L2VNIs allows the creation of Layer 2 logical segments that can be extended across the Fabric.

As with traditional VLAN deployments, communication between endpoints belonging to separate L2VNIs is possible only through a Layer 3 routing function.

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

```
vlan 100
  vn-segment 30000
vlan 101
  vn-segment 30001
```

Once the VLAN-to-VNI mappings have been defined, it is then required to associate those created L2VNIs to an NVE logical interface, as shown in the configuration sample below.

```
interface nve1
  no shutdown
  source-interface loopback0
  host-reachability protocol bgp
  member vni 30000
    suppress-arp
    mcast-group 239.239.239.100
  member vni 30001
    suppress-arp
    mcast-group 239.239.239.101
```

In the definition of the NVE logical interface, the loopback interface created as part of the underlay configuration is specified to be used for VXLAN encapsulation and decapsulation.

It is also required to associate the EVPN control plane to the VXLAN deployment, instead of the original flood and learn model. At the time of writing, this configuration has a global scope for a given VXLAN deployment, hence, it is not possible to mix the two modes of operation (control plane or flood and learn based) in the same Fabric.

When multicast is the deployment choice for handling the replication of BUM traffic, a specific multicast group is associated to each defined L2VNI. The assignment of multi-

cast groups to the L2VNIs is quite flexible and the chosen configuration depends on the following considerations:

- Using a unique multicast group for each defined VNI would allow the most granular distribution of BUM traffic, which will be only flooded to the leaf nodes where that specific L2VNI is defined. On the other side, this design choice would drastically increase the amount of multicast state in the Fabric leaf and spine devices.
- Using a common multicast group for all the defined L2VNIs would reduce at a minimum the amount of multicast state in the core of the network, but would cause BUM traffic for a given L2VNI to be flooded to all the leaf nodes even where that specific VNI is not present (the traffic would then be discarded by the leaf).

The generally recommended approach is a balance of the two options above and suggests to assign a common multicast group to all the L2VNIs defined for a given tenant (VRF); different multicast groups can instead be used across tenants.

Finally, as part of the L2VNI configuration, it is possible to enable ARP suppression. This removes the need to flood ARP requests across the Fabric, which usually represents the large majority of L2 broadcast traffic. ARP suppression can be enabled since each leaf node learns about all the endpoints connected to the Fabric via the EVPN control plane. When receiving an ARP request originated by a locally connected endpoint trying to identify the MAC of the remotely connected endpoint, the leaf can then perform a lookup in a local cache populated upon reception of EVPN updates. If the MAC/IP information for the remote endpoint is available, the leaf can then reply to the local endpoint with the ARP mapping information on behalf of the remote endpoint. If the MAC/IP information for the remote endpoint is not available, the ARP request is flooded across the Fabric by encapsulating the packet in a VXLAN frame destined to the multicast group associated to the L2VNI of the local endpoint. ARP suppression can also be enabled or disabled on a per L2VNI basis.

Because most endpoints send ARP requests to announce themselves to the network right after they come online, the local VTEP will immediately have the opportunity to learn their MAC and IP addresses and distribute this information to other VTEPs through the MP-BGP EVPN control plane. Therefore, most active IP hosts in VXLAN EVPN should be learned by the VTEPs either through local learning or control plane-

based remote learning. As a result, ARP suppression reduces the network flooding caused by host ARP learning behavior.

Layer 3 Multi-Tenancy (VRFs and Layer 3 VNIs)

The logical Layer 2 segment created by mapping a locally significant VLAN with a globally significant L2VNI is normally associated with an IP subnet. When endpoints connected to the L2VNI 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 functionality on each leaf node, a deployment model commonly referred to as Distributed Anycast Gateway. In a VXLAN deployment, the various Layer 2 segments defined by combining local VLANs and global VNIs can be associated to a VRF if they need to communicate.

Communication between local endpoints connected to different L2VNIs can occur via normal Layer 3 routing in the context of the VRF (i.e. no VXLAN encapsulation is required).

The deployment of Symmetric Integrated Routing and Bridging (IRB), already introduced in the Fundamental Concepts chapter, requires the introduction of a transit Layer 3 VNI (L3VNI) offering L3 segmentation services per tenant VRF. Each VRF instance is mapped to a unique L3VNI in the network. Different L2VNIs for the same tenant are usually associated to the same VRF. As a result, the inter-VXLAN routing is performed throughout the L3VNI within a particular VRF instance.

The Symmetric IRB model assumes that the default gateway for all the L2VNIs is fully distributed to all the leaf nodes. At the time of this writing, the distributed gateway model is the only one supported with VXLAN EVPN and can be enabled by applying the configuration below on all the leaf nodes:

```
fabric forwarding anycast-gateway-mac 2020.2020.2020

vlan 100
 vn-segment 30000

interface Vlan100
  no shutdown
  vrf member Tenant-1
  ip address 192.168.100.1/24 tag 21921
  fabric forwarding mode anycast-gateway
```

The first command defines a common virtual MAC address to be used for the default gateway. The same value is used for all the IP subnets associated with the L2VNI segments, independently from the VRF they belong to (fabric-wide configuration). The "fabric forwarding mode anycast-gateway" command is used to enable the distributed default gateway functionality on the VTEP nodes. This command must be applied to all the SVIs for the VLANs that are mapped to L2VNIs.

The configuration shown above must be repeated for all the local VLANs mapped to L2VNIs if there is a requirement to route traffic to separate IP subnets. The SVI is associated to a specific VRF instance ("vrf member" command). The use of VRFs provides Layer 3 logical isolation, a concept often referred to as "multi-tenancy".

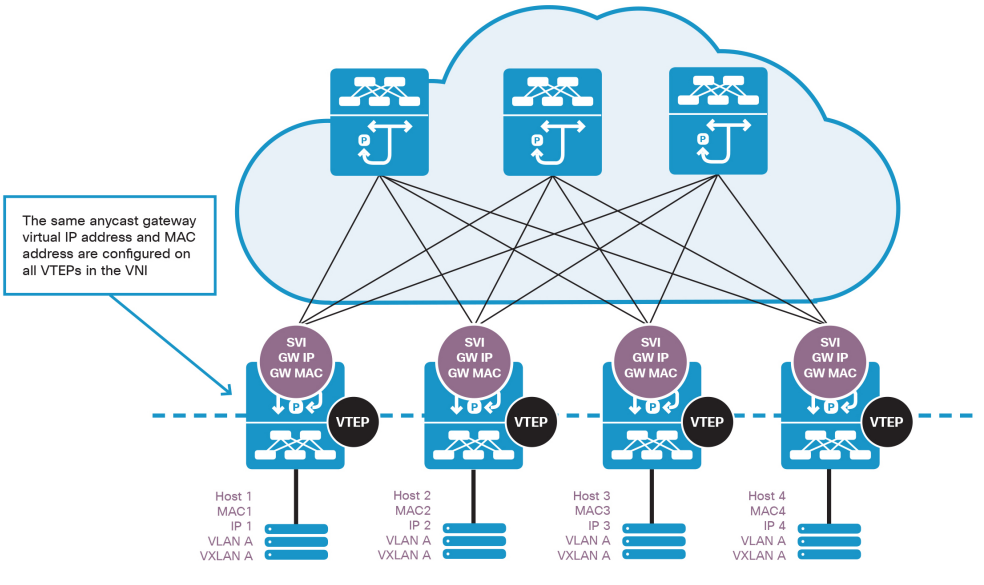
The additional required configuration for each defined VRF is shown below.

```
vlan 2500
  name L3_Tenant1
  vn-segment 50000

vrf context Tenant-1
  vni 50000
  rd auto
 address-family ipv4 unicast
    route-target import auto
    route-target import auto evpn
    route-target export auto
    route-target export auto evpn

interface 2500
 description L3_Tenant1
  no shutdown
  mtu 9216
  vrf member Tenant-1
  ip forward

interface nve1
  member vni 50000 associate-vrf
```

In MP-BGP EVPN, any VTEP in a VNI can be the Distributed Anycast Gateway for end hosts in an IP subnet by supporting the same virtual gateway IP address and virtual gateway MAC address. When using Distributed Anycast Gateway with EVPN, routed traffic from an endpoint is always processed by the closest leaf node. This capability enables optimal forwarding for northbound traffic from endpoints in the VXLAN overlay network. East-West traffic between endpoints connected to the same leaf is locally routed by that leaf. This is especially important for applications with rack awareness like some Hadoop distributions. A Distributed Anycast Gateway also offers seamless host mobility in the VXLAN overlay network. The gateway IP and virtual MAC address are identically provisioned on all VTEPs within a VNI, therefore, when an end host

moves from one VTEP to another VTEP, it doesn't need to send another ARP request to re-learn the gateway MAC address.

Figure: Distributed Anycast Gateway



Multicast in the Overlay

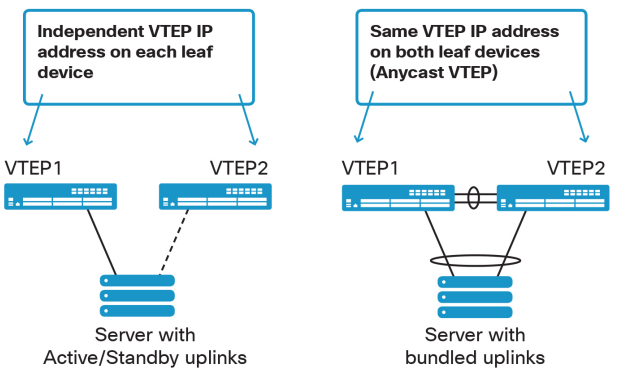
At the time of writing of this publication, there is no standard covering the implementation of IP multicast in the VXLAN overlay. This implies that if Layer 3 multicast services are required in the VXLAN overlay networks, the only possibility is connecting external multicast routers to the Fabric. Configuration of the external multicast routers is outside the scope of this publication.

Host Connectivity

When connecting endpoints (bare-metal servers, hypervisor hosts, network service nodes, etc.) to the network in a redundant fashion, two options are available:

- Using an Active/Standby attachment mode, where the endpoint leverages one or more active links to one leaf switch and one or more standby links to a second leaf switch. This ensures the endpoint can survive the failure of a single leaf switch and regain network connectivity simply by activating the standby links. This configuration does not require any specific functionality to be supported on the leaf, as normal Layer 2 learning and forwarding can be performed to deliver traffic to the locally connected endpoints.
- Using an Active/Active attachment mode, static or dynamic bundling of physical interfaces using Link Aggregation Control Protocol (LACP). This ensures that all available links are always active and used to send and receive traffic. This model requires that the leaf switches support a Multi-Chassis Link Aggregation (MC-LAG) functionality to appear as a single logical entity to the locally connected endpoints. Cisco Nexus switches offer Virtual Port-Channel (vPC) to achieve this.

Figure: VTEPs and Server Attachment Models



In a VXLAN Fabric, there are some additional aspects to consider.

- When the endpoint connects in Active/Standby mode, each leaf switch is configured with an independent VTEP IP address. Traffic originated by the server will always be VXLAN encapsulated and decapsulated by the leaf switch connected to the active port. Remote VTEPs will always point to VTEP 1 or VTEP 2 (depending on which link is active) when remote endpoints need to send traffic to devices locally connected to those leaf switches.
- When the endpoint connects in vPC mode (Active/Active), the two VXLAN leaf switches are deployed as part of the same vPC domain and a common Anycast VTEP is defined. As a consequence, no matter which physical uplink is used by the local endpoint to send the traffic into the network, remote VTEPs always associate the endpoint information to the source Anycast VTEP address. This is critical for the consistent Layer 2 MAC learning of the endpoint within the Fabric, in order to avoid continuous flapping of information in the MAC tables of the remote VTEPs.

In the Cisco NX-OS implementation, the Anycast VTEP address is defined as a common secondary IP address associated to the VTEP loopback interface of both VXLAN leaf switches part of the same vPC domain.

```
interface loopback0
description VTEP
ip address 10.254.254.102/32
ip address 10.254.254.1/32 secondary
```

It is worth noting that once a pair of VXLAN switches is configured as part of a vPC domain, the Anycast VTEP is always used as next-hop for all the EVPN advertisements relative to directly connected endpoints. This is valid also for local endpoints connected in Active/Standby fashion. The consequence is that roughly half of the flows destined to those devices may be delivered from the spines to the VTEP device connected to the standby endpoints (the spines have two equal cost paths to reach the Anycast VTEP IP address); the traffic would hence have to take an extra hop across the peer-link in order to be delivered to the active interface of the endpoint.

This suboptimal behavior can be avoided by grouping endpoints based on the types of connectivity (Active/Standby vs LACP) and connecting them to separate sets of leaf switches.

External Connectivity for VXLAN Fabric

Introduction

In real world data center deployments, the Fabric is never an isolated environment and connectivity with external networks is always required. The required connectivity typically depends on which type of external networks the VXLAN Fabric is connected to. For example, when connecting to the campus, WAN, or the Internet, Layer 3 routing is normally used. When extending Layer 2 outside of the VXLAN Fabric, additional connectivity considerations are required.

In addition to external connectivity, the VXLAN Fabric will typically be deployed into an existing data center environment, so interoperability with the existing network and the ability to migrate workloads to the new Fabric will be very relevant.

This chapter provides detail on both Layer 2 and Layer 3 external connectivity to the VXLAN Fabric and how to use those concepts to deploy a VXLAN Fabric into an existing data center.

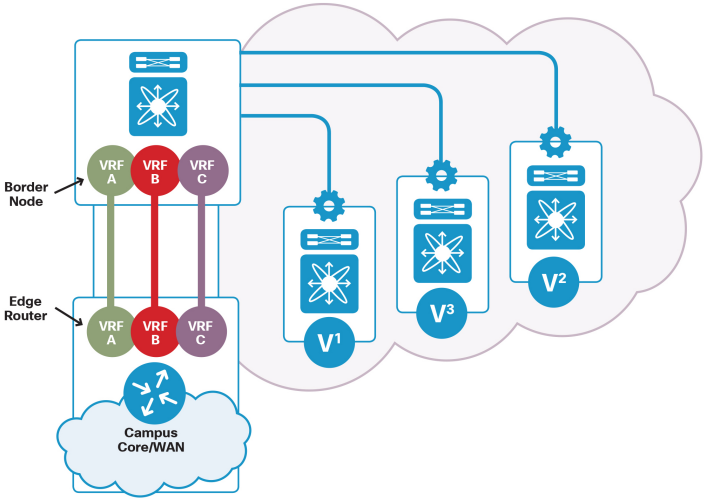
Layer 3 Connectivity

Multi-tenancy is one of the primary use cases for deployment of a VXLAN BGP EVPN Fabric. Different VRFs could be defined and segmented as different organizations, business units, mergers and acquisitions, user-groups, applications, or simply security segmentation and policy enforcement.

In the context of VXLAN BGP EVPN, each instance (i.e. VRF/VLAN) is logically isolated, but physically integrated into the overall Fabric as a shared infrastructure. When extending Layer 3 connectivity outside the VXLAN Fabric, two different scenarios are usually considered:

1 Extend the logical isolation between VRFs into the externally routed domain. This scenario is typically deployed when connecting the VXLAN Fabric to the campus network or to the WAN, as shown in the figure below.

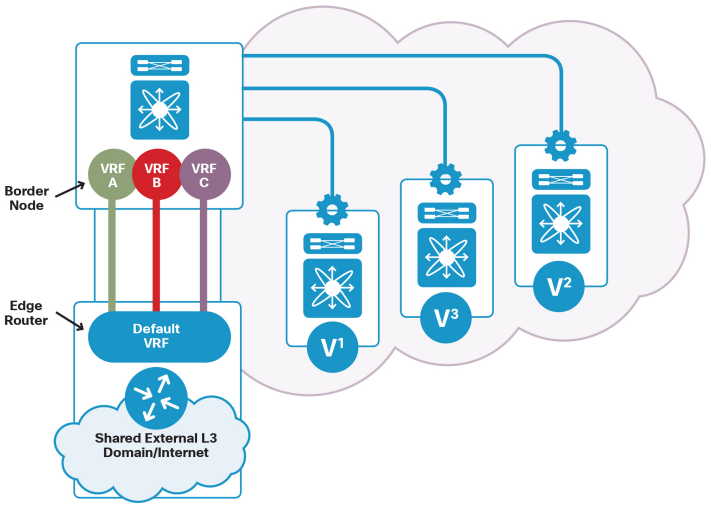
Figure: Extending Layer 3 Multi-Tenancy to the External Layer 3 Domain



The border node represents the edge of the VXLAN Fabric and normally terminates the VXLAN data plane encapsulation to provide Layer 3 hand-off functionality toward the edge router. The border node role could be implemented on a leaf or spine switch. The edge router takes care of extending multi-tenancy connectivity across the external network, leveraging one of the deployment options discussed in the sections below. It is worth noting this model allows full support for overlapping IP address space across different tenants, providing end-to-end logical isolation.

2 Provide shared access to a common external service. This scenario allows different tenants to have common access to shared resources such as the Internet.

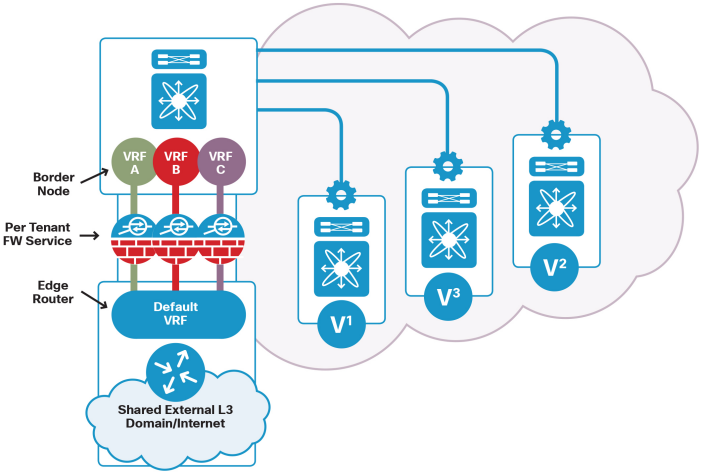
Figure: Accessing External Layer 3 Shared Resources



The simple use case shown above does not allow overlapping IP address space across different tenants, as this merges all the routing information into the “Default VRF” routing table. As an extension to the previous example, access to shared resources may be provided by front-ending each tenant with a security device. This provides an enforcement point for security policy when a tenant needs to access external resources or to communicate with other tenants as shown in the figure below.

In this case, it is common to leverage NAT (Network Address Translation) functionality offered by a firewall for tenants that must support overlapping address space.

Figure: Secure Access to External Layer 3 Shared Resources



The above diagrams illustrate how Layer 3 connectivity external to the VXLAN Fabric is provided by a border node device. The first design decision to make is the placement of the border node. Two main options are usually considered:

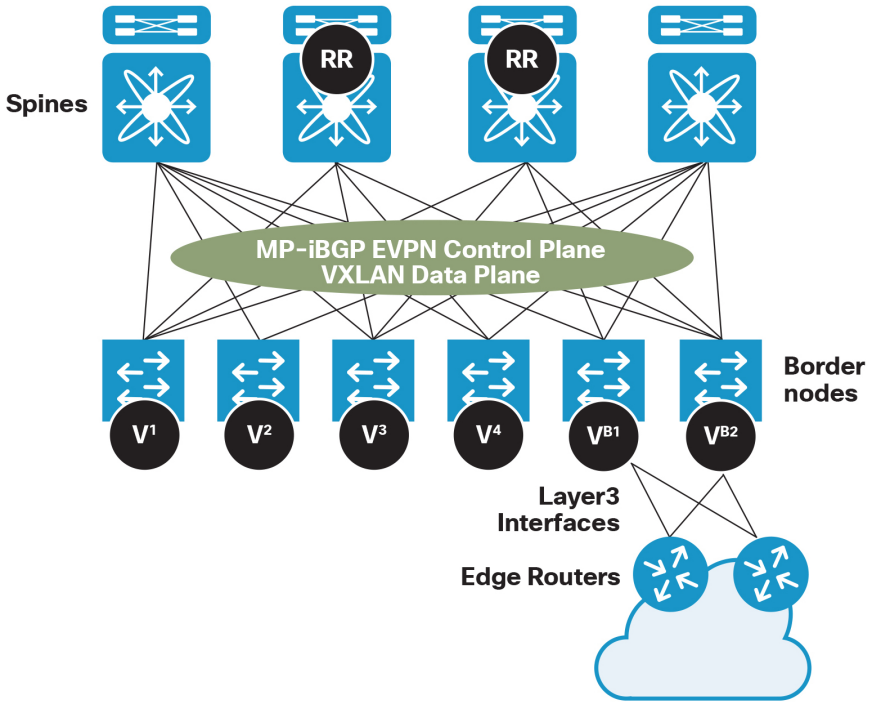
1 Border node on a leaf device is termed border leaf. This is a natural choice as the leaf nodes are deployed as VTEP devices capable of supporting the required control plane and data plane functionalities. Deploying the VTEP capabilities only on the leaf nodes keeps the configuration on the spine switches much simpler. The spine provides the Fabric backplane functionality, routing VXLAN encapsulated traffic between the leaf nodes. The border leaf only services north-south communication.

2 Border node on a spine device is termed border spine. This deployment option provides the advantage of optimizing the north-south communication with external resources. At the same time, it introduces the requirement to deploy a spine device that is capable of supporting VXLAN control and data plane functionality (VTEP). The border spine will most likely also serve as BGP Route Reflector (RR) and Multi-

cast Rendezvous Point (RP). The border spine services north-south as well as east-west communication.

A good network design always provides resiliency and redundancy for key network elements. The border node performs a key function, interconnecting the VXLAN Fabric to the external network domain, so it is critical to ensure resiliency. It is recommended to design the Fabric with redundant border nodes and edge routers, each leveraging redundant physical connections, as shown below.

Figure: Redundant Border Nodes and Connections to the Edge Routers



Regarding Layer 3 hand-off functionality, it is a fair assumption that the links between the border nodes and the edge routers are routed interfaces. Depending on how Layer 3 communication is extended outside the VXLAN Fabric, those Layer 3 interfaces could

be dedicated for each tenant or shared across multiple tenants. The following sections provide an overview of the different deployment options. All the scenarios depict a border leaf deployment, but the same considerations can be applied in the border spine case.

VRF-Lite Hand-Off

The use of VRF enables the ability to have multiple routing tables that are completely independent and isolated. VRF-Lite represents a common and well-known mechanism to extend the tenant Layer 3 VRF information beyond the VXLAN Fabric.

The VRF-Lite approach dictates using a two-box solution where the border node and the edge router are physically independent devices. With VRF-Lite, connectivity for different tenants from the VXLAN Fabric is extended externally on a hop-by-hop basis. The border leaf participates in the VXLAN Fabric and has the full VTEP configuration to perform the VXLAN encapsulation and decapsulation along with routing toward the edge routing device.

For this to happen, the following two requirements must be met:

- At the control plane level, the border node is responsible for exchanging per-tenant routing information between the VXLAN Fabric and the external network. The border node runs IPv4 or IPv6 unicast routing for each of the tenant VRFs with the external edge routing device to learn the external routes and to advertise the Fabric subnet/host routes to the external network. The border node also redistributes and advertises the external routes through MP-BGP EVPN to the internal nodes on the Fabric.
- The routing protocol used to communicate with the edge router can be BGP or an IGP routing protocol of your choice. When using BGP to peer with external routers, MP-BGP EVPN automatically imports the BGP routes learned from the VRF-lite IPv4 or IPv6 unicast address family into the L2VPN EVPN address family. This represents a common option adopted in many real world deployments. With other routing protocols, redistribution of routes is required to ensure routes are exchanged between the VXLAN Fabric and the external router.

When the border node learns the external routes from the edge router, it advertises the prefixes inside the VXLAN Fabric domain as EVPN Type-5 routes. This information is distributed to the other VTEP nodes. At the same time, the border node is configured to send EVPN routes learned from the L2VPN EVPN address family to the IPv4 or IPv6 unicast address family and advertise them to the external edge router.

The sample configuration below shows the example where eBGP with a sub-interface is used as routing protocol between the border node and the edge router.

```
vrf context Tenant-1
  vni 50000
  rd auto
  address-family ipv4 unicast
      route-target import auto evpn
      route-target export auto evpn
      route-target import auto
      route-target export auto

interface Ethernet1/10.100
  encapsulation dot1q 100
  vrf member Tenant-1
  ip address 192.168.5.254/30

router bgp 65500
  router-id 10.254.254.200
  neighbor 10.254.254.3
    remote-as 65500
    update-source loopback1
    address-family l2vpn evpn
      send-community both
  neighbor 192.168.1.1
    remote-as 65535
    address-family ipv4 unicast
    prefix-list filter-host-routes out
```

```
  vrf Tenant-1
     address-family ipv4 unicast
       advertise l2vpn evpn

ip prefix-list filter-host-routes seq 10 deny 0.0.0.0/0 eq 32
ip prefix-list filter-host-routes seq 20 permit 0.0.0.0/0 le 32
```

In this example, the “advertise l2vpn evpn” command under the VRF IPv4 address family ensures that:

- All the EVPN Fabric internal IP prefixes are advertised from EVPN into the VRF
- All the external IP prefixes learned from the edge router are advertised from the VRF into EVPN
- By default, all the Fabric internal prefixes, including host routes for the connected endpoints, are advertised toward the edge router. If this is not the desirable behavior, it is possible to apply route policy to eliminate host routes.

Similar configuration with the exception of the EVPN address family specific commands must then be applied on the edge router to ensure the BGP session can be established with the border node.

At the data plane level, traffic for different tenants must be carried between the border node and the edge router. This can be achieved by dedicating an interface (logical or physical) to each VRF. The available options are:

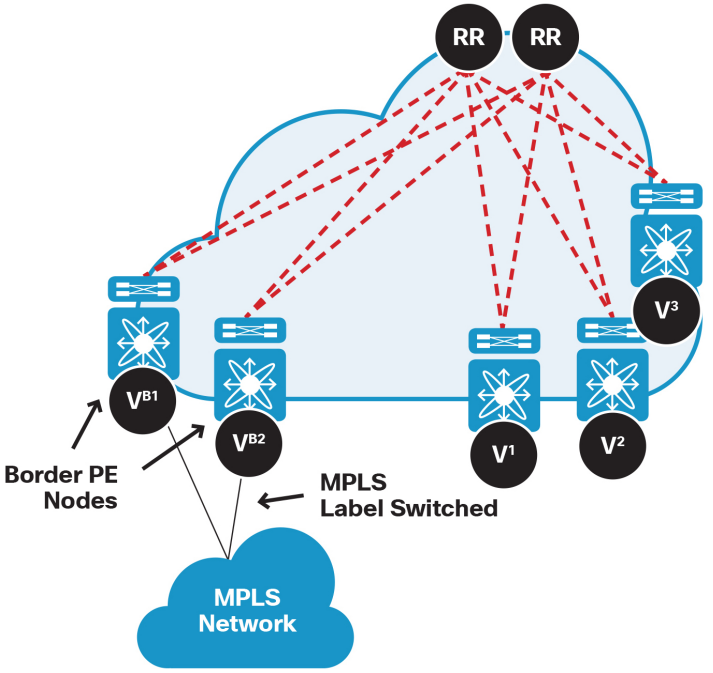
- Physical Routed Ports: this implies using a dedicated physical interface for each tenant
- Sub-Interfaces: one logical sub-interface can be carved for each tenant to carry traffic on the same physical connection.

As shown above, it is important to note, that for each VRF, manual configuration is required along the entire Layer 3 path. Since VRF-lite needs to be configured on a hop-by-hop basis, scalability becomes a concern for large numbers of tenants/VRFs; this is the advantage of an MPLS hand-off.

MPLS Hand-Off

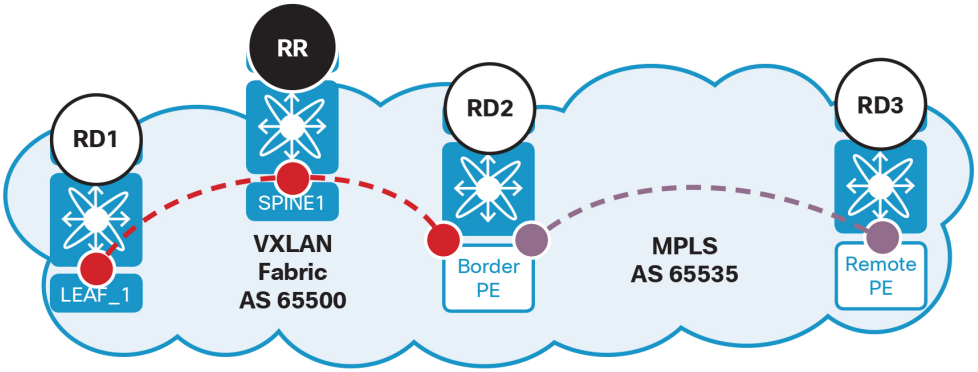
In many two-device deployments, the edge router can act as an MPLS-provider edge node. Alternatively, a single device solution can be used to terminate MPLS and VXLAN routing on the same device. This solution merges the border node and the MPLS Provider Edge (PE) router functionalities into a single physical device, usually referred to as the Border PE node. This scenario is depicted below.

Figure: Single Device Solution with Border PE Nodes



This section summarizes the steps for configuring the Border PE device deployed on a Cisco NX-OS based platform using manual configuration, with reference to the simple network topology shown below.

Figure: Single Device Configuration Example



The sample configuration below shows a Border PE example configuration.

```
    vrf context Tenant-1
      vni 50000
      rd auto
      address-family ipv4 unicast
          route-target import auto evpn
          route-target export auto evpn
          route-target import auto
          route-target export auto
          route-target import 65535:1
          route-target export 65535:1

    Note: The additional route-target have to match the one used in MPLS L3VPN
for each VRF.

    interface Ethernet1/10
      ip address 192.168.5.254/30
      ip router ospf MPLS-CORE
      mpls ip
```

```
router bgp 65500
  router-id 10.254.254.200
  neighbor 10.254.254.3
    remote-as 65500
    update-source loopback1
    address-family l2vpn evpn
      import vpn unicast reoriginate
        send-community both
  neighbor 192.168.1.1
    remote-as 65535
    address-family vpnv4 unicast
      import l2vpn evpn reoriginate
  vrf Tenant-1
    address-family ipv4 unicast
      advertise l2vpn evpn
```

The Border PE re-originates IP prefixes from the VXLAN Fabric EVPN address family to the MPLS VPNv4 address family and vice versa. The required commands to achieve this are “import vpn unicast reoriginate” or “import l2vpn evpn reoriginate” respectively in the opposite address-family. It is required to use an eBGP peering between the Border PE and the MPLS PEs. For the import and export to MPLS L3VPN, the appropriate route-targets have to be chosen for each VRF.

LISP Hand-Off

In Active/Active data center deployments, workload mobility allows applications to move between geographically dispersed locations. This brings the challenge of ingress route optimization when the workloads change location. Locator/Identifier Separation Protocol (LISP) solves this challenge by routing the client traffic to the correct location where the resources are located. The routing information for LISP does not add any additional prefixes to the underlay routing domain.

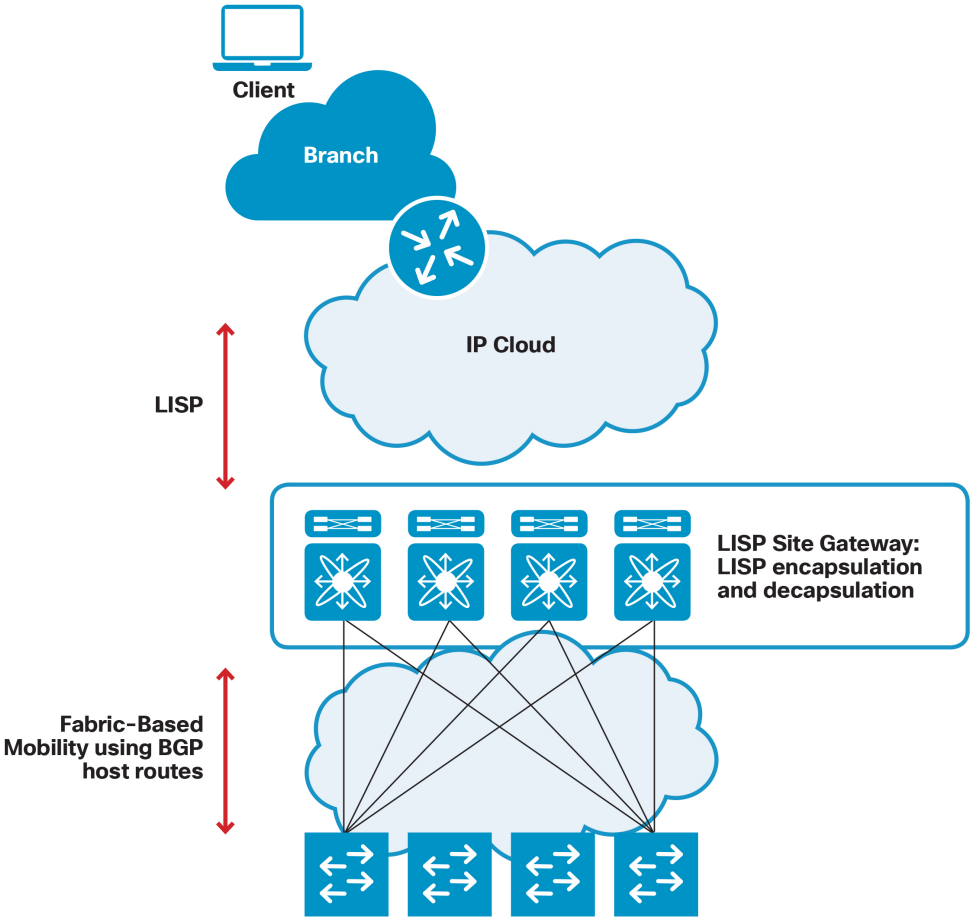
LISP, as defined in RFC 6830, is a routing architecture that enables a new paradigm for IP addressing. IP addresses are scoped in two distinct namespaces: Endpoint Identifiers (EIDs), which are assigned to end-hosts, and Routing Locators (RLOCs), which are assigned to networking devices. The LISP protocol provides all the messaging necessary to maintain and access a mapping database in which EIDs are correlated to RLOCs. LISP uses a map-and-encapsulate forwarding model in which traffic destined for an EID is encapsulated and sent to the RLOC of the device through which it is connected, based on the results of a lookup in a mapping database. Traffic is sent to a device's RLOC rather than directly to the destination EID. This approach relieves the core network of the responsibility of handling EID information. Using this approach, the LISP architecture augments the current routed infrastructure to facilitate new functionality with minimal disruption to the existing network infrastructure.

LISP is a directory of addresses and their locations, not a traditional routing protocol. LISP uses a demand-based model where edge-devices request location information as required. This demand model is in contrast with the push model used by routing protocols and results in a reduced load on the device's hardware tables. LISP has other advantages noted below:

- Mobility: EID portability
- Scalability: On-demand routing
- Security: Tenant ID-based segmentation
- DCI: Ingress route optimization

LISP mappings can be classified to give VPN and tenant semantics to each prefix handled by LISP. This classification is encoded in the LISP control plane as stipulated in the standard definition of the protocol. The LISP data plane also supports segmentation of traffic into multiple VPNs. LISP binds VRFs to instance IDs, and then these IDs are included in the LISP header to provide data plane (traffic flow) separation for single or multi-hop forwarding. The LISP multi-tenancy solution promises to exceed the scalability of current segmentation solutions significantly because it uses on-demand routing and does not require maintenance of traditional routing adjacencies.

Figure: Border Spine and LISP Hand-Off



Border Spine and LISP Hand-Off

This section focuses on a design where the spine device acts as a border node and supports LISP handoff. Since all of the spines are connected to the WAN edge routers, this allows us to have ECMP from the spine devices to given external sites. This allows hosts connected to the VXLAN Fabric to communicate to the external sites. Although we focus on the scenario where the spines act as border nodes, a similar design can be implemented on border leaf nodes.

In this scenario, the spine device acts as a LISP xTR. A LISP xTR refers to a device that can act as both a LISP Ingress Tunnel Router (ITR) and a LISP Egress Tunnel Router (ETR). With LISP, regular IPv4/IPv6 host routes originating from the data center are not advertised which helps optimize the routing table.

LISP has a mapping database system that keeps track of routes learned from all spine devices in the EVPN Fabric. LISP also tracks addresses on remote sites and adds them to the mapping database. Routes learned from leaf VTEPs are added into the Routing Information Base (RIB) at the xTR. LISP selects these routes from the RIB and adds them dynamically to the mapping database as Locator-Identity mappings.

The spine advertises a default route to attract northbound traffic from the leaf VTEPs. When a leaf VTEP receives a packet and it does not have a specific route, it sends the packet to the spine using the overlay. The spine decapsulates the packets, performs a lookup in the LISP mapping database, does a LISP encapsulation and forwards the packet northbound across the WAN.

The spine devices continue to have L3VNIs configured, as they act as VTEPs for northbound traffic coming from attached leaf devices. They would also act as tunnel endpoints for southbound traffic coming from the remote LISP xTR destined to the hosts connected to the leaf. The spine devices would not need to be configured with L2VNIs and has the advantage of allowing Layer 3 multi-pathing across the VXLAN Fabric.

North-South Traffic with VXLAN Host in the POD

In this scenario, packet forwarding involves two encapsulations:

1 LISP encapsulation between the external sites and the border spine

2 VXLAN encapsulation between the border spine and the leaf

The following scenario discusses host detection and packet forwarding:

1 The VXLAN Fabric can be connected across an IP cloud to connect to external sites for north-south traffic using LISP. Making use of the Border PE provider edge solution to connect the data centers and external sites using LISP.

2 In the VXLAN Fabric, the host routes and MAC address information are distributed in the MP-BGP EVPN control plane from the leaf nodes, which means that the Fabric itself performs the host detection. The LISP site gateways use these host routes for triggering the LISP encapsulation and de-encapsulation.

3 When the LISP site gateway (Border PE, also running MP-BGP EVPN in the Fabric) detects this host based on the route received in BGP, it sends a map-register message to the map system database to register the new IP address in its own data center

4 When remote sites want to talk to the data center hosts, they send an inquiry to the mapping system requesting the location of the host. The mapping system replies with the location of the LISP site gateway where the destination EID is located.

5 Communication is then established between the remote client and the data center host leveraging the LISP and VXLAN technologies as described earlier

Layer 3 Connectivity Summary

External Layer 3 connectivity from a VXLAN Fabric can be achieved using three different technologies.

- VRF-lite provides an IP hand-off using sub-interfaces with IEEE 802.1Q tags to separate the VRFs
- MPLS uses VPN labels to separate traffic on a per-VRF basis
- LISP uses an IP-in-IP encapsulation and instance-ID to segregate the VRFs

Layer 2 Connectivity

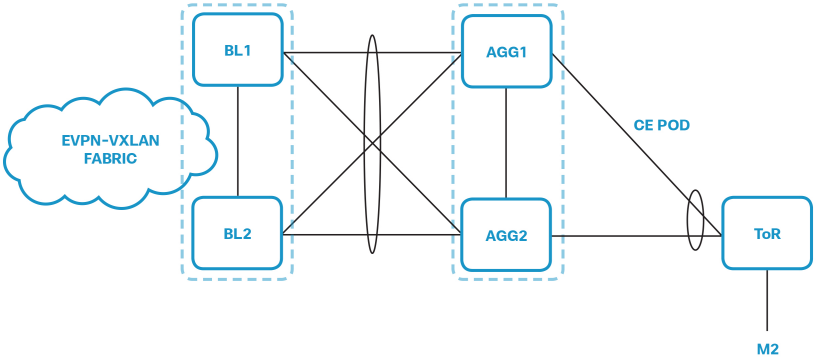
There are two major use-cases for Layer 2 hand-off and connectivity. The first is for migration scenarios, where the VXLAN Fabric needs to be connected to an existing non-VXLAN network infrastructure. The second is the extension of Layer 2 broadcast domains between separate VXLAN Fabrics, referred to as multi-site.

Layer 2 vPC Hand-Off

In this scenario, the VXLAN Fabric is connected to an external Layer 2 network via Ethernet 802.1Q VLAN trunks. This external Layer 2 network will be referred to as a Classical Ethernet (CE) POD. Layer 2 connectivity can be extended between the two environments, taking the form of an L2VNI on the VXLAN Fabric and a traditional VLAN on the CE POD.

A vPC border node pair on the VXLAN Fabric can be used as redundant Layer 2 gateway for the hand-off. In this case, the two environments can be connected via a vPC without introducing loops to the extended Layer 2 networks.

Figure: Traffic Flow Between a VXLAN Fabric and a CE POD



In the illustration above:

- BL1 and BL2 provide Layer 2 border leaf functionality for the VXLAN Fabric
- The border leafs form a back-back vPC to redundantly connect with the CE pod aggregation switches
- The assumption is that the aggregation layer switches support vPC

vPC Unicast Communication

With vPC, the pair of border leaf switches shares a single virtual VTEP IP address and MAC address. This allows both devices to handle the forwarding and receipt of unicast traffic.

When the VTEPs learn MAC reachability information for devices in the CE POD, they inject this information into the EVPN Fabric control plane. They associate the virtual VTEP IP address to the endpoint MAC addresses connected to the CE pod. This ensures that the other VTEPs within the VXLAN Fabric receive this information and program it in their Layer 2 forwarding tables. Any leaf in the VXLAN Fabric can reach resources connected to the CE pod by encapsulating traffic in VXLAN packets destined to the single virtual VTEP next-hop address. This implies traffic in the underlay Fabric network can be load-balanced across Equal Cost Multipath (ECMP) paths. In the event of a failure of a border leaf node, minimal impact is observed given the redundancy built into the Fabric. MAC addresses learned from the CE POD in the data plane are synched across the vPC peer link so both border leaf switches are capable of forwarding unicast Layer 2 traffic directed towards them.

The border leaf switches are aware of the IP and MAC addresses of all the endpoints connected to the VTEPs in the VXLAN Fabric, so traffic received from the CE POD can be VXLAN encapsulated and forwarded inside the Fabric towards the destination VTEP.

vPC Multicast Communication

For BUM traffic handling, the border leaf simply floods the packet to the vPC which supports an Active-Standby model for multi-destination packet forwarding. Only one of the vPC peers is selected as the designated forwarder on a per group basis and is responsible for forwarding the BUM traffic to avoid creating multiple copies of the same packets.

When a device in the VXLAN Fabric sends a multicast packet:

- Both vPC border nodes receive the multicast traffic encapsulated in VXLAN
- The designated forwarder will decapsulate the packet and forward it to the CE POD
- The non-designated forwarder switch drops the ingress packet

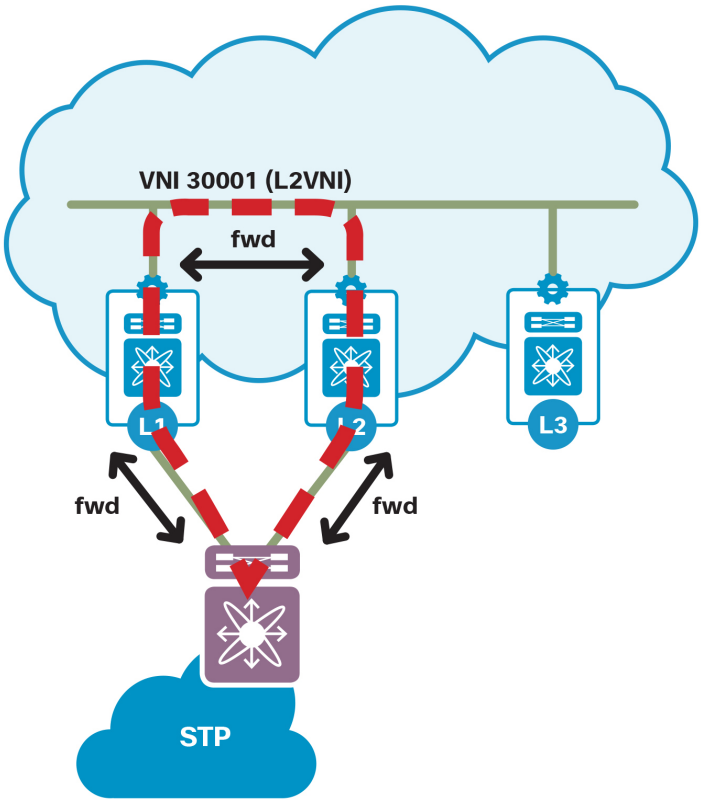
When a device in the CE POD sends a multicast packet:

- When the packet reaches the border nodes, one of the vPC peers is designated as the forwarder for that VLAN. That vPC peer will take responsibility for encapsulating the multicast packet into the VXLAN EVPN Fabric.

Loop Prevention

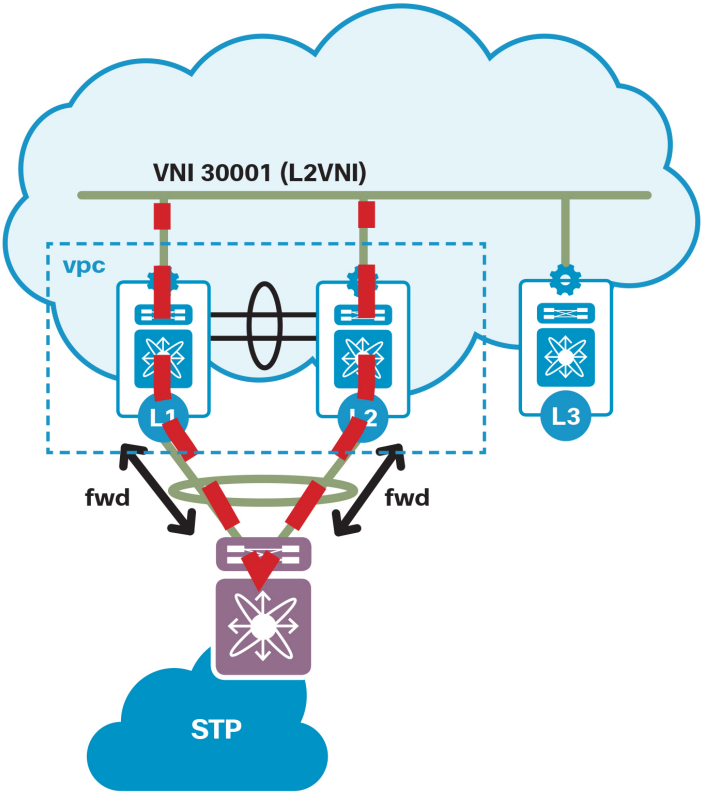
As Layer 2 is extended outside the VXLAN Fabric, it is important to remember that the border node participates in the VXLAN Fabric, both from a control and data plane perspective. VXLAN does not currently provide any integration with Spanning Tree (STP), meaning VXLAN does not forward BPDUs across the Fabric. Therefore, establishing redundant Layer 2 connections between the VXLAN Fabric and the external network may result in the creation of a loop as highlighted in the figure below.

Figure: Creation of a Layer 2 Loop



In order to have a multi-homed loop-free topology, Cisco recommends using vPC for the southbound connectivity of Edge Devices as shown below.

Figure: Layer 2 Loop-Free Topology with vPC



Since the border nodes are already participating in the Layer 3 hand-off, it is a natural choice to leverage them to extend Layer 2 connectivity outside the Fabric. The recommendation is to enable vPC on the border node and provision parallel interfaces between the border node for the Layer 2 VLANs that need to be extended. It is important to note, that the border node does not have to be used for Layer 2 extension, and it is possible to leverage another pair of leaf switches for this function. It is not recommended to leverage the border spine for the Layer 2 connectivity between locations as the spine devices should be independent nodes.

Integration and Migration

Greenfield scenarios do not require much focus on integration with legacy technologies. This section focuses on brownfield scenarios, using the information provided in previous sections of this chapter.

Brownfield data centers typically integrate new network technologies, VXLAN Fabrics are no exceptions, using one of the following methodologies:

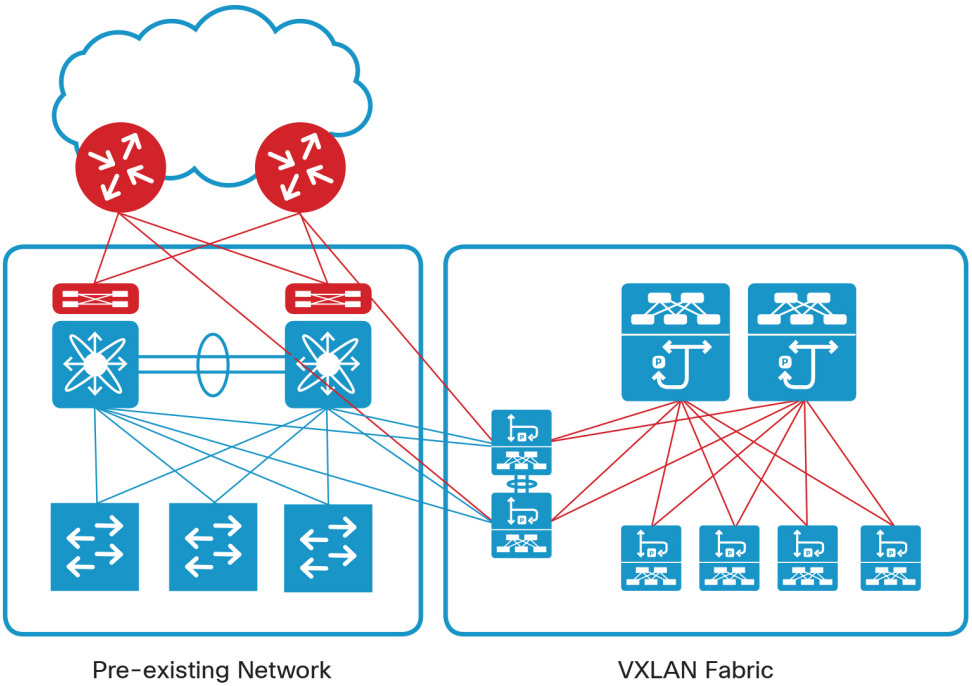
- **Layer 2 POD expansion**: the new VXLAN Fabric will be connected with the existing network using Layer 2.
- **Layer 3 POD addition:** the new VXLAN Fabric will be connected with the existing network using Layer 3

Since the implementation of the new VXLAN Fabric as an additional Layer 3, POD in the DC does not typically involve migrating workloads from the existing network to the new Fabric. The next section will focus on the first use case, deploying a VXLAN Fabric as a Layer 2 extension of an existing network.

Expansion of an Existing POD with a VXLAN Fabric

This section provides an overview of migrating to a VXLAN Fabric from an existing network that could have been built leveraging vPC, Fabricpath or traditional STP technologies. The VXLAN Fabric is built as a new POD deployment and the scope does not cover conversion of the existing network devices into VXLAN nodes. The goal is to provide network integration and a path for the migration of endpoints and services to the new VXLAN Fabric with minimal service disruption. Once the integration is complete, the Layer 3 and L4-L7 services could optionally be migrated to the VXLAN Fabric.

Figure: Layer 2 and Layer 3 Interconnect to Assist Integration and Migration



Layer 2 Interconnect

Using the techniques described earlier in this chapter, the new VXLAN Fabric can be interconnected with the existing network, leveraging vPC and loop prevention techniques such as BPDU Guard, Root Guard and storm control to deliver a redundant Layer 2 path between the two environments.

At the VXLAN Fabric border nodes, the same VLAN IDs need to be used in order to map Layer 2 segments from the existing network to L2VNIs to establish Layer 2 connectivity. Virtual machines and endpoints can now be seamlessly moved from the existing network to the new VXLAN Fabric with minimal impact. The endpoints in the VXLAN Fabric will still have Layer 2 connectivity with the endpoints that have not yet been mi-

grated. The default gateway for all Layer 2 segments still resides in the original network at this point.

Moving Endpoints to the VXLAN Fabric

Once Layer 2 connectivity between the legacy network and the new VXLAN Fabric is operational, workloads can be migrated. Migrating physical servers will typically require recabling and a service disruption for the server being migrated. On the other hand, virtual machines can be migrated over live migration without any noticeable network impact.

Moving L4-L7 Network Services to the VXLAN Fabric

L4-L7 services appliances are essentially endpoints, so they may be migrated in the same way as the server workloads. It may be possible to migrate virtual L4-L7 appliances without disruption depending on their capabilities and configuration. In the case of physical appliances, high availability features such as clustering can help to minimize disruption and allow for migration to the new Fabric.

You can find additional details about how L4-L7 services can be connected to a VXLAN Fabric in the chapter Layer 4-Layer 7 Services.

Moving the Default Gateway

Once all endpoints have been migrated to the VXLAN Fabric, it would be suboptimal to still have the default gateway in the existing network. The next step in the migration is to move the default gateway to the new Fabric. The new Fabric must have an existing Layer 3 uplink into the network core to preserve connectivity to the existing routed network.

This migration is a four-step process:

1. Disable the default gateway in the existing network
2. Configure the gateway IP address as a Distributed Anycast Gateway in the new VXLAN Fabric. By using the MAC address of the original default gateway, the endpoints do not need to re-ARP for the new default gateway

3 Ensure that the subnet is advertised upstream to the Layer 3 network core

4 Remove the default gateway and routing configuration from the existing network

Note that although step 3 can be configured in advance, step 2 needs to be done sequentially after step 1. There will be a short outage from step 1 until the anycast gateway is fully functional in the VXLAN Fabric and all endpoints have learned the MAC address for the new gateway.

Relearning the gateway's MAC address is not required if the anycast gateway in the VXLAN Fabric can overtake the same MAC address that the old default gateway had. One restriction is that, as it has been described in previous chapters, there is a single MAC address for the whole Fabric for all Distributed Anycast Gateway, so if the default gateways in the legacy network had multiple MAC addresses (for example if multiple HSRP or VRRP groups were used), a migration where the MAC address of the default gateway stays the same will not be possible.

In case there are still endpoints in the original network, they will have to use the anycast gateway in the VXLAN Fabric to communicate with other network segments. The ARP suppression mechanism may cause traffic blackholing so it should not be enabled until all endpoints have been migrated. That is the reason why, in order to reduce the complexity of the migration, the recommendation is to completely migrate all endpoints from the legacy network to the new VXLAN Fabric. Once that is done, the whole VXLAN configuration can be removed from the legacy network.

At this point, communication between subnets where the default gateway is still in the legacy network and subnets whose default gateway has already been migrated to the VXLAN Fabric is suboptimal and can potentially follow asymmetric paths. Therefore, it is recommended to complete the migration of all endpoints and segments from the old network to the new VXLAN Fabric in a period of time as short as possible.

Layer4-Layer7 Services

Introduction

This chapter provides an overview of Layer4-Layer7 services, deployment models, a focus on design and on deployment use-cases.

A VXLAN Fabric provides Layer 2 and Layer 3 connectivity; however, additional services are required in the data center. These services are provided by dedicated appliances (physical or virtual), and require connectivity to the fabric. These dedicated functions are referred to as Layer4-Layer7 services.

Traditional hierarchical network designs connect Layer4-Layer7 services at the aggregation layer. Within a VXLAN Fabric, Layer4-Layer7 appliances can be connected to any leaf switch or connected to a dedicated leaf pair referred to as a “service leaf".

There are different connectivity options for the physical and virtual appliances. The following section discusses different options for connectivity for the Layer4-Layer7 services devices.

Layer4-Layer7 Device Types

Depending on the requirements, multiple Layer4-Layer7 services may be implemented to provide a complete network and service function stack. These functions include the following:

- **Stateful Layer 4 firewalling**: Many organizations implement network security on dedicated firewalls where complex firewall policies are enforced. The firewall policies permit or deny communication between different organizational or application tiers. There are many other functions firewalls can perform such as Network Address Translation (NAT).
- **Application Firewalls**: Most attack vectors today focus on the application. The attacks leverage standard TCP ports to exploit application vulnerabilities. Examples include SQL Code Injection or Cross-Site Scripting. Application-level firewalls can help prevent these types of modern day attacks.

- **Intrusion Detection (IDS) / Intrusion Prevention (IPS)**: The solution detects attacks and prevents systems from being compromised. It also prevents a compromised system from originating suspicious network activity. Examples are network reconnaissance with ping sweeps and port scans.
- **WAN Optimization**: The goal of this service is to improve the user experience through techniques such as optimization of the TCP stack, compression, and content caching.
- **Application Delivery Controllers (ADC):** The ADC includes server load balancing, SSL offload and other application functionality. ADCs can be deployed by themselves or in tandem with other service nodes.

Some Layer4-Layer7 appliance vendors might integrate several of these above categories in a single product such as FW and IPS. In addition, another commonly used term is a service-chain, when multiple Layer4-Layer7 devices are implemented in sequence, such as WAN optimization, FW and ADC.

Deployment Models

In addition to the functionality of the Layer4-Layer7 services, an important factor to consider is how to deploy the service appliances. The following section describes different deployment models for Layer4-Layer7 services.

Virtual vs Physical

Layer4-Layer7 services come in different form factors including physical and virtual appliances. There are certain considerations required for virtual appliances, including the following:

- With virtual appliances, there is typically a virtual switch between the physical leaf and the VM hosting the services appliance
- Virtual services have different NIC redundancy models; these functions are provided by the hypervisor

The decision whether to use virtual or physical appliances requires additional considerations including that physical appliances are generally specialized hardware which offers better performance than generic x86 platforms, particularly with encryption services.

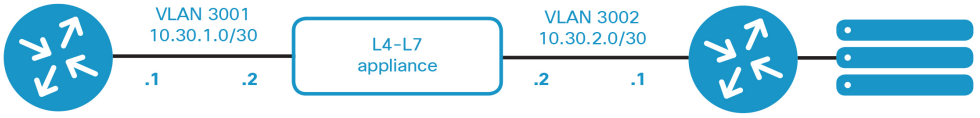
Transparent vs Routed

There are two deployment models with service appliances, transparent mode and routed mode. In transparent mode, the service appliance is deployed as a bump-in-the-wire and does not change any MAC information. With transparent mode, a fail safe mechanism needs to be implemented to prevent Layer 2 data plane loops.

Figure: Layer4-Layer7 Service in Transparent Mode

On the other hand, routed deployments are not prone to Layer 2 loops because they follow IP routing semantics. Layer4-Layer7 appliances inserted in routed mode can participate with dynamic routing protocols. The benefit of implementing a dynamic routing protocol is that it allows for Route Health Injection (RHI) that influences the ingress routing path to the services appliance.

Figure: Layer4-Layer7 Service in Routed Mode

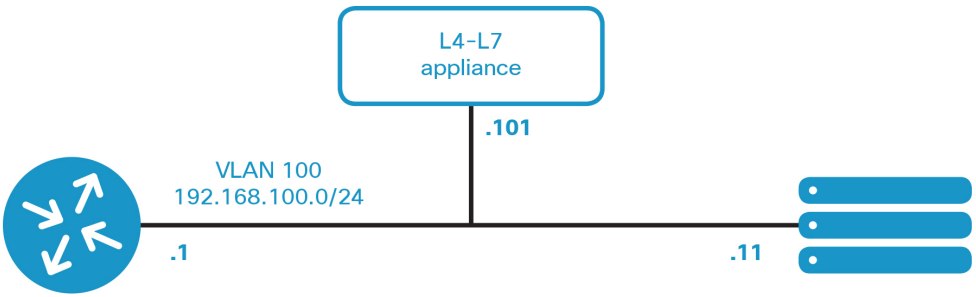


One-arm vs Two-arm designs

Firewalls have two or more interfaces, an internal interface, and an external interface. ADC can be connected in a two-arm or one-arm mode; one-arm mode implements a single logical or physical interface. The ADC typically implements Network Address Translation (NAT) to ensure that the return traffic is sent back to the original ADC appliance.

The following figure illustrates the one-arm design option.

Figure: Layer4-Layer7 service in one-arm mode

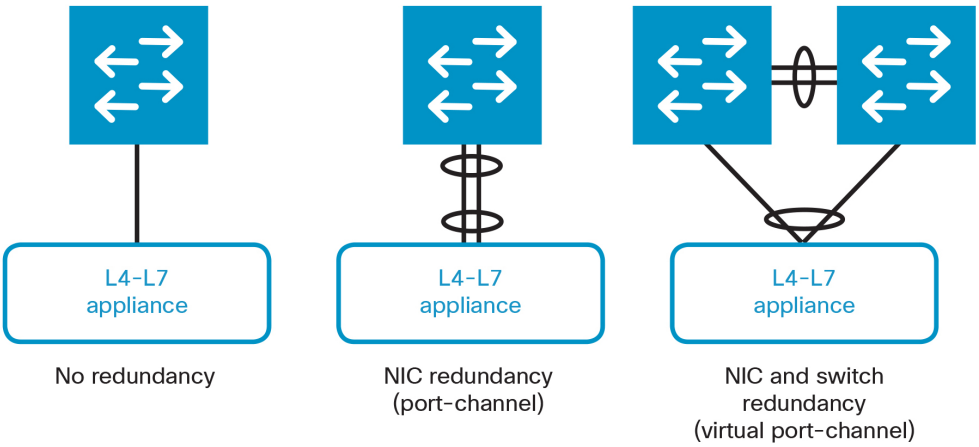


Physical Connectivity

Layer4-7 services have different connectivity and redundancy deployment models, as discussed below.

- **No redundancy**: one logical interface maps to one physical interface, resulting in a single network connection
- **Redundancy at the NIC level (port-channel)**: one logical interface maps to multiple physical interfaces. These two interfaces are configured as a single port-channel connected to a single leaf switch
- **Redundancy at the NIC and switch level (vPC)**: one logical interface maps to multiple physical interfaces. These two interfaces are configured as a single port-channel connected to two different leaf switches. The two different switches are implemented as a vPC pair.

Figure: Physical Connectivity Options



Redundancy Model

Different redundancy models will have an impact on how the network will behave in case of an Layer4-Layer7 appliance outage:

- **No redundancy**: This mode is sometimes used for non-critical environments, and is typically deployed in conjunction with virtual Layer4-Layer7 appliances that leverage High Availability features of the hypervisor.
- **Active/Standby**: Two Layer4-Layer7 appliances are deployed, and one of them handles all traffic. When the active device fails, the standby device will become active. The network converges away from the failed appliance while the previous standby node becomes active. With the active / standby model, traffic flows are deterministic and this simplifies the forwarding path through the network.
- **Clustering (Active/Active)**: There are two different models of clustering, where all services appliances are serving the workload. While one model uses the approach of a local port-channel per services appliance, the second model represents the services cluster as a single port-channel.

Integration into the VXLAN Fabric

In most cases the Layer4-Layer7 appliance is seen by the fabric as an endpoint. This does not require any additional control plane interaction with the fabric. However, in some cases, the Layer4-Layer7 vendor has implemented VXLAN encapsulation support to the service appliance. This provides the flexibility to leverage VXLAN for data plane integration. In this scenario, the service appliance would act as a VTEP.

When considering integrating Layer4-Layer7 service appliances into a VXLAN Fabric, the implementation detail needs to align between the two. For example, if the VXLAN Fabric is running with a BGP EVPN control plane, the service appliance needs to support this deployment model also. Within this book, use of a service appliance as a VTEP is not considered.

Use Cases

There are multiple options that are possible to deploy Layer 4-Layer 7 services in a VXLAN Fabric: physical single-arm, vPC-based, ADC deployment in active/standby mode, virtual active/active firewalls in routed mode, transparent virtual intrusion prevention systems, etc. The following sections focus on the most frequent use cases.

Firewall as Default Gateway

Using the firewall as the default gateway is one of the simplest use cases.

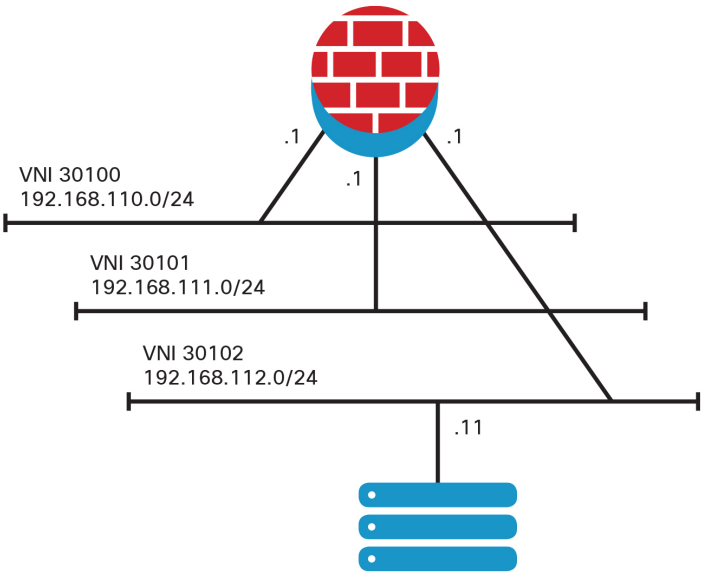
In this design, the VXLAN Fabric provides a Layer 2-only service. All communication that requires crossing the Layer 2 demarcation must be sent to the firewall to be routed.

For example:

```
vlan 1100
  name WEB
  vn-segment 30100
vlan 1101
  name APPLICATION
  vn-segment 30101
vlan 1102
  name DATABASE
  vn-segment 30102
```

The firewall will have a logical Layer 3 interface in each VNI that will serve as the default gateway for all endpoints. Routing between IP subnets, represented by a VNI, has to flow through the firewall. The firewall becomes the Layer 3 gateway for all VNIs for the VXLAN Fabric.

Figure: Firewall as a Default Gateway with a Layer 2 VXLAN Fabric



For example, an ASA firewall with four physical ports grouped in two logical port-channels:

```
int po10.1100
 vlan 1100
 nameif WEB
 security-level 100
 ip address 192.168.110.1 255.255.255.0

int po10.1101
 vlan 1101
 nameif APPLICATION
 security-level 100
 ip address 198.168.111.1 255.255.255.0
```

```
int po10.1102
 vlan 1102
 nameif DATABASE
 security-level 100
 ip address 198.168.112.1 255.255.255.0

int po20
 nameif OUTSIDE
 security-level 50
 ip address 192.168.100.255 255.255.255.0
```

The firewall becomes the single point for inter-subnet communication in the fabric, consequently, it is important to properly size the appliance for resilient, performance, and scale reasons. When a failure occurs in an active/standby deployment, the newly-active firewall will notify the network of the change, normally sending GARP (gratuitous ARP) or RARP (reverse ARP) packets. These will trigger the re-learning of the MAC addresses on the ports connected to the standby firewall.

Transparent Firewall Insertion

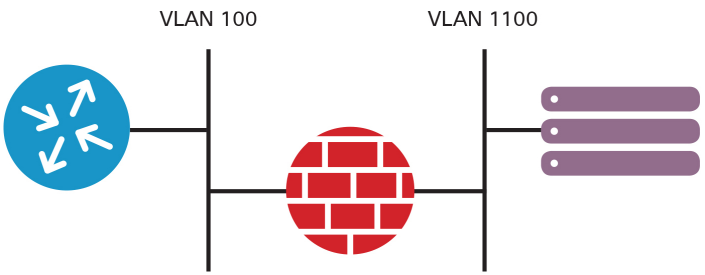
Another popular option for deploying firewalls is to transparently insert the firewall into the network, between the server's default gateway and the server itself. Some reasons to use transparent firewall insertion include:

- Ability to add firewall services without changing existing IP addressing of the servers
- Multicast streams can easily traverse the firewall
- Non-IP traffic can be forwarded via the firewall
- Protocols such as HSRP and VRPP can pass through the firewall
- Routing Protocols can establish adjacencies through the firewall

From a logical standpoint, the fabric is the default gateway for the servers. For example, the servers are deployed in the 192.168.100.0/24 subnet and the VXLAN Fabric anycast gateway is configured as the server's default gateway of 192.168.100.1.

The firewall needs to be inserted transparently into the datapath. Instead of the servers being deployed in the same VLAN/VNI as the default gateway, the servers will be configured in a different VLAN/VNI. For example, the default gateway resides in VLAN 100 (unprotected), while the servers are being placed in VLAN 1100 (protected). The firewall in transparent mode is stitching both VLAN/VNI together, meaning the firewall is in the datapath between VLAN 100 and VLAN 1100. Whenever a server requires reaching the default gateway, the traffic has to pass the firewall.

Figure: Transparent Firewall



The firewall enforces the security policies applied for data passing between the protected and unprotected VLANs and maintains the appropriate forwarding between them. This design can be used to deploy a micro-segmentation service inside of a subnet for servers that might have been compromised. As an example, instead of readdressing the servers you can dynamically move them behind a firewall and isolate the infected hosts from the rest of the fabric.

Example:

```
vlan 100
  name UnProtected-SVI
  vn-segment 30000

vlan 1100
  name Protected-VLAN
```

```
    vn-segment 31000

interface Vlan100
    no shutdown
    vrf member Tenant-1
    no ip redirects
    ip address 192.168.100.1/24 tag 21921
    fabric forwarding mode anycast-gateway
```

In this configuration, VLAN 100 (unprotected) is the outside interface and VLAN 1100 (protected) is the inside interface.

The firewall configuration to stitch VLAN 100 to VLAN 1100 would be as follows:

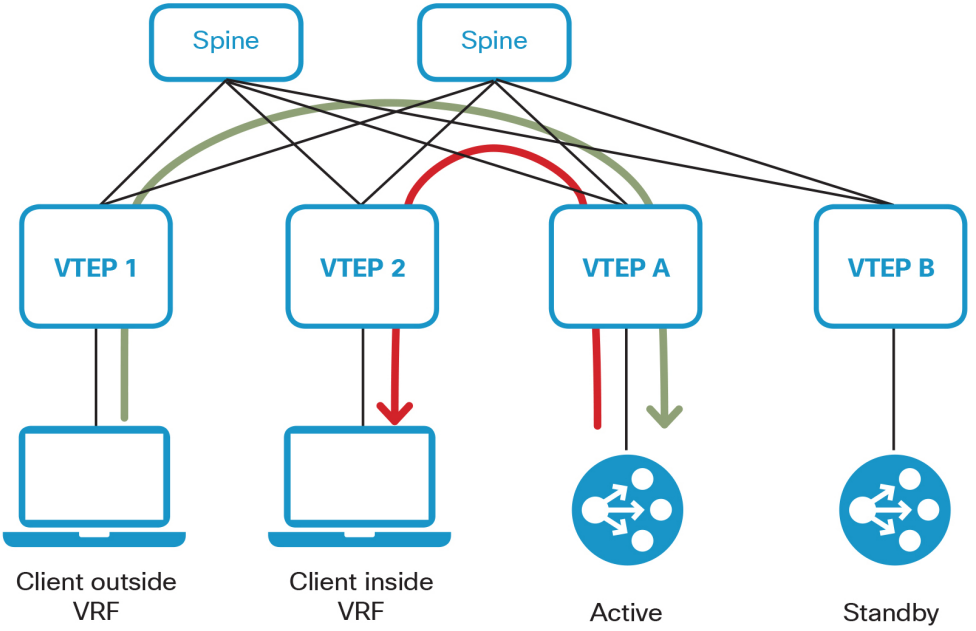
```
firewall transparent

int po10.100
 vlan 100
 nameif sviVLAN
 bridge-group 1
 security-level 0

int po10.1100
 vlan 1100
 nameif serverVLAN
 bridge-group 1
 security-level 100
```

Integrating Layer 3 Firewall - Multi-Tenancy

A common requirement is to provide security policy for inter-tenant traffic and for accessing shared-services in a dedicated VRF. As we have seen, the VXLAN Fabric provides multi-tenancy through MP-BGP and VRF technologies. Multi-tenant communication is routed throughout the VXLAN Fabric and tenant isolation is maintained.

A Layer 3 firewall involves separating different security zones using different subnets. The firewall routes traffic between subnets and applies the firewall rules.When integrating Layer 3 firewall in a VXLAN EVPN Fabric using Distributed Anycast Gateway, each of these zones must correspond to a VRF on the fabric. The traffic within a VRF will be routed by the fabric and traffic between the VRFs will be routed by the firewall.

Figure: Layer 3 Firewall Traffic Flow



The example below shows a configuration snippet from VTEP A running OSPF with the firewall.

SVIs are defined on VTEP for both INSIDE-VRF and OUTSIDE-VRF and the VTEP will peer with a firewall on each of these VRF to dynamically learn routing information to go from one VRF to the other.

```
    FIREWALL Configuration:
      int po10.3001
       vlan 3001
       nameif OUTSIDE
       security-level 50
       ip address 10.30.1.2 255.255.255.252

    int po10.3002&#;      vlan 3002&#;      nameif INSIDE&#;      security-level
100&#;      ip address 10.30.2.2 255.255.255.252&#;        router ospf 1
       network 10.30.1.0 255.255.255.0 area 0
       network 10.30.2.0 255.255.255.0 area 0

    VTEP A Configuration

    interface VLAN 3001
      description outside_vlan
      vrf member OUTSIDE-VRF
      ip address 10.30.1.1/30
      ip router ospf 1 area 0

    interface VLAN 3002
      description inside_vlan
      VRF member Tenant-1
      ip address 10.30.2.1/30
      ip router ospf 1 area 0

    router bgp 65500
      vrf OUTSIDE-VRF
      address-family ipv4 unicast
      advertise l2vpn evpn
```

```
      redistribute ospf 1 route-map OSPF_OUT
      vrf Tenant-1
       address-family ipv4 unciast
        advertise l2vpn evpn
        redistribute ospf 1 route-map OSPF_TENANT1

    Inspecting these routes on VTEP 1

    show ip route ospf-1 vrf OUTSIDE-VRF
    IP Route Table for VRF "OUTSIDE-VRF"
    '*' denotes best ucast next-hop
    '**' denotes best mcast next-hop
    '[x/y]' denotes [preference/metric]
    '%<string>' in via output denotes VRF <string>

    192.168.100.0/24, ubest/mbest: 1/0
        *via 10.30.1.2  Vlan3001, [110/41], 1w5d, ospf-1, intra
```

The OSPF routes are advertised by the VTEP into the VXLAN Fabric. All other VTEPs will import these routes in each VRF, pointing to VTEP A as the next hop. The example below shows the routing table on VTEP 1. VTEP A's IP address 10.30.1.2 (OUTSIDE-VRF) is the next hop.

```
    VTEP1# show ip route 192.168.100.0/24 vrf OUTSIDE-VRF
    IP Route Table for VRF "OUTSIDE-VRF"
    '*' denotes best ucast next-hop
    '**' denotes best mcast next-hop
    '[x/y]' denotes [preference/metric]
    '%<string>' in via output denotes VRF <string>

    192.168.100.0/24 ubest/mbest: 1/0
        *via 10.30.1.2%default, [200/41], 1w1d, bgp-65500, internal, tag 65500
(evpn) segid: 55555 tunnelid: 0xa010112 encap: VXLAN
```

Traffic from VTEP 1 will be encapsulated towards VTEP A, decapsulated and sent to the firewall. The firewall enforces the policy and sends the traffic back to VTEP A on the INSIDE-VRF. VTEP A will encapsulate the traffic and send it to the destination VTEP 2 where traffic is decapsulated and sent to the endpoint.

Firewall Failover

When the active firewall fails and the standby firewall takes over, routes are withdrawn from services VTEP A. As the previous standby becomes active, routes are now advertised to the fabric through services VTEP B.

If it is not desirable to run a dynamic routing protocol on the firewall, there is a need for static routes pointing to the firewall as next hop. It is critical to ensure that only the VTEP serving the active firewall is advertising the static route.

The first way to accomplish this task is to track active firewall reachability by validating it is locally learned via HMM (Host Mobility Manager). The second approach is to configure the static route at all the compute VTEPs instead of the services VTEPs. Both approaches are introduced to ensure that only the route towards the service VTEP with the active firewall is used.

The approach using HMM tracking ensures that if the active firewall is connected to VTEP A, only VTEP A will have and advertise the static route. VTEP A will track how the static route's next hop (firewall IP) is learned. Only if the next hop is learned as an HMM route (directly connected), VTEP A will advertise the static route through redistribution. If the active firewall fails and the standby takes over, VTEP A starts to learn the next hop IP through BGP and VTEP B starts to know the firewall's IP address as next hop through HMM. VTEP A will then withdraw the tracked routes and VTEP B starts advertising its routes into the fabric.

For example:

```
VRF context Tenant-1
  ip route 0.0.0.0/0 10.30.2.2 track 10
track 10 ip route 0.0.0.0/0 reachability hmm
```

```
      vrf member Tenant-1

    VTEPA# show track 10
    Track 10
      IP Route 0.0.0.0/0 Reachability
      Reachability is UP

    VTEPA# show ip route 0.0.0.0 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 10.30.2.2 [1/0], 00:00:08, static

    Firewall Failure on VTEP A caused the track to go down causing VTEP A to
withdraw the static route

    VTEPA# show track 10
    Track 10
      IP Route 0.0.0.0/0 Reachability
      Reachability is DOWN

    VTEPA# show ip route 0.0.0.0 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>

    Route not found
```

In this case, where the static route is configured in the compute attached VTEPs (VTEP 1 and VTEP 2), no additional configuration is necessary as recursive route lookup will ensure that the static route is only active if the next hop is reachable. Only the VTEP with the active firewall will advertise the firewall IP. This approach ensures that traffic will only be routed towards the VTEP with the active firewall.

```
    VRF context Tenant-1
      ip route 0.0.0.0/0 10.30.2.2

    VTEP1# show ip route 0.0.0.0 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 10.30.2.2 [1/0], 00:00:08, static

    VTEP1# show ip route 10.30.2.2/32 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>

    10.30.2.2/32 ubest/mbest: 1/0
        *via 10.254.254.111%default, [200/41], 1w1d, bgp-65500, internal, tag
65500 (evpn) segid: 50000 tunnelid: 0xa010112 encap: VXLAN

    Firewall Failure on VTEP A (10.254.254.111) caused the recursive lookup to
change toward VTEP B (10.254.254.112)
```

```
VTEP1# show ip route 10.30.2.2/32 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>

10.30.2.2/32 ubest/mbest: 1/0
    *via 10.254.254.112%default, [200/41], 00:00:01, bgp-65500, internal, tag
65500 (evpn) segid: 50000 tunnelid: 0xa010112 encap: VXLAN
```

Integrating Application Delivery Controllers

ADC is another category of network service that many applications require. The introduction described the different deployment modes to bring an ADC into a network. This section focuses on the one-arm design with source NAT.

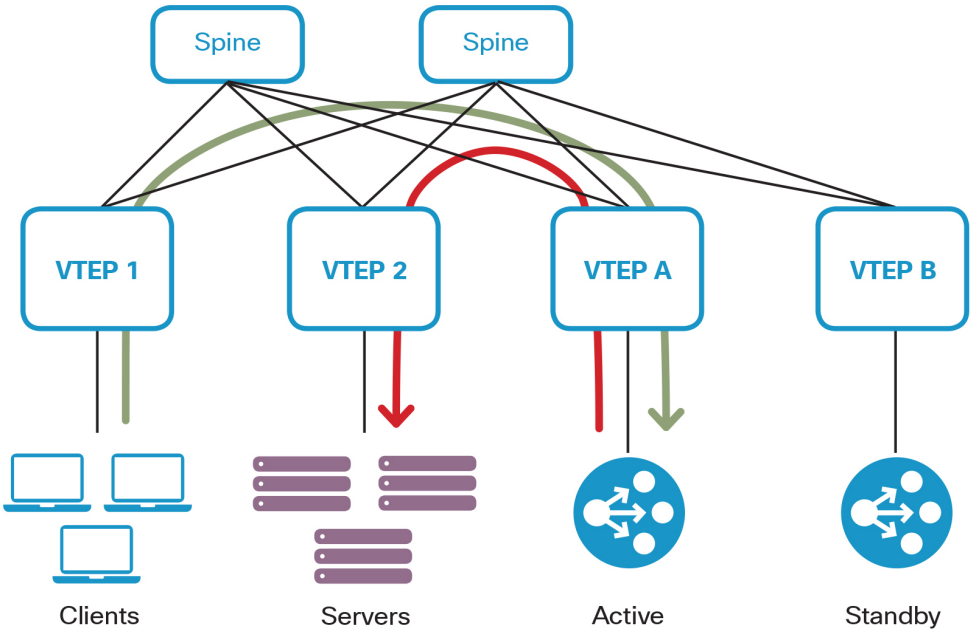
In the one-arm deployment model, the traffic flow is as follows:

- Client traffic enters the interface presenting the virtual IP address (VIP)
- ADC decides which real server to send the request to
- ADC then translates the destination address, which was previously the VIP, with the IP address of the real server.
- The request towards the real server is exiting the same interface as the client request came from
- The source IP address is translated via source NAT
- The real server will see the ADC IP address as the source IP

The ADC is connected to a service VTEP or a pair of service VTEPs with vPC. ADCs are commonly deployed as a High-Availability pair. Within the HA pair, the active ADC advertises the VIP to the service VTEP. This can be achieved by simple MAC/IP learning

advertised in the VXLAN Fabric as an EVPN Type-2 route. Alternatively, the ADC can be implemented with a dynamic routing protocol and advertise the VIP as an EVPN Type-5 route.

Figure: ADC Traffic Flow



Traffic flow is as follows:

- Client traffic will be encapsulated by VTEP 1 towards services VTEP A
- VTEP A decapsulates and sends the traffic to the active ADC
- The ADC sends the traffic destined to the real server back to services VTEP A
- VTEP A encapsulates and sends the traffic to the destination VTEP 2
- Traffic gets decapsulated at VTEP 2 and sent to the real server
- The response back from the real server is sent back to the ADC, since the ADC performed source NAT

Application Delivery Controller Failover

When the active ADC fails and standby one takes over, routes are withdrawn from the services VTEP A. The newly active ADC advertises the VIP on service VTEP B, the same way as the previous active ADC has done it on VTEP A. Service VTEP B is now responsible for requests towards the ADC VIP.

Integrating Service Chaining

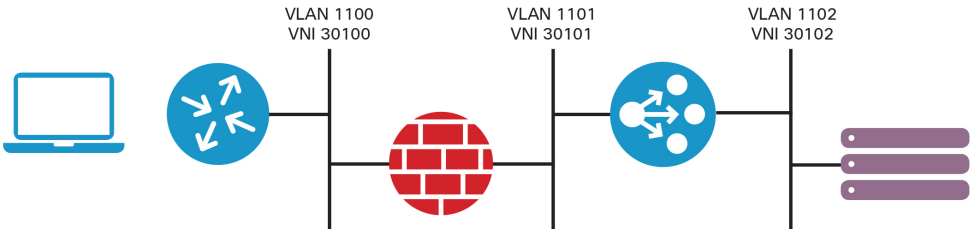
Typically, organizations find that Layer 4-Layer 7 services are not inserted individually, but as a chain. For example, a certain application might require security services from a firewall, and load balancing services from an Application Delivery Controller. Tying these network services together is commonly referred to as a service chain.

Consideration needs to be given to placement of the devices so traffic does not take excessive hops across the fabric when going between the firewall and the load balancer. It is common to have multiple firewalls and ADCs connected to a dedicated pair of switches as service nodes. The placement of the appliances in the VXLAN Fabric is consolidated to a pair of services under a service node pair. Traffic flow is as follows:

- Traffic will be VXLAN-encapsulated from the client VTEP 1 towards the services VTEP A.
- The service VTEP responsible for the active firewall decapsulates and sends the traffic to the active firewall.
- The firewall then sends the traffic towards the ADC's VIP address. This is done with the assumption that the firewall and the ADC are connected to the same service VTEP. If firewall and ADC are on different VTEPs, traffic will be VXLAN-encapsulated towards the service VTEP hosting the ADC.
- ADC then sends the traffic destined to the real server back to the services VTEP, which encapsulates and sends it to the destination VTEP 2.
- Traffic gets decapsulated at VTEP 2 and sent to the real server.
- The response back from the real server is sent back to the ADC as the ADC is using source NAT. With the usage of source NAT, the X-Forwarded-For HTTP header field is going to be inserted to preserve client IP address visibility. Subsequently, the traffic will be inspected by the firewall on its way back to the client.

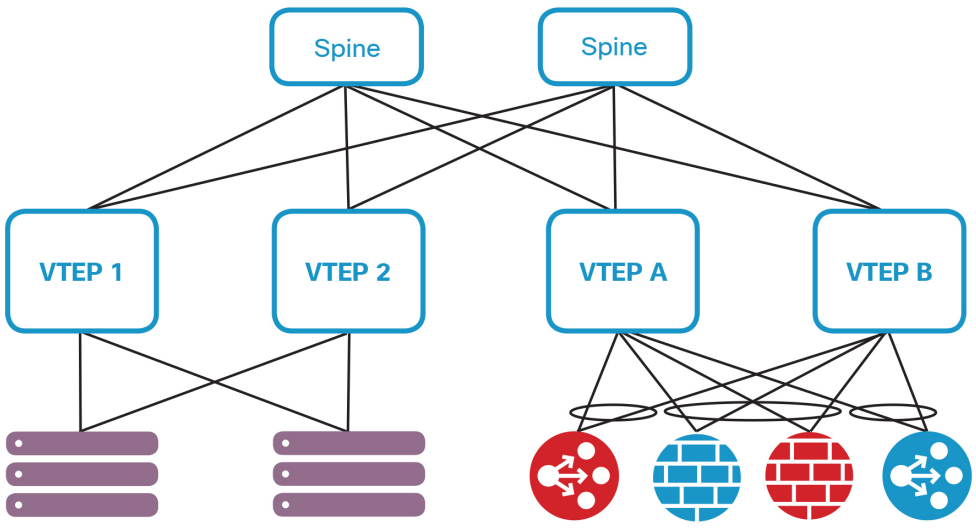
The diagram below shows a logical representation of a service chain.

Figure: Layer 4-Layer 7 Service Chain



The diagram below shows a physical representation of VXLAN Fabric with a dedicated service VTEP pair. Firewalls and ADCs are commonly connected to the services VTEPs. This can be achieved with or without vPC (vPC shown in diagram).

Figure: VXLAN Fabric with Service Leaf



To avoid additional encapsulations and decapsulations, affinity can be created between the active firewall and the active ADC, and they can be placed on the same services VTEPs.

Multi-POD & Multi-Site Designs

Introduction

In an increasingly competitive, globally connected business environment, organizations are faced with enormous pressures to ensure continuous availability of critical business applications. With digital strategies driving innovative new business opportunities, these organizations are looking for IT infrastructures that offer the agility, performance and availability required to support these new application infrastructures.

When building the IT infrastructure to support these business critical environments, today's data center deployments require geographical diversity and scale, ensuring the ability to deliver rapid scale, high performance and "always on" availability.

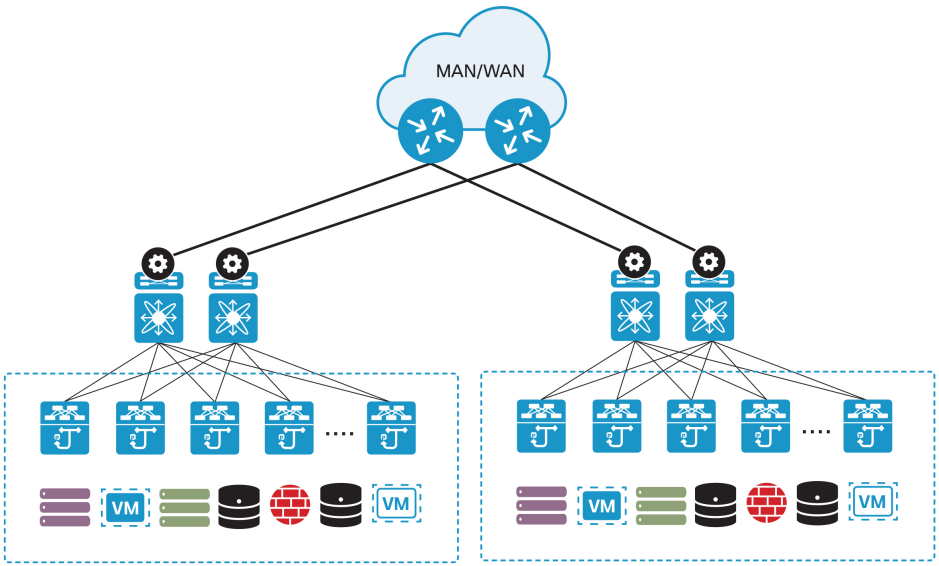
As a consequence, data center networks are building built as scalable, highly available network fabrics which are distributed across multiple data centers, whether separated within or across a metro area, or across the globe.

This chapter presents different deployment options for the interconnection of VXLAN Fabrics, distinguishing between the multi-POD and multi-site approaches based on the specific needs for scalability and availability and the existing physical and operational constraints.

Fundamentals

A Point of Delivery (POD) is a network building block which can easily be replicated within a data center. The predictable and homogeneous characteristics of a POD provide self-containment and a pre-assigned scale and performance requirement (POD planning). The architecture of a POD should be modular to allow for it to be replicated and interconnected, keeping a homogeneous design.

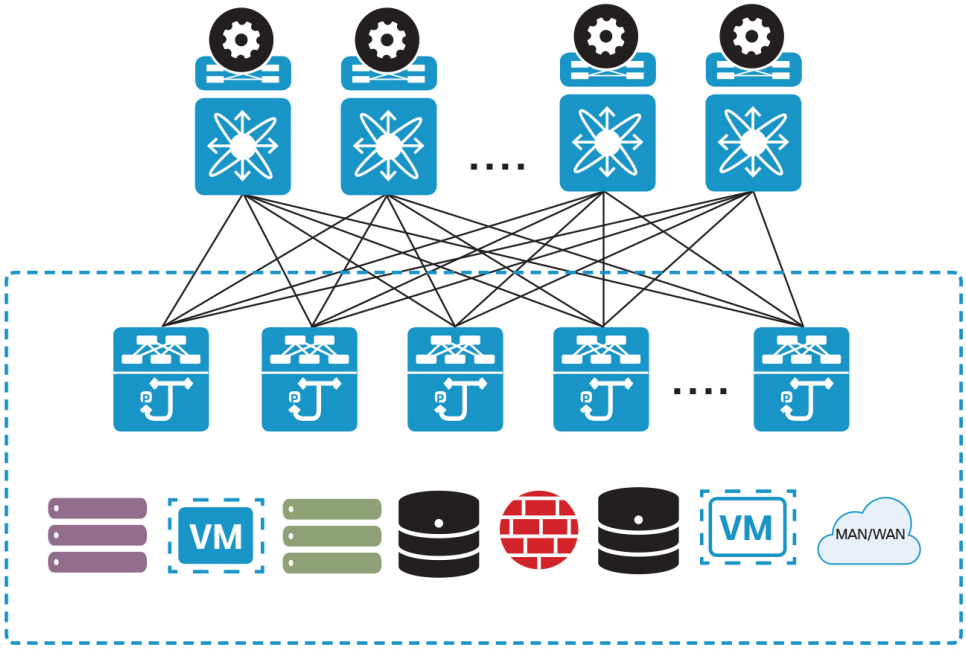
Figure: 3 Tier Architecture



In classic hierarchical network design, the POD is formed by the Access and Aggregation Layer, where the Aggregation Layer provided the Layer 2 demarcation. Layer 2 traffic is terminated and routed across the Core to reach other PODs or external networks. With the demarcation at the Aggregation Layer, a Layer 2 VLAN or an IP Subnet

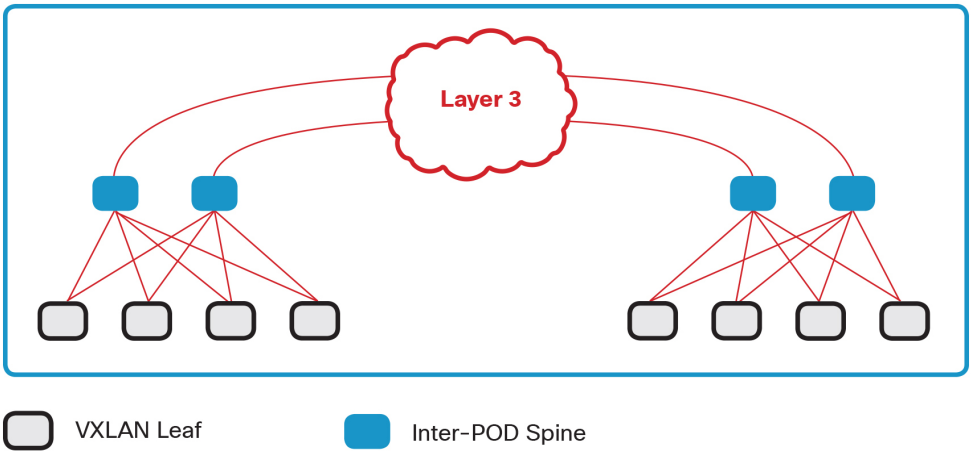
is localized within a single-POD, therefore Layer 2 communication between PODs is not possible. As a consequence, host mobility across PODs is difficult to implement.

Figure: 2-Stage Clos Architecture



With the evolution of the hierarchical Core/Aggregation/Access design into spine-leaf topologies, the location of functions and interconnectivity within a POD shifts. Simply migrating the topology from a hierarchical design to spine-leaf does not bring about a change in functions, however, the addition of overlays introduces greater versatility. With Integrated Route and Bridging (IRB) and VXLAN, the leaf not only provides the default gateway but also a Layer 2 bridging service to other leaf switches. With this approach, it is possible to extend Layer 2 services beyond a single-POD by using an overlay with end-to-end encapsulation. When structuring multiple PODs and enabling extended Layer 2 and Layer 3 services, use cases such as host mobility are now easier to implement.

Figure: Interconnecting Two Clos Networks



The interconnection within a multi-POD site can be achieved in various ways. Spines can be interconnected back to back, an additional super-spine layer can be introduced, or PODs can be interconnected at designated leaf switches.

When multiple physical locations are present, multi-site designs come into consideration. A site defines a set of PODs (multi-POD) which share the same domain constructs, providing the same set of Layer 2 and Layer 3 segments at a given physical location. As a result, within a given site the end-to-end overlay encapsulation starts at a leaf in one POD and can extend to a leaf in another POD.

Multi-POD designs can be stretched across physical locations, however, this is not the recommendation given the high availability requirements for geographically dispersed data centers. Further design aspects are covered throughout this chapter.

In a multi-site design, the most significant aspects to consider are how to connect the sites with each other at the control and data plane levels. When considering north-south connectivity, the first option in a multi-POD design is to consolidate all external connectivity into a single point of access. Alternately, single points of access for each individual POD for distributed ingress and egress forwarding can be defined.

The east-west communication has to solve the challenge of connecting sites to each other allowing workload mobility, but at the same time isolating the sites so that they are independent from each other from a business continuity perspective. A fault in one site should not propagate to the other.

Why Deploy Multiple PODs?

The optimal way to efficiently scale a system is through modularity. Any monolithic architecture will only grow to a certain point, after which inefficiencies will appear. A data center is an example of a system that requires a flexible way to scale the network infrastructure. Frequently a data center build-out starts in a single room and later expands across multiple rooms.

Beside scale, physical facility and infrastructure layouts can be another motivation for multi-POD designs. Multi-POD designs fit very well in situations where a physical location is partitioned across multiple rooms with limited cabling, but maintaining end-to-end Layer 2 and Layer 3 connectivity is still required. Any service within one POD can be made available to any other POD within this multi-POD topology. As an example, consider a high availability (HA) cluster being deployed at a single physical location but spread across different rooms due to the site's local HA capabilities (different Power Distribution Unit - PDU, Uninterruptible Power Supply - UPS etc.).

Why Deploy Multiple Sites?

Modern data center environments must meet the needs for high availability within the data center and across geographically-distributed data center infrastructure. This type of distributed architecture offers multiple benefits for highly available application delivery. Applications can be delivered in an active/active or active/standby deployment model and form the foundation for an effective business continuity or disaster recovery strategy.

There are many factors which determine the applicability and design of the multi-site data center environment including physical constraints such as site location and requirements for geographical diversity. Other considerations include bandwidth and service availability for infrastructure such as dark fiber or wavelength service, and la-

tency which may impact application performance. These factors determine the Recovery Point Objective (RPO) and Recovery Time Objective (RTO) for application availability.

In contrast to a single site deployment a networking solution for multiple sites must also address the need to maintain a level of separation. Any event whether planned or unplanned impacting one site should not spread to any other site as it would impact overall application availability.

When deploying a network infrastructure based on VXLAN EVPN, the consistent delivery of Layer 2, Layer 3 and IP multicast services must be maintained. Together, these allow for the delivery of distributed application architectures and geographically-dispersed clustered infrastructure to support highly available storage access and compute virtualization.

Design criteria to be considered for such deployments include:

- **Physical Connectivity**: In many cases, given the constraints outlined above, the availability of connectivity services may be limited. As an example, dark fiber or wavelength services availability may be limited or cost-prohibitive over large distances, whereas a routed Layer 3 or MPLS service may be readily available at an achievable price point. The design must take into consideration the need to allow for multiple connection types ranging from high bandwidth dark fiber through to bandwidth-constrained service provider-delivered Layer 3 services.
- **Fault Isolation**: When connecting multiple discrete network environments together, the risk of a failure event propagating between sites increases significantly unless controls are applied to restrict the control plane and data plane activity. Examples include selection and configuration of control plane protocols such as BGP, and the control or restriction of data plane activity such as ARP suppression/spoofing and storm control.

Based on these criteria, the multi-site solution must deliver the appropriate set of features and functionality required to meet the specific demands of a particular deployment.

In subsequent chapters the options for multi-POD and multi-site deployment are explored further, including back-to-back vPC, OTV, and PBB-EVPN for a comprehensive DCI solution in order to maintain control plane and data plane isolation and at the same time provide workload mobility.

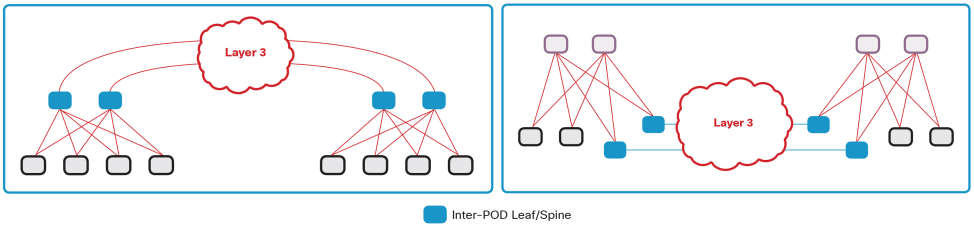
Multi-POD Design

With a multi-POD design, multiple data center PODs are interconnected using redundant Layer 3 paths and run the same VXLAN EVPN control plane. Each POD can have its own Clos Fabric architecture with independent spine and leaf layers. Physical connectivity between PODs can be established by interconnecting them on either the leaf or spine layer. This session discusses different options in the multi-POD fabric design.

Placement of Inter-POD Connecting Points

The physical connections between PODs provide the Layer 3 path for the EVPN control and data planes, as a part of the underlay IP transport network. Therefore, the inter-POD links do not bear any special functional requirements for VXLAN EVPN. The minimum requirements for inter-POD links are Layer 3 IP unicast and multicast routing in the underlay. Placement of the inter-POD connecting points is flexible as it can be either on a leaf node or a spine node. Consideration around the link speeds, the optic types, and/or the cabling plan could be driving factors for the decision to interconnect PODs on leaf or spine nodes. Figures below illustrate the topologies options.

Figure: Multi-POD Topology with Inter-Connections on Leaf Nodes or Spine Nodes



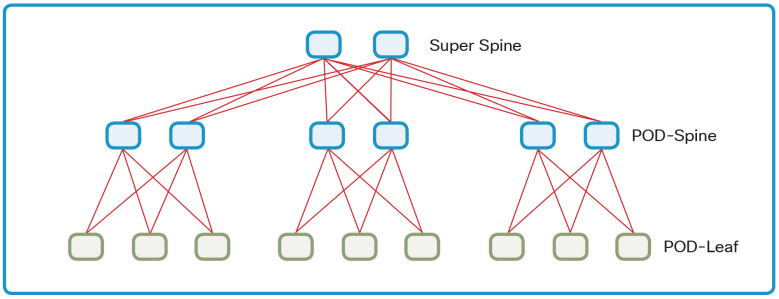
To achieve high availability for inter-POD connectivity, the recommendation is to leverage redundant inter-POD paths using two or more devices in each POD. Since the inter-POD connections only need to provide Layer 3 connectivity, the redundant interconnecting devices do not need to be in vPC pair.

Scale Multi-POD with Multi-Stage Clos Architecture

When the number of PODs increases, simple yet scalable multi-POD design becomes an important decision point. Following the n-stage data center fabric design principle, one design option to connect multiple data center PODs is to introduce a super spine layer that interconnects the spine layer of each POD.

This essentially builds a multi-stage hierarchical fabric topology. MP-BGP EVPN is running between the Fabric nodes to distribute the VXLAN EVPN routes. This multi-stage fabric design with a super-spine layer simplifies the interconnection topology among PODs, making it easier to scale the number of PODs. It is the most efficient way of providing consistent forwarding hop counts for inter-POD traffic. If underlay multicast replication is used to transport the VXLAN Fabric BUM traffic, a multi-stage Clos Fabric design also helps reduce the number of Multicast Output Interfaces (OIF) required. Since most switch platforms support a limited number of multicast OIFs, the super-spine will allow the VXLAN Fabric to scale without exceeding the maximum number of OIFs supported on a single spine device.

Figure: Multi-Stage Fabric



Scalability

When designing for control plane scale for an inter-POD Fabric, platform OIF, multicast groups, and VTEPs need to be considered in addition to host MAC and MAC/IP. It is important to look at the hardware verified scalability guidelines.

For example, in a simple multi-POD scenario, if the spine supports 256 OIF, then subtract 2 OIF for the uplink towards the L3 core, leaving 254 OIFs for southbound connectivity to the leafs in the vPC domains. This would give 254 leaves, or 127 vPC domains, to connect southbound if each leaf in the vPC domain has a single link to each spine in the POD.

Looking closer at the above example, both vPC VTEP switches independently send the IP PIM register to the Rendezvous Point for the multicast group of the VXLAN VNI. Both source the register packets from the anycast VTEP address and each installs the corresponding (*, G) entry in their multicast routing tables with the VTEP interface (NVE1) in the output interface (OIF) list.

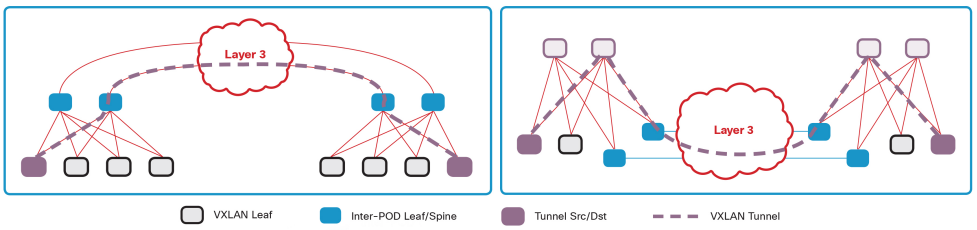
In addition, consideration needs to be given to host MAC and IP scale per leaf. A leaf will learn all BGP routes across the multi-POD environment but will not program the hardware tables Forwarding Information Base / Routing Information Base (FIB/RIB) unless the leaf needs to know about them. If the leaf knows about the VRF and is importing the route-targets it will program the RIB for the MAC/IP routes. In addition, the leaf only programs the FIB with the MAC address of the VNIs of the VRFs it has locally defined.

Building the Overlay

Data Plane Operation

Like a single-POD, the tunnels run between VTEP devices in a multi-POD fabric. In a multi-POD fabric the tunnel headend VTEPs can reside in different PODs if the traffic traverses the POD boundary, which results in an inter-POD tunnel. VXLAN encapsulation and decapsulation only take places on the ingress and egress VTEPs. The other devices along the forwarding path only need to route the encapsulated VXLAN packets. This provides very efficient end-to-end, single-tunnel overlay data plane processing.

Figure: Data Plane Operation

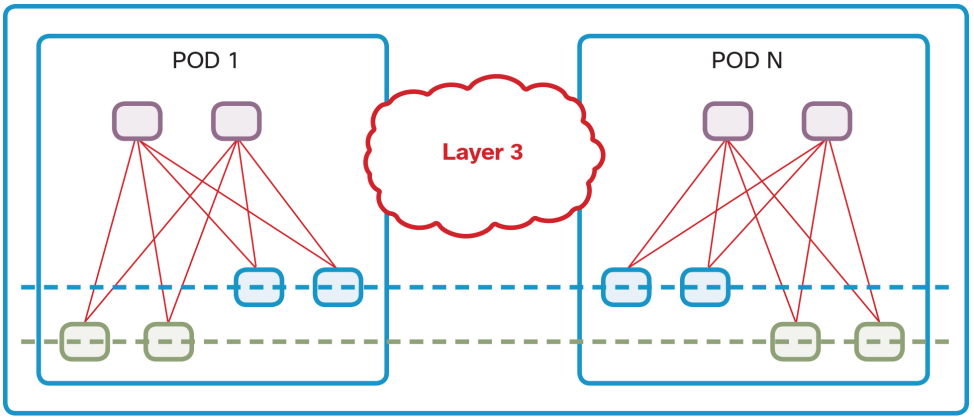


IP Gateway Localization

In networks without Distributed Anycast Gateway the default gateway is made redundant through the use of a First Hop Redundancy Protocol (FHRP). When a network segment spans across multiple physical locations, the same concept can force all traffic through single VTEP. Alternatively you can provide localization by having an active instance of the default gateway in each location. Using localization provides an more optimal forwarding path between subnets within the same location. If application workload mobility is required between locations, it is important to maintain the same default gateway IP and MAC address. With gateway localization, endpoints do not need to relearn these information at the new location.

Multi-POD VXLAN Fabric with Distributed Anycast Gateway makes it easy to meet this requirement. Similar to single-POD, all the VTEPs serving the same IP subnet can use the Distributed Anycast Gateway.

Figure: Gateway Localization with Distributed Anycast Gateway



In the illustration above, the VTEP leafs in blue are Distributed Anycast Gateway for Layer 2 VNI "Blue" while the VTEP leafs in green are Distributed Anycast Gateway for Layer 2 VNI "Green".

Control Plane Operation

EVPN MP-BGP control protocol runs throughout multi-PODs the same way as it does within a single-POD. The fabric nodes running EVPN exchange MP-BGP EVPN routes with one another. Each VTEP device detects its local endpoints and installs HMM routes for endpoint tracking. The HMM routes are automatically injected into MP-BGP EVPN address-family and distributed to other EVPN nodes as EVPN type-2 routes. Upon receiving the EVPN routes, the rest of the VTEP devices will install the endpoint reachability information into their L2 RIB and L3 RIB tables. Further programming of the hardware forwarding tables, including the MAC-address table and host/LPM forwarding tables based on the RIB information will happen if they possess the corresponding L2VNI and L3VNI information.

Placement of MP-BGP EVPN Peering

In a VXLAN Fabric the same EVPN control plane runs throughout the entire environment. Within a POD, EVPN sessions are formed between leaf and spine nodes. Between PODs, EVPN peering does not necessarily need to coincide with the physical connection topology. The following drawing depicts the available designs in which EVPN MP-BGP peering occurs between the connected leafs or between the spine nodes of different PODs.

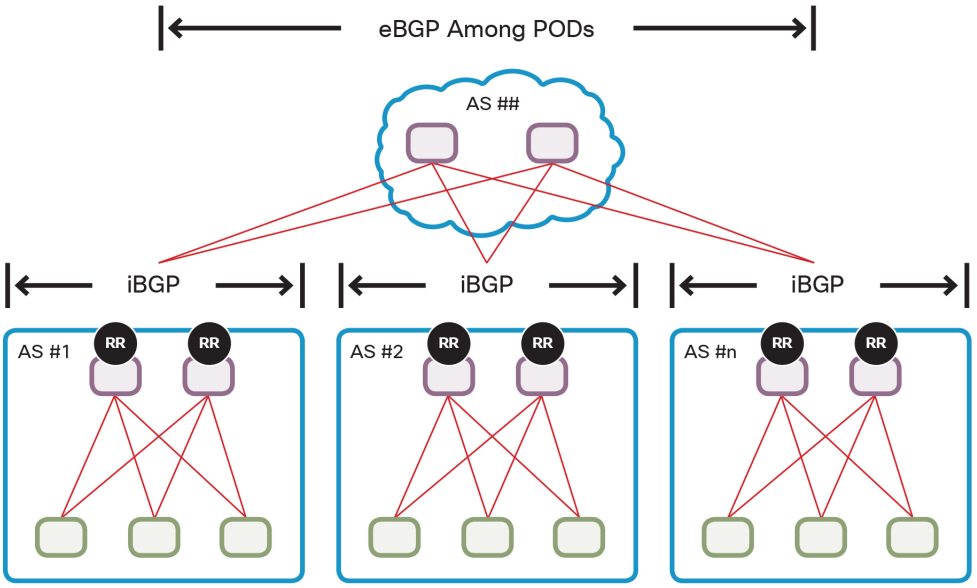
Often, switch hardware platforms with more control plane capacity and higher bandwidth are chosen for the spine layer. Also, due to their centralized location in a POD, the spine nodes are often chosen as the control point for MP-BGP EVPN route distribution. For example, in a MP-iBGP Fabric, the spine nodes are often chosen to be the iBGP route reflectors. In this case, peering on the spine nodes between PODs can take advantage of the more scalable control plane and the complete set of EVPN routing information on the spine nodes.

MP-iBGP vs MP-eBGP

MP-BGP EVPN distributes the Layer 2 and Layer 3 reachability information for the VXLAN overlay network. It supports both iBGP and eBGP topology, which provides the design flexibility to run MP-BGP in a multi-POD environment. It is not within the scope of this book to document all the possible combinations of iBGP and/or eBGP designs in a multi-POD Fabric. The common practice designs will be discussed to illustrate the design principles.

The Figure below describes a common multi-POD design in which each POD runs MP-iBGP EVPN between leafs and spines whereas MP-eBGP EVPN is used to interconnect the PODs. The drawing does not indicate any physical topology for connecting multiple PODs together rather, it depicts the peering topology. Conceptually, the Route Reflectors (RR) of different PODs are exchanging EVPN routes via MP-eBGP so that reachability information can be extented from one POD to another.

Figure: eBGP Peering Among PODs

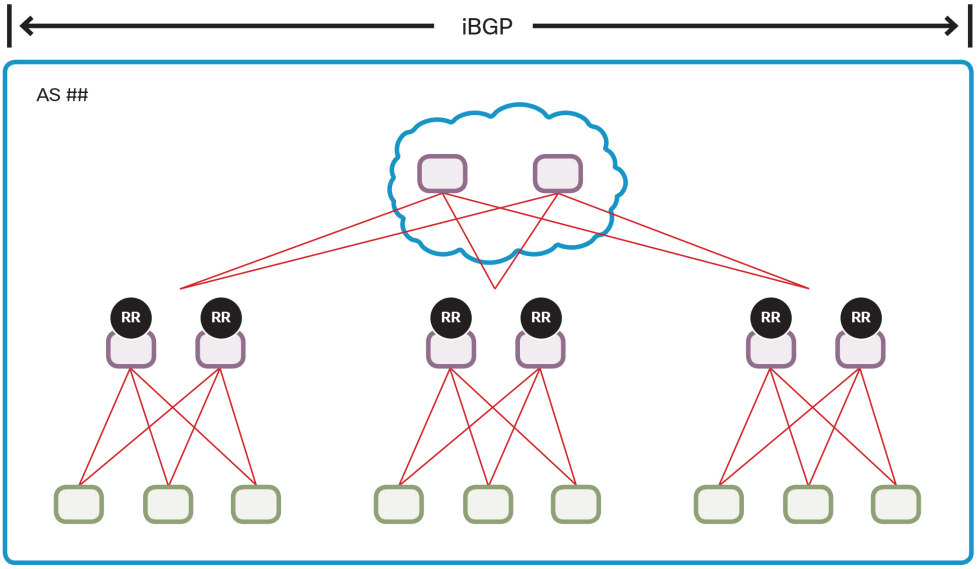


With this design, BGP peering among multiple PODs is simple. EVPN routes can be distributed among PODs through MP-eBGP peering without the need for additional configuration. Additional considerations need to be given to how to preserve the attributes in an EVPN route when it is distributed within the Fabric as eBGP default behavior may cause some of the attributes to be overwritten:

- By default, a router overwrites the next-hop in the route to itself when sending a route to its eBGP peers.
- If each AS generates EVPN route-targets (RT) automatically, they may end up having different RTs for the same L2VNI or L3VNI as often the auto-RT function uses the BGP AS number as one of the elements to derive EVPN RTs. So additional caution needs to be applied when configuring the EVPN RT import and export policies to ensure the routes within the same VNI shall have the same import/export RTs on VTEPs with different PODs so that the route distribution can be complete end-to-end.

Another design is to use a single BGP AS across all PODs so that the multi-POD Fabric runs EVPN MP-iBGP.

Figure: iBGP Peering Among PODs



With this design, additional iBGP design principles need to be applied to ensure EVPN routes are distributed end-to-end through the BGP AS. As a loop prevention mechanism, iBGP has the rule that routes learned from one iBGP peer will not be advertised to the other iBGP peer. That is the reason why iBGP route reflectors (RR) are needed to reflect routes between the peers. In this multi-POD design, RRs from different PODs are further interconnected within the same BGP AS, there is a need for another layer of RRs to pass MP-iBGP EVPN routes among the POD RRs. However, by design MP-iBGP preserves EVPN route attributes better than MP-eBGP.

- iBGP by design preserves BGP next-hop. Therefore, when an EVPN route is distributed within an iBGP topology, the originating VTEP address will be preserved in the BGP next-hop.
- iBGP does not change the EVPN Route-Target (RT) value while distributing the routes.

- Auto-RT function will generate the same EVPN RT for the same VNI across different PODs. This ensures that VTEPs in different PODs will have consistent import and export RT value for the same VNI.

When comparing the two common EVPN MP-BGP designs, each of them offers simplicity in one aspect while introducing complexity in another. The following table summarizes the comparison.

	MP-eBGP	MP-iBGP
BGP Peering	Simple	Complex
EVPN Route Distribute	Complex	Simple

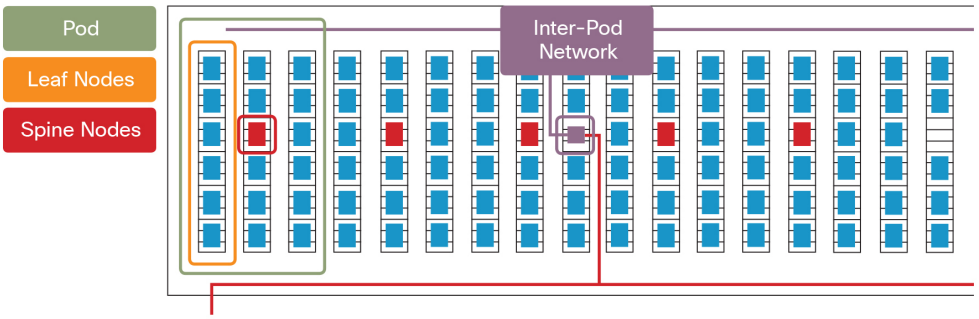
If a multi-POD Fabric is deployed in a network that already has a BGP deployment, the decision on whether to use MP-iBGP or MP-eBGP peering will depend on the existing BGP deployment. It is worth noting that in the context of multi-POD design, the advantage of using different BGP AS's for better control plane segmentation is not significant as the entire multi-POD fabric is under the same administrative scope and MP-BGP EVPN domain.

Building the Underlay

Cabling

Most of the existing cabling infrastructures were designed to handle 3-tier physical cable layouts which may lend itself to a multi-Pod topology due to limited cabling capacity. In N+1 POD environments, a cabling infrastructure providing central core connectivity will address scale out and facilitates adding more capacity. Another cabling option is multi-POD using dark fiber, MAN or DWDM.

Figure: Cabling Infrastructure



IP Multicast Replication vs Ingress Replication

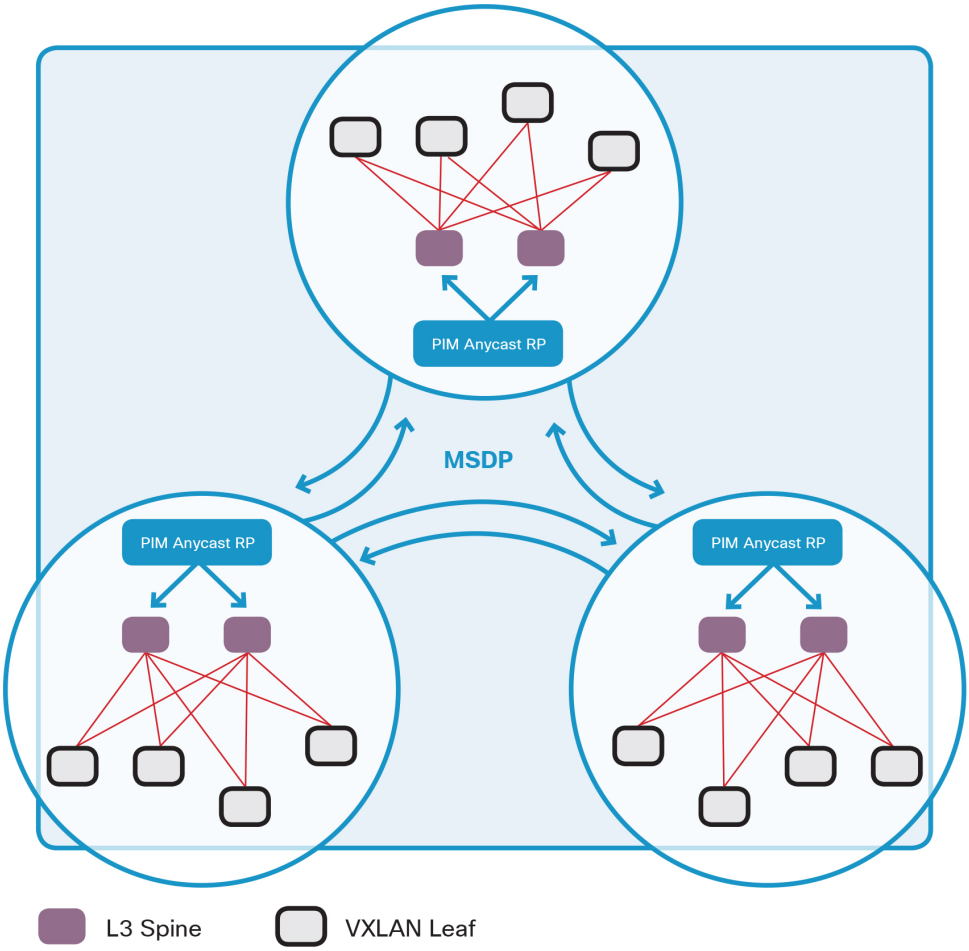
The MP-BGP EVPN control plane is used to discover endpoints and exchange host information between leaf nodes, IP Multicast or Ingress Replication is used for BUM traffic. There are two ways to replicate BUM traffic throughout the fabric, either one to many, or one to one many times. ARP requests are one example of BUM traffic that needs to be replicated using a multicast group or ingress replication.

Ingress replication can have scale issues as the switch needs to replicate BUM packets as many times as there are VTEPs that own the VNI needing to see that traffic. As an example, with 50 VTEPs that own the same VNI that require BUM traffic, replication needs to be performed 50 times. Replicated BUM transmissions consume a lot of bandwidth in the network. In contrast, IP multicast across a multi-POD environment is a much more scalable solution to handle BUM traffic as the fabric natively provides the capabilities for the required replication. IP multicast reduces network load, improves performance, and increases scalability across multi-POD environments.

When an anycast RP is configured, the restriction of having one active RP per multicast group instead deploy redundant RPs for the same group range. The RP routers share a single unicast IP address between PODs. This method provides RP redundancy and load sharing within the domain. Sources from one RP are known to other RPs in other PODs using the Multicast Source Discovery Protocol (MSDP). Sources and receivers use the closest RP, as determined by the IGP. During an RP failure, sources and receivers seam-

lessly failover to a new RP based on the underlay routing domain. In multi-POD environments, PIM-SM RP and RP redundancy should be positioned locally inside of each respective POD. Anycast RP clustering can be used for RP redundancy across PODs but for better control of the multicast environment MSDP is the recommended solution to connect multiple PIM-SM domains.

Figure: MSDP for Inter Site Multicast



More details of configuration examples can be found at http://www.cisco.com/c/en/us/support/docs/ip/ip-multicast/115011-anycast-pim.html

Multi-POD Routing Design

Multi-POD IP Routing designs need to take into consideration the requirements for both underlay and overlay. The underlay will establish Layer 3 connectivity between VTEPs deployed across multiple PODs.

The underlay should only consist of VTEP reachability information for all networking devices in the fabric.

When interconnecting an EVPN multi-POD environment, it is important to maintain as much as POD independence as possible. In very large multi-POD environments it may be beneficial to have multiple IGP areas to improve fault tolerance across the PODs. As an example, each POD can be optionally a Stub Area. As a Stub Area, each DC area knows its own topology and has a default route towards the border leaf; while the backbone area has a view of the full multi-POD Fabric. However, for most designs a single area across multiple PODs will suffice as simplicity outweighs complexity.

The underlay can be built with any routing protocol. BGP may not be the best choice as an underlay protocol as it is a distance vector routing protocol and it does not take into account link speed or path cost, and in a multi-POD environment multiple paths with different link speeds might be used to interconnect the PODs. Driving simplicity in the routing design in the underlay will help to improve overall convergence in the overlay. Tuning IGP timers may help improve convergence time, however, there is no generic recommendation, and this must be qualified and validated for each deployment.

Other IGPs such as OSPF, ISIS would be a better option for underlay routing. Please refer to the Single-POD design chapter for a more detailed discussion on routing protocols.

Multi-POD			
Underlay Control-Plane	Single Area (IGP)	Single Area (IGP)	Multi Area (IGP+BGP)
Overlay Control-Plane	Single AS (EVPN)	Single AS (EVPN)	Multi AS (EVPN)
BUM (Multicast)	Anycast RP (PIM)	Anycast RP (PIM | MSDP)	Anycast RP (PIM + MSDP)
	Single-POD (2-Stage CLOS)	**Multi-POD (Single AS)**	**Multi-POD (Multi AS)**

Service Integration

In a multi-POD design, it is a recommended practice to have all the services infrastructure such as firewalls or load balancers connected to a separate services node POD. This helps with scalability and high availability for services across a multi-POD design.

Design Options

When connecting data center sites based on VXLAN Fabrics, there are a number of design considerations which will determine the overall performance, availability, and scale of the environment.

These design considerations include the following aspects.

- Determining the Inter-Site Border Connection Points: The Fabric border provides an edge function to allow for external connectivity in and out of the Fabric and also provides an attachment point for the DCI services which deliver the required inter-site connectivity. Although the Fabric border for Layer 2 Layer 3 External Connectivity and DCI services have similar characteristics, they may or may not be combined depending on factors detailed in the External Connectivity chapter.
- DCI Service Delivery: An appropriate selection of DCI service will be a primary factor in the multi-site design as each will have different properties as explained further in the External Connectivity chapter.

Multi-Site Interconnect (DCI)			
Layer 2	IEEE 802.1q (with VPC)	OTV	VXLAN BGP EVPN
Layer 3	VRF-lite	MPLS L3VPN	VXLAN BGP EVPN
Multicast	PIM	MVPN	-
	Ethernet-Based	**Overlay-Based**	**VXLAN-Based**

- L3 services including L3VPN, VRF Lite, LISP or VXLAN
- L2 services including Ethernet over Dark Fibre/DWDM, OTV, PBB-EVPN, MPLS EVPN, VPLS or VXLAN

Figure:Multi-Site DCI over L3 Service

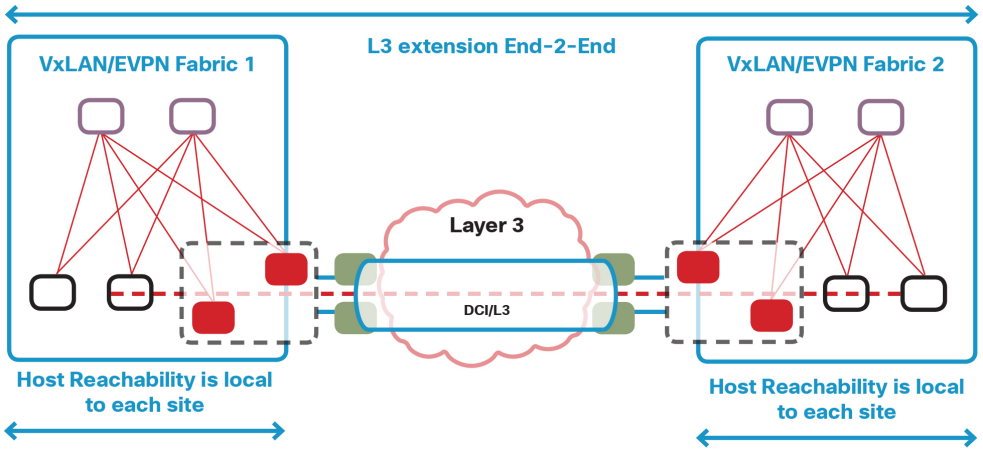
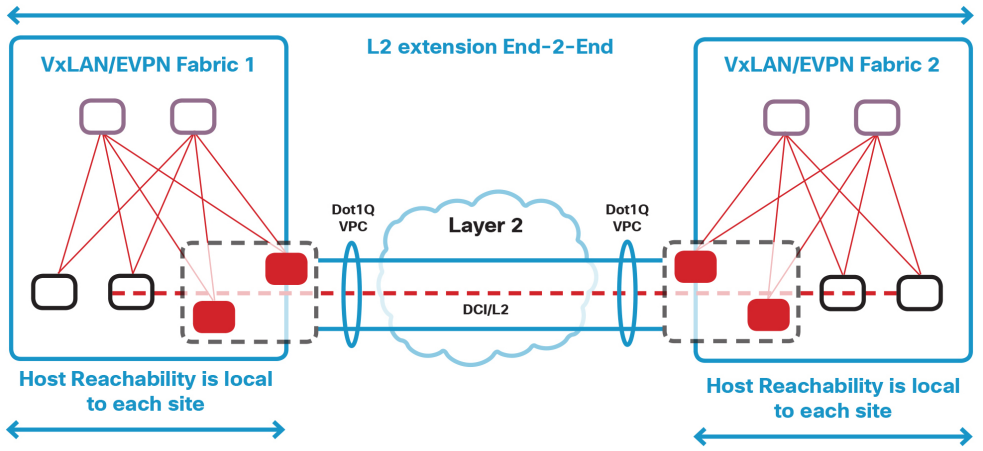


Figure: Multi-Site DCI over L2 Service



Border Leaf Scalability

The border leaf provides the attachment point between multiple Fabrics, delivering Layer 3 routing and Layer 2 extension between sites. In order to perform host routing for the traffic traversing the DCI transport, it must also maintain a host routing table in hardware for connectivity within and across multiple sites.

The key factors which typically determine multi-site scale at Layer 2 and Layer 3 include:

- Virtual Network Identifiers (VNI) - Layer 2 and Layer 3
- MAC Addresses
- IP Host Routes (IPv4/IPv6)

Building the Multi-Site Inter-Connectivity

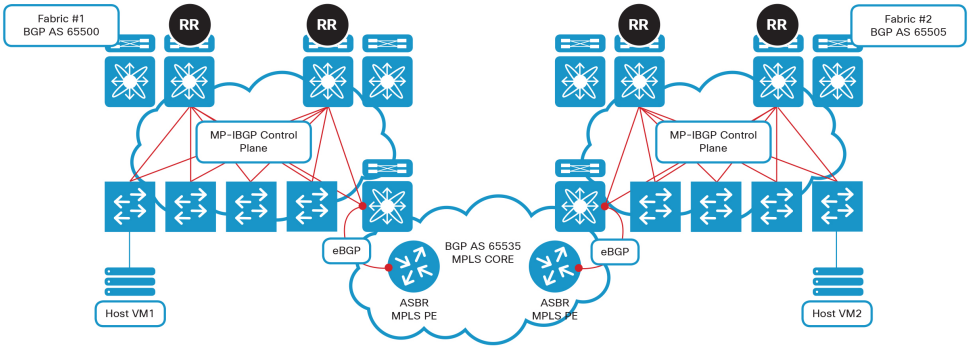
The need for creating multiple sites is to ensure that any impact in one availability zone will have minimal to zero impact on the other availability zones. An independent fabric is one that has its own control plane and data plane. Multi-site provides the ability to interconnect the independent fabrics using a DCI solution such as back-to-back vPC, OTV, EVPN, PBB-EVPN and Layer 3 connectivity with VRF-lite, MPLS or LISP.

A continuously available, active/active, flexible environment provides several benefits to the business:

- Increased uptime
- Disaster avoidance
- Easier maintenance
- Flexible workload placement
- Extremely low RTO

It is important to remember that host reachability information is contained within a single site and extended using a DCI technology. The Layer 3 diagrams below demonstrate independent control planes in each site and will highlight how to extend Layer 2 connectivity.

Figure: Layer 3 DCI for VXLAN Interconnect



To provide true active/active architecture, it is also required to integrate Layer4-Layer7 services such as firewalls and ADCs. Cisco Adaptive Security Appliance (ASA) provides support for multi-site active/active firewall clustering with sites located hundreds of kilometers/miles apart.

Layer 2 Reachability Across Sites

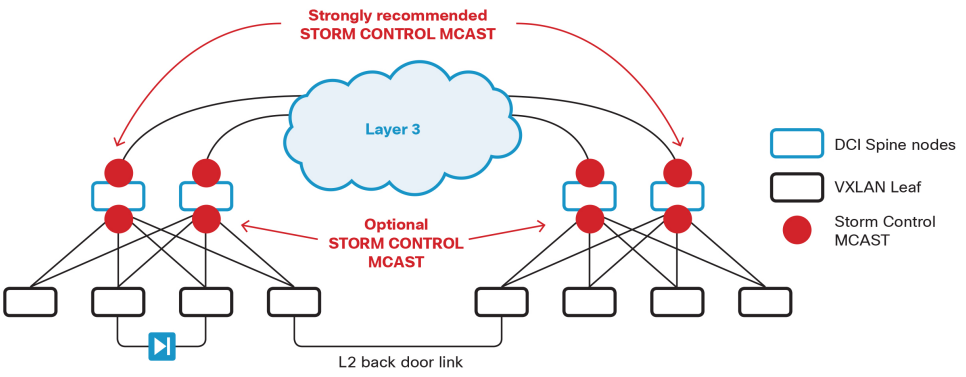
Due to requirements for disaster avoidance and workload mobility, there is a requirement to extend VLANs across different VXLAN Fabrics. The discussion below will cover three DCI options: vPC-based, OTV, and VXLAN.

The DCI solution should provide Layer 2 and Layer 3 extension, and ensure that a failure in one data center will not be propagated to the other data center. To prevent this from happening, the key technical requirement is the capability to control the broadcast, unknown unicast and multicast flood at the data plane level while ensuring control plane independence.

Multi-Site Interconnect (DCI)			
Layer 2	IEEE 802.1q (with VPC)	OTV	VXLAN BGP EVPN
Layer 3	VRF-lite	MPLS L3VPN	VXLAN BGP EVPN
Multicast	PIM	MVPN	-
	Ethernet-Based	**Overlay-Based**	**VXLAN-Based**

Layer 2 extension must be dual homed for redundancy while prohibiting end-to-end Layer 2 loops that would lead to traffic storms causing link overflows, saturate switch CPUs and virtual machine CPUs. This is why in Data Center Interconnect deployments, one key complementary feature to Layer 2 extension is storm control.

Figure: Storm Control



VPC as a DCI Transport

Two VXLAN fabrics can be directly connected using back-to-back vPC. On each side, one pair of border nodes are leveraging a back-to-back vPC connection to extend Layer 2 connectivity across sites. This dual link vPC could use dark fiber or DWDM. vPC is extremely simple to configure however there are certain limitations to using vPC as a DCI such as:

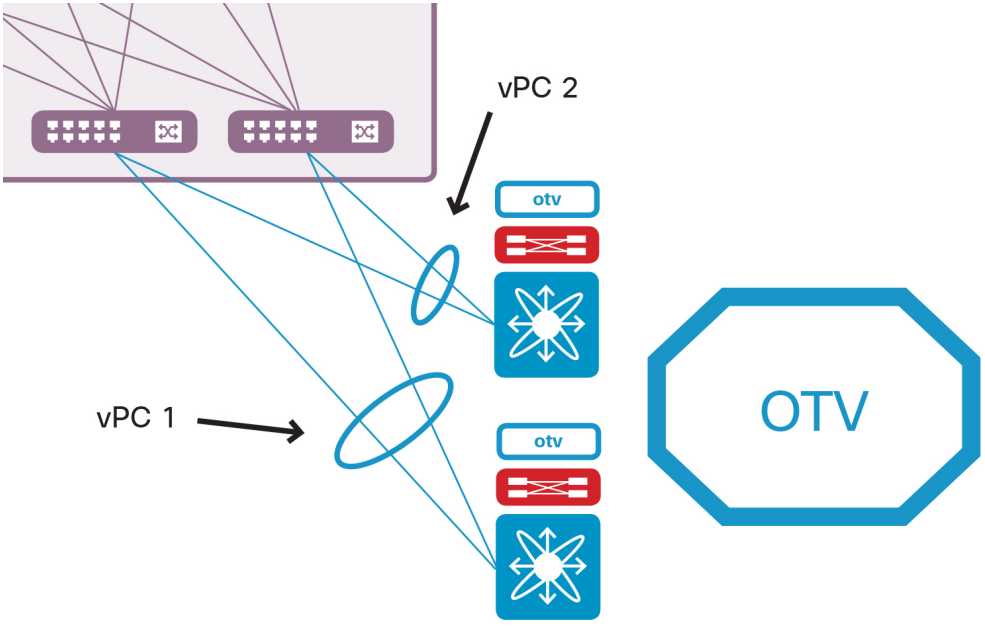
- Cannot interconnect more than two sites
- Lack of failure boundaries
- Site independence is not preserved

OTV as DCI Transport

OTV provides a proven and extremely simple way to interconnect multiple data centers. OTV has been designed for the data center interconnect space, and is considered the most mature and functionally rich solution to extend multi-point Layer 2 connectivity over a generic IP network. In addition, it offers native functions that allow strengthening the DCI connection and increasing the independence of the fabrics.

- Spanning Tree (STP) isolation
- Unknown Unicast traffic suppression
- ARP optimization
- Layer 2 broadcast policy control
- Fault isolation
- Site independence
- Failure boundary preservation

Figure: OTV as a DCI



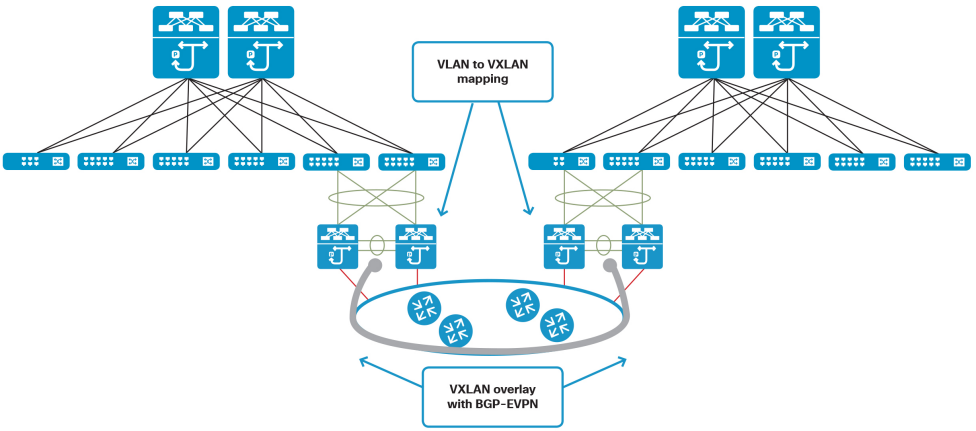
VXLAN as DCI Transport

VXLAN allows Layer 2 extension, not only inside a data center but also the ability to allow Layer 2 interconnections between multiple sites. Today VXLAN EVPN as a DCI offers some benefits but is not yet mature in terms of native functions like provided by OTV.

New functionalities are being added to the VXLAN control plane which would make it a very viable DCI solution in the future. This is further discussed in the introduction chapter.

As depicted in the diagram below, logical back-to-back vPC connections are used between the VXLAN border leaf nodes and the local pair of VXLAN DCI devices to interconnect multiple sites.

Figure: VXLAN as a DCI



Layer 3 Reachability Across Sites

In addition to the Layer 2 solutions, the VXLAN fabric can also be connected across Layer 3 boundaries using MPLS L3VPN, VRF-Lite and LISP.

VRF-lite Based Approach

VRF-lite uses a two-device approach to providing Layer 3 connectivity between multi-site fabrics. In this approach, each tenant VRF in the fabric is extended using a subinterface per tenant with Interior Gateway Protocol (IGP) or External Border Gateway Protocol (eBGP) peering between the border node and the edge router at each site. The External Connectivity chapter discusses in detail the procedure to connect a Layer 3 handoff to the VXLAN fabric. The same principles can be used to provide multi-site interconnect using VRF-lite.

MPLS-Based Approach

This approach uses a single device to interconnect multi-site VXLAN fabrics and achieve segmentation using MPLS L3VPNs. The single device called Border PE can be used to terminate MPLS and VXLAN routing on the same device. The External Connectivity chapter provides additional details about using a MPLS handoff to the VXLAN fabric. The same principles can be used to provide multi-site interconnectivity as well.

LISP-Based Approach

The third approach for interconnecting multi-site VXLAN fabrics is LISP. It offers the same segmentation benefits as MPLS and can be used as an alternative solution. The External Connectivity chapter provides additional details about using a LISP handoff to the VXLAN fabric. The same principles can be used to provide multi-site interconnectivity as well.

Operations & Management

Introduction

This final chapter is focused on providing guidance for operational aspects of building, operating and maintaining a VXLAN Fabric. The latter half of the chapter covers APIs and off-the-shelf and open source tools for automating the management of the fabric.

For the last 20 years, networks have been managed as independent elements leveraging purpose-built protocols and interfaces, such as Simple Network Management Protocol (SNMP), Command-Line Interface (CLI), and NetConf, to name a few. These protocols have served network administrators well, and have mostly fulfilled their objectives for Fault, Configuration, Accounting, Performance and Security management tasks (also known as the FCAPS framework). However, to meet the new scale requirements, the network has to be viewed and managed as a system to enable faster and more consistent delivery of services.

Several years ago, the server industry, driven by scale requirements, went through the same transition. Server teams were faced with the need to manage large pools of resources that drove the need for more automated configuration management tools. Today, server management teams leverage popular configuration management tools such as Puppet, Chef or Ansible. These tools are changing organizational processes which support agile development and DevOps initiatives.

It is questionable when this disruption will impact the network industry; however, the configuration management tools listed above are now able to provide comparable ways to manage network elements. Over time, IT organization will evolve to this new way of managing IT infrastructure, however, it is important that organizations have time to complete this transition. IT systems require the ability to support the existing management paradigms as well as these new models during a transition phase towards more efficient processes.

VXLAN technology will benefit from solutions that can consistently deploy configurations across multiple switches when creating new tenants or new networks. The management and operations of VXLAN will depend on tools that can provide visibility and diagnostic analysis of the underlying infrastructure.

Management tasks

VXLAN Fabrics are no different from other technologies, given that they require foundational infrastructure that needs to be operationally managed in a simplifed manner.

Multiple traditional frameworks exist to define what the operations of IT infrastructure entail, such as IT Infrastructure Library (ITIL) or FCAPS. Some organizations have started to incorporate IT operational practices from other areas of the industry such as application development taken from DevOps (Development + Operations), as is covered in the next section.

Agility is one of the main objectives that most data center leaders covet, and is part of the overall operations process. Automating the data center enables provisioning network resources in a reliable manner while maintaining configuration consistency to reduce downtime. In place of the standard command line interface (CLI), network automation can be used to simplify these commands, for consistency, standardization, and reduction of human error. Network provisioning normally starts with script-level automation and can progress to more advanced models of deployment.

Whatever tools or process are used, whether automation or manual intervention, there are some basic tasks that need to be performed. The network management lifecycle is divided into three different phases:

- Day 0: install
- Day 1: configure/optimize
- Day N: upgrade and monitoring

Day 0

Traditionally, Day 0 activities have included installing the device into a rack, powering it up, some basic bootstrap configuration, and optionally, updating the firmware. This is how many organizations have dealt with Day 0 tasks until now. As a consequence, it is not uncommon to see a network with multiple versions of software deployed and differing standards of configuration. In order to reduce the inconsistencies in the network when the equipment is deployed, automation of the initial deployment is a crucial first step. This provides a solid foundation for successful network operation.

Day 1

After the base configuration and common software releases have been deployed across the Fabric, the next step is to provision the overlay and device-specific configurations. These configuration steps include items such as MP-BGP, Multicast, VNIs, VRFs, VLANs, Anycast Gateway and core capabilities.

The VXLAN Fabric requires more configuration than was previously needed in traditional designs. The burden of the additional VXLAN Fabric configuration requirements can be eased by automation.

For this phase a few options exist to help automate configuration deployment. There are tools such as Cisco Prime Data Center Network Manager (DCNM), Cisco Nexus Fabric Manager (NFM), Python scripts or scripting languages that can configure the devices directly or via API. Another option is configuration management tools (CMT) such as Puppet, Chef, and Ansible that deliver configuration standardization. Instead of just pushing configuration commands to the switches, CMT checks the running configuration and updates changes to the configuration. This allows the creation of manifests, recipes, or playbooks with the desired end state of the specific elements in the network. For example, the spine switches would have a very different configuration than the leaf switches, but the leaf switch configurations would likely be very similar to one another across the fabric.

As a result of virtualization and cloud provisioning, another item to consider is VMM integration. Whether or not the configuration of a switch should be dynamically modified based on a trigger event is discussed at length in the Software Overlay chapter.

Day N

Once the network is configured, running, and optimized, changes and software upgrades to the Fabric will be needed. CMT solutions can automate software upgrades and configuration changes to multiple devices.

Another important Day-N task is configuration backups, revision control, and the ability to roll back to a previous snapshot. This traditional configuration Management can be done with tools such as DCNM, NFM, the aforementioned CMT solutions, or with open source tools such as RANCID.

Monitoring the network and reacting to events is a critical part of Day N operations. Traditional network management tools used SNMP to monitor device parameters such as interface utilization or available memory. With NX-OS programmability functions, using new tools, such as Carbon/Graphite, Zenoss, or Splunk enables access to richer information. Linux-based monitoring agents can be installed natively on the switch. Examples such as OpenTSDB (http://opentsdb.net/) provide a collector agent which sends information to a central repository for consolidation.

Visibility is another important Day-N function. Traditional visibility tools are still available with a VXLAN-based solution including network TAPs, switchport analyzer (SPAN), Netflow and/or sFlow, where applicable. Nexus Data Broker (NDB) clients can be leveraged to consolidate SPAN from leaf switches into a common switch aggregation point to build scalable network TAPs and SPAN aggregation infrastructures.

VXLAN OAM (Operations, Administration and Management)

With the addition of the overlay, a level of indirection has been introduced, resulting in a certain degree of abstraction from the underlying network. This abstraction became a simple and efficient way of providing services without considering the intermediate devices. When service degradation occurs, the effective physical component or path used can become hidden. To identify the path the application traffic takes from endpoint to endpoint requires tremendous effort. The problem is exacerbated because VXLAN changes the UDP source-port to achieve entropy. In addition with ECMP the number of paths increases significantly, magnifying the problem scope.

VXLAN OAM (Operations, Administration, and Management) and CFM (Connectivity Fault Management) provide a simple and comprehensive solution for problems described earlier. Rather than trying to understand load-balancing hashing algorithms to figure out which path has been used, a single probe could provide the respective feedback, and acknowledge reachability of the expected destination. In the case where one path is affected by performance degradation, the probe could determine this by returning potential packet loss statistics. In the presence of an application or payload profile, the probe will mimic application behavior, and the result will plot the effective physical path used by the workloads in question.

An additional tool within VXLAN OAM is the “tissa”-based tracepath, following the “draft-tissa-nvo3-oam-fm” IETF draft. This tool not only gets the exact path plotting from an underlay perspective, but it also derives the specific VTEP where the destination is actually attached. Furthermore, with additional input parameters it is possible to identify the egress VTEP, the underlay path from ingress to egress VTEP including all intermediate hops, as well as all involved interfaces. In addition, the load and error counters for those interfaces can be provided as well.

The sample output below shows a "tissa"-based overlay pathtrace. The functionality exposes the physical path (underlay) from leaf via spine to border, while the request was initiated in the VXLAN overlay.

```
Path trace Request to peer ip 10.254.254.200 source ip 10.254.254.102
Sender handle: 38

Hop   Code   ReplyIP   IngressI/f  EgressI/f   State
=====================================================
  1 !Reply from 10.254.254.101, Eth2/1 Eth4/5 UP / UP
  Input Stats:
    discards:0
    errors:0
    unknown:0
    bandwidth:42949672970000000
  Output Stats:
    discards:0
```

```
    errors:0
    bandwidth:42949672970000000

  2 !Reply from 10.254.254.200, Eth6/1 - UP / -
  Input Stats:
    discards:0
    errors:0
    unknown:0
    bandwidth:42949672970000000
```

VXLAN OAM is implemented within NX-OS platforms and can be executed using CLI or with an API-driven approach using NX-API. It is possible to execute the various probes across a programmatic interface and also retrieve the statistical information from the VTEP in the same way. As an acknowledgment of the probe execution, a statistic identifier is sent. The statistic identifier provides current and historic statistics from the VTEP local OAM database in a programmatic or CLI driven way. Further enhancements in VXLAN OAM include the introduction of periodic probes and respective notification to more proactively manage the overlay network with its physical underlay. In order to make the collected path information and statistics meaningful, VXLAN OAM is going to integrate with VXLAN related management systems like DCNM or VTS.

Available Tools

There are multiple approaches to address the challenges described in the previous section. These approaches can be classified into the following categories:

- Traditional (CLI, scripting)
- Off-the-shelf tools
- DevOps (Puppet, Chef, Ansible)

Traditional Tools

Command Line Interface

For VXLAN there are a number of new commands to help with configuring, monitoring and troubleshooting the fabric. However, it needs to be noted that network operators who rely on CLI to manage their fabric will be confronted with two issues:

1. VXLAN configuration is command-intensive. The creation of new tenants or segments requires multiple lines of configuration, potentially across a large number of devices.
2. VXLAN technology depends on the presence of a considerable number of underlying protocols, making it more burdensome to deploy when compared to other technologies like Spanning Tree or FabricPath.

Python Scripting

Python scripting has been used by network operators for years; however, with NX-OS running on the switches, scripting can be taken to a whole new dimension. APIs and Software Development Kits (SDKs) are available for NX-OS. An example of an SDK for NX-OS is the nxtoolkit which is freely available for download: https://github.com/datacenter/nxtoolkit.

An example Python script for VXLAN is located at the following: https://github.com/erjosito/evpn_shell. This script is essentially an external CLI that can be used to create, delete, and view tenants, VNIs, and relevant configuration elements across all VTEPs in a VXLAN EVPN Fabric. This script makes use of infrastructure variables such as management IP addresses, credentials etc. and with a single command deploys all the required VXLAN EVPN configuration to create a tenant or a network inside of a tenant.

For more scripting examples, please check GitHub (https://github.com/datacenter) or the Cisco Developer Community for NXOS (https://opennxos.cisco.com).

Scripting with Other Programming Languages

Multiple scripting languages can make use of NX-OS APIs using HTTP, assuming they can parse JSON or XML strings, even if no SDK is available for that specific language.

There are two APIs available in NXOS:

- NX-API: HTTP-based API over which CLI commands are sent to the device. The outputs can be sent back in JSON or XML format.
- REST API: RESTful API leverages an object model. Being completely object-based this makes development of SDKs possible. Python SDKs are available in Github (see https://github.com/datacenter). Commands and configuration are sent using XML or JSON format, and command outputs are returned similarly.

Languages such as XML and JSON are used to structure commands and outputs and they eliminate the need to parse human-readable strings formatted in paragraphs and tables. String parsing is commonly used in scripting but has version dependencies. That puts a burden on lifecycle management for these automation scripts that have kept many organizations from using them.The APIs available in NXOS are an improvement over traditional scripting methods, and will improve the automation processes.

Off-the-Shelf Tools

Cisco Data Center Network Manager (DCNM)

Cisco DCNM is a general purpose Network Management Software (NMS) / Operational Support Software (OSS) product targeted at NX-OS networking equipment. It supports classical Spanning Tree deployments with or without Virtual Port Channels, FabricPath and VXLAN.

In the context of a VXLAN-based solution, DCNM can be utilized for the following purposes:

1. Firstly, to provide for the Fabric underlay configuration. DCNM has built-in Power On Auto Provisioning (POAP) support to deliver zero-touch auto-provisioning of the network devices that build the VXLAN Fabric.
2. Once the Fabric is up and running, DCNM can also be utilized for provisioning the VXLAN overlay configuration.
3. DCNM supports monitoring of the performance and utilization of the network switches, as well as fault management and syslog aggregation.
4. Managing the software running on the switches and performing software upgrades and downgrades.

This provisioning can be performed in a top-down (push) fashion, where DCNM tracks deployment events and simply pushes the required CLI config for the access port onto the switch.

Alternatively, a more dynamic mechanism is possible, where the leaf switches “pull” the configuration from the LDAP database of DCNM based on a specific event, such as a local attachment of an endpoint. A typical example of this more dynamic mechanism is the support on the VXLAN leaf nodes of a functionality called Virtual Machine Tracker Auto-Config (VM Tracker), which automatically provisions a specific tenant configuration. The commands required for provisioning the tenant are stored in the form of a configuration profile. A configuration profile is a set of commands that will be required

for provisioning a particular tenant, except the required parameters are written as variables instead of actual values in a command.

Specific to VXLAN management DCNM provides the following capabilities:

- DCNM provides integrated Power-On Auto Provisioning (POAP) to boot new switches for a greenfield Fabric or add new switches to an existing VXLAN Fabric. DCNM manages this POAP workflow so that an admin simply assigns a device to a preconfigured template.
- In addition, the POAP configuration Diff/Sync feature lets the admin know if a device's configuration does not match its POAP template and then lets the user resolve these differences.
- DCNM also presents topology views showing physical and overlay networks on the same page, helping network admins quickly identify the extent of virtual overlay networks on a Fabric.
- DCNM also presents smart topology views showing virtual port channels (vPCs) and virtual device contexts. In topology view, DCNM shows VXLAN Tunnel endpoint status as well as VXLAN search. DCNM shows VXLAN network identifier (VNI) status and other VXLAN information on a per-switch basis.
- Built-in search allows admins to search by VM Name, VM IP Address, VM MAC Address, VNI, or Switch ID.

More information on Cisco Data Center Network Manager can be found at: http://www.cisco.com/go/dcnm.

Ignite

Day-0 tasks are extremely important in order to have a consistent Fabric. Ignite is a simple hands-off approach to bootstrap a device with the appropriate code level and initial device setup. To achieve that, Ignite leverages the POAP capabilities of Cisco Nexus switches.

Ignite is an open-source tool that can be downloaded at no cost from Github: https://github.com/datacenter/ignite.

In order to have a POAP environment that allows for the automation of deployment of firmware and initial configuration, there are some external components that Ignite requires:

- A DHCP server to bootstrap the interface and DNS information of switches that are booting up.
- A TFTP server that contains the configuration script used to automate the software image installation and configuration process.
- An Ubuntu server where Ignite will be installed, that contains the desired software images and rules to dynamically build configuration files.

Cisco Nexus Fabric Manager

The Cisco Nexus Fabric Manager (NFM) is a management system designed to highly simplify and optimize the full lifecycle management of a switch fabric built with NX-OS based platforms (at the time of writing of this book, NFM support is limited to the Nexus 9000 family).

Cisco NFM has a fabric-wide focus and allows for the auto-provisioning and management of the whole network. NFM provides point-and-click methods for performing fabric management tasks such as adding, removing, and configuring network components such as switchpools, switches, switch interfaces, VRFs, port channels and broadcast domains.

Cisco NFM builds a VXLAN EVPN Fabric, but abstracts the complexity . It is still possible to log into the switches and view the configuration that has been deployed by Cisco NFM, troubleshoot with the CLI, or use any other standard monitoring solution to verify the state of the network.

Cisco NFM covers various phases of the Fabric management lifecycle:

- **Creation**: NFM allows for a zero-touch boot up of the Fabric, performing some Day-0 operations like cabling topology verification and automatic VXLAN underlay provisioning

- **Connection**: NFM fully manages the entire VXLAN configuration, removing the operational associated hurdles. This essentially implies that a user does not necessarily need to know that VXLAN with MP-BGP EVPN is deployed as the key functionality to enable endpoint communication
- **Expansion**: there are more day-N type of operations, such as zero-touch addition of switches to the Fabric and auto-upgrade of existing fabric devices
- **Fault Management**: NFM offers a built-in fault management system
- **Reporting**: Cisco NFM communicates to the switches deployed in the fabric by leveraging software agents embedded into the switches

More information regarding Cisco Nexus Fabric Manager is available at: http://www.cisco.com/go/nexusfabricmanager.

Cisco Virtual Topology System (VTS)

Service providers have very specific requirements regarding data center network management and operations:

1 Support for a mix of software and hardware VTEPs

2 Integration with the hypervisor layer

3 Support of a multivendor Fabric

4 Overlay and underlay operated by different teams

VTS is an add-on to a VXLAN Fabric consisting of the following elements:

- Virtual Topology Controller: this is a management platform that offers ways to deploy tenants and networks over a GUI or a northbound RESTful API. It integrates with VMware vCenter and with Openstack/KVM, so customers can manage the overlay directly from the VMM. The Virtual Topology Controller will roll out the required changes using southbound APIs such as NX-API or NetConf/YANG.

- IOS XRv: this is a virtual router instance that can take over required control plane functionality in case of a deployment consisting exclusively of software VTEPs. This component is responsible for distributing routes to hardware VTEPs over EVPN BGP, and to software VTEPs using the RESTCONF API.
- Virtual Topology Forwarder (VTF): this is a software VTEP that can be installed in a VMware vSphere host or an Openstack compute node. It is controlled by the Virtual Topology Controller, offering L2 and L3 connectivity between VMs running in the local or remote servers. VTF is a virtual machine running in user space, so it does not need any modification to the vSphere code. VTF exploits performance optimization technologies such as the open-source-licensed DPDK (http://dpdk.org/) and Cisco Vector Packet Processing (VPP).

Cisco VTS supports flood and learn as well MP-BGP EVPN control planes. It includes functionality such as ARP suppression capabilities, symmetric IRB, VTEP authentication and fast convergence upon network failures and endpoint mobility.

One important concept to understand is that Cisco Virtual Topology System does not manage the underlay. It is assumed that the required underlay configuration is already in place.

More information regarding Cisco Virtual Topology System is available at: https://www.cisco.com/go/vts.

DevOps Tools

Configuration Management Tools (CMT) are a new generation of intent-based tools that have gained great popularity, mainly in the Linux community. They can be classified into two categories: Agent-based and agentless tools.

- In agent-based configuration management, changes are made centrally on a master node, and are pulled down and executed by the agent. The device agents periodically connect with the master for configuration information and the changes are pulled down and executed. Only the changes that are needed are pulled.

- Agentless Configuration Management is push-based instead of pull-based. Configuration management scripts are run on the master and the master connects to the managed devices and executes the task over an API.

Puppet and Chef are examples of agent-based configuration management tools. With these agent-based systems, the user leverages a custom declarative language to describe the system configuration which needs to be configured on the remote systems. Both of these tools have similar functionality which is continually evolving. Puppet recently released modules to configure, provision, and manage a Cisco VXLAN-based Fabrics plus several standard top-of-rack switch features.

Puppet uses modules that include descriptions about which features are supported, and manifests that are the actual descriptions of how those devices should be configured. Manifests can be static, dynamically incorporate conditions or even use Ruby logic. Some conditions will depend on which system is being managed, and a wealth of that information is gathered by Puppet's companion tool "facter". The Puppet agent will pull the manifest from the Puppet server (Puppet Master) and implement it.

There are some examples manifests in Github under https://github.com/cisco/cisco-network-puppet-module.

Chef architecture is very similar, but instead of manifests the jargon is "recipes", that is where the expected state of the managed devices is documented. Recipes can be grouped together in Cookbooks for easier management. As already described, Chef runs in a client/server architecture, but it has an additional standalone mode called "Chef solo".

As with Puppet, some examples of Chef recipes for Cisco NX-OS are available in Github under https://github.com/cisco/cisco-network-chef-cookbook.

Ansible is an example of an agentless based configuration management system that manages nodes via SSH and has the ability to execute the scripts locally on the managed node or on the local server connects via the Cisco NX-API. Ansible uses the concept of Modules, Tasks, Plays, and Playbooks to manage the configuration on the remote devices.

- **Modules**: units of work that Ansible ships out to remote machines. Some modules pre-installed, custom modules can be manually installed as well
- **Tasks**: combination of modules with arguments and description names
- **Plays**: mapping of hosts or groups to their tasks
- **Playbooks**: collection of Plays by which Ansible orchestrates, configures, administers, or deploys systems. Playbooks are written in YAML

Summary Table

The following table illustrates how the tools discussed above contribute to the Day 0, 1 or N operations of network fabrics:

	Day0	Day1	DayN
CLI		X	X
Python		X	X
Cisco Data Center Network Manager	X	X	X
Cisco Nexus Fabric Manager	X	X	X
Ignite	X		
Cisco Virtual Topology System		X	X
Ansible		X	X
Puppet		X	X
Chef		X	X

Acronyms

Acronyms

ACI: Application Centric Infrastructure

ADC: Application Delivery Controllers

API: Application Program Interface

ARP: Address Resolution Protocol

BD: Bridge Domain

BGP: Border Gateway Protocol

CLI: Command-Line Interface

DAG: Distributed Anycast Gateway

DCNM: Data Center Network Manager

ECMP: Equal Cost Multi-Path

ETR: Egress Tunnel Router

EVPN: Ethernet Virtual Private Network

FCAPS: Fault, Configuration, Accounting, Performance and Security

GENEVE: Generic Network Virtualization Encapsulation

GPE: Generic Protocol Encapsulation

IDS: Intrusion Detection System

IEEE: Institute of Electrical and Electronics Engineers

IGP: Interior Gateway Protocol

IPS: Intrusion Prevention System

IRB: Integrated Routing and Bridging

ITR: Ingress Tunnel Router

LISP: Locator/ID Separation Protocol

LSA: Link State Advertisement

MP-BGP: Multi-Protocol BGP

MPLS: Multi-Protocol Label Switching

MSDP: Multicast Source Discovery Protocol

MTU: Maximum Transmission Unit

NAT: Network Address Translation

NDB: Nexus Data Broker

NFM: Nexus Fabric Manager

NLRI: Network Layer Reachability Information

NSH: Network Service Header

NVO: Network Virtualization Overlay

OAM: Operations, Administration and Management

OTV: Overlay Transport Virtualization

PBB: Provider Backbone Bridges

PIM: Protocol-Independent Multicast

POD: Point of Delivery

PVP: Path Vector Protocol

RD: Route Distinguisher

RP: Rendezvous Point

RR: Route Reflector

RT: Route Target

SDK: Software Development Kit

SDN: Software Defined Networking

SNMP: Simple Network Management Protocol

VMM: Virtual Machine Manager

VNI: Virtual Network Instance

VNID: VXLAN Network Identifier

vPC: Virtual Port-Channel

VRF: Virtual Routing and Forwarding

VTC: Virtual Topology Controller

VTEP: Virtual Tunnel Endpoint

VTF: Virtual Topology Forwarder

VTS: Virtual Topology System