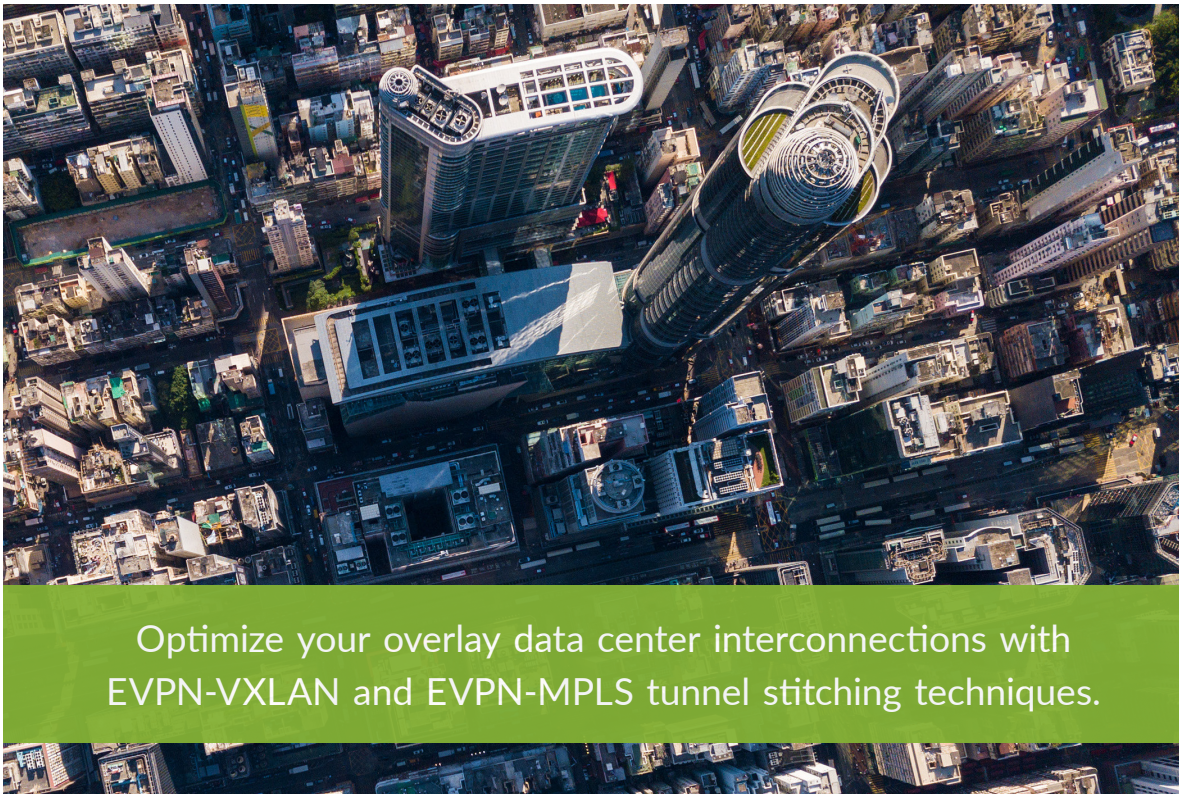
JUNIPER NETWORKS

DAY ONE: SEAMLESS EVPN-VXLAN TUNNEL STITCHING FOR DC AND DCI NETWORK OVERLAY



Optimize your overlay data center interconnections with EVPN-VXLAN and EVPN-MPLS tunnel stitching techniques.

By Elisabeth Rodrigues, Michal Styszynski, Kishore Tiruveedhula

DAY ONE: SEAMLESS EVPN-VXLAN TUNNEL STITCHING FOR DC AND DCI NETWORK OVERLAY

The various data center interconnect options covered in this book show how flexible the EVPN control plane has become over the last couple of years. It proves that it can be used in greenfield scenarios, stitching VXLAN-to-VXLAN, as well as brownfield where sometimes VXLAN-to-MPLS stitching is a better choice to quickly interconnect remote data centers, as well as larger scale POP locations or remote campuses sites, allowing them to access data from different data center locations.

This book covers six different DCI options with a larger focus on EVPN-VXLAN-to-EVPN-VXLAN for L2 bridging/switching purposes, and EVPN-VXLAN-to-EVPN-VXLAN for L3 IP prefix advertisement purposes. While the MPLS DCI is still very popular, the authors contend that the VXLAN-to-VXLAN option fulfills most site requirements, opening it to additional emerging use cases where Group Based Policy (GBP) profiles are also extended between DC sites for micro-segmentation purposes.

Day One: Seamless EVPN-VXLAN Tunnel Stitching for DC and DCI Network Overlay is a thorough examination of Junos tunnel stitching techniques for data centers by some of the leading experts at Juniper Networks.

"The data center space is hot with emerging technology. Indeed, data center design has fundamentally changed over the last several years as new technology has brought new capabilities together with better operational design. But navigating new technology can be difficult. How do you know which designs to use and how to avoid pitfalls of working in only abstract ideas? This book introduces various options for handling DCI, providing step-by-step instruction on how to decide, design, deploy, and manage. This is a difference-maker for network architects, engineers, and operators."

Michael Bushong, GVP Cloud-Ready Data Center, Juniper Networks

IT'S DAY ONE AND YOU HAVE A JOB TO DO:

- Understand the new DCI overlay options
- Configure seamless stitching for different forms of encapsulation
- Learn advanced EVPN topics related to the fabric deployment on different data center sites
- Deploy EVPN-VXLAN to EVPN-VXLAN and EVPN-VXLAN to EVPN-MPLS seamless stitching
- Verify the operation of EVPN-VXLAN fabrics and DCI

Juniper Networks Books are focused on network reliability and efficiency. Peruse the complete library at www.juniper.net/dayone.

JUNIPER NETWORKS

Day One: Seamless EVPN-VXLAN Tunnel Stitching for DC and DCI Network Overlay

By Elisabeth Rodrigues, Michal Styszynski, and Kishore Tiruveedhula

JUNIPER NETWORKS

Published by Juniper Networks Books
Authors: Elisabeth Rodrigues, Michal Styszynski, and Kishore Tiruveedhula
Reviewers: Sean Clarke, Adrien Desportes
Editor in Chief: Patrick Ames

Version History: v1, February 2023
2 3 4 5 6 7 8 9 10

About the Authors

Elisabeth Rodrigues is a technical solution consultant specialized in data center networking. She supports the sales teams in their data center projects by providing guidance in the design of solutions, delivering technical deep-dive presentations and demonstrations to customers, and leading Proof-of-Concept tests. She has been working for Juniper for 8 years and has been a Junos fan for more than 15 years.

Michal Styszynski is a Sr Product Manager in CRDC - Cloud Ready Data Center business unit at Juniper Networks, focusing on Junos and Junos Evolved for the QFX product line. Michal joined Juniper Networks over 10 years ago. Before his current PLM role, he also worked in technical marketing and product consulting, focusing on data centers, storage networking projects for major telcos and large enterprises. Before Juniper, he worked for about 10 years at Orange/FT R&D and TPSA. Michal graduated from the Electronics & Telecommunication faculty at Wroclaw University of Science & Technology. He's JNCIE-DC#523 and PEC, PLC, PMC certified from the Product School, San Francisco, California.

Kishore Tiruveedhula is a Senior Staff Engineer with Juniper Networks. He has over 20 years of experience designing and implementing routing protocols features in MPLS and VPN technologies. And most recently developed and implemented multiple EVPN DCI features like seamless interconnection of EVPN-VXLAN to EVPN-VXLAN, EVPN-VXLAN to EVPN-MPLS and DCI type 5 stitching, which are covered in this *Day One* book.

Authors' Acknowledgments

Elisabeth Rodrigues thanks her beloved partner, Sylvain, and her children, Rachel and Ruben, for their patience and love. A special thanks to Michal Styszynski for accepting her contribution to this book. A big thank you to her managers and directors, Dirk van den Borne, Selmane B. Slama, and Tom Ruban for their support.

Michal Styszynski would like to thank his wife Kasia, sons Ernest and Marcel for their love, patience, and understanding. He thanks the support when writing this book to his current managers Michael Bushong, Subramaniam Vinod, and Mahesh Subramaniam, as well as his former managers, Tim McCarthy and Praful Lalchandani.

Kishore Tiruveedhula would like to thank Michal Styszynski for starting this great initiative *Day One* book and the opportunity to contribute to this book. He would like to thank Wen Lin, Distinguished Engineer, and Selvakumar Sivaraj, Distinguished Engineer, for their valuable suggestions during implementing the DCI features. He thanks managers Vrishab Sikand, Raveendra Torvi, and Manish Gupta for giving him the opportunity to work on implementing these EVPN DCI features.

Everything shared in this book was possible thanks to the hard work of Juniper's software development and test engineering colleagues. Special thanks go to Wen Lin and Selvakumar Sivaraj from the Distinguished Engineering team for their leadership in architecting EVPN solutions. Appreciations also go to our Editor in Chief, Patrick Ames, reviewers Sean Clarke, Sr. Manager for Worldwide Proof-of-Concept, and Adrien Desportes, Sales Engineering Manager at Juniper Networks.

Welcome to Day One

This book is part of the *Day One* library, produced and published by Juniper Networks Books. *Day One* books feature Juniper Networks technology with straightforward explanations, step-by-step instructions, and practical examples that are easy to follow.

- Download a free PDF edition at https://www.juniper.net/dayone.
- Purchase the paper edition at Vervante Corporation (www.vervante.com).

Key EVPN-VXLAN Resources

The authors of this book highly recommend the following resources:

- Requirements for Ethernet VPN (EVPN): https://datatracker.ietf.org/doc/rfc7209/
- Problem Statement: Overlays for Network Virtualization: https://datatracker.ietf.org/doc/rfc7364/
- Framework for Data Center (DC) Network Virtualization: https://datatracker.ietf.org/doc/rfc7365/
- BGP MPLS-Based Ethernet VPN: https://datatracker.ietf.org/doc/pdf/rfc7432/
- Use of BGP for Routing in Large-Scale Data Centers: https://datatracker.ietf.org/doc/rfc7938/
- A Network Virtualization Solution using Ethernet VPN (EVPN): https://datatracker.ietf.org/doc/rfc8365/
- Interconnect Solution for Ethernet VPN (EVPN) Overlay Networks: https://datatracker.ietf.org/doc/rfc9014/
- Integrated Routing and Bridging in Ethernet VPN (EVPN) https://datatracker.ietf.org/doc/rfc9135/
- IP Prefix advertisement in Ethernet VPN: https://datatracker.ietf.org/doc/rfc9136/

What You Need to Know Before Reading This Book

Before reading this book, you need to be familiar with the basic administrative functions of the Junos operating system, including the ability to work with operational commands and to read, understand, and change Junos configurations. There are several books in the Day One Fundamentals Series on learning the Junos OS, at http://www.juniper.net/dayone.

This book makes a few assumptions about you, the reader:

- You are familiar with and versed in using the Junos CLI for switch and router configuration
- You are an architect of the data center LAN or DCI network
- You are the operational team leader in DC production network
- You have a basic understanding of IP routing and switching
- You can build out the lab topologies used in this book without detailed instructions.

What You Will Be Able To Do After Reading This Book

- Understand the new DCI overlay options
- Configure seamless stitching for different form of encapsulations
- Learn advanced EVPN topics related to the fabric deployment on different data center sites
- Deploy EVPN-VXLAN to EVPN-VXLAN and EVPN-VXLAN to EVPN-MPLS seamless stitching
- Verify the operation of EVPN-VXLAN fabrics and DCI.

Seamless Stitching Platform Support

To check platform and release support for the seamless stitching features documented in this Day One book, see the *Seamless EVPN-VXLAN stitching* feature in Feature Explorer.

Preface

Secure and 'always-on' access to the data was important for the last 20 years but with the global pandemic outbreak, many organizations realized even more about the significance of getting access to the most relevant and up to date information.

The critical part in getting information on time is the data center ecosystem, where massive files are stored and where servers offer ultra-fast data processing. Interconnecting data center networks located in different geographic regions is likewise important to ensure the data gets replicated between different regions and access to the information is still offered in case of natural disaster (wildfires, floods), power outages or persistent security attack at one of the data center network locations. For many organizations, 24/7 secure access to the data through private data center interconnect investments is representing an important competitive advantage.

To ensure multi-site data center interconnect is delivered in a simple, secure, and agile way, the use of network virtualization overlay (NVO) techniques recently became more popular.

This book focuses on how to unify and optimize the overlay data center interconnections with EVPN-VXLAN and EVPN-MPLS tunnel stitching techniques. We'll walk through the main use cases and architectures where the NVO (Network Virtualization Overlay) seamless tunnel stitching technique can become even more relevant. The implementation and verification tasks using prescriptive DC and DCI designs should help you to understand precisely how to use it in your production networks

Glossary

ARP = Address Resolution Protocol

aka = also known as

AS = Autonomous System

ASN = Autonomous System Number

BGP = Border Gateway Protocol

BO = Bridged Overlay

CE = Customer Edge

CFM = Connectivity Fault Management

CLI = Command Line Interface

CRB = Centrally-Routed Bridging

DC = Datacenter

DCI = Datacenter Interconnect

ECMP = Equal-cost multipath

ERB = Edge-Routed Bridging

ESI = Ethernet Segment Identifier

EVI = EVPN Instance

EVPN = Ethernet Virtual Private Network

GBP = Group Based Policy

GW = Gateway

IANA = Internet Assigned Numbers Authority

IP = Internet Protocol

IRB = Integrated Routing and Bridging

IFL = Logical Interface

IGP = Interior Gateway Protocol

IS-IS = Intermediate system to intermediate system

iRT = Interconnect Route Target

iRD = Interconnect Route Distinguisher

iESI = Interconnect ESI (EVPN Segment Identifier)

LACP = Link Aggregation Control Protocol

LAG = link aggregation group

LAN = Local Area Network

MAC = Media Access Control

MAN = Metropolitan Area Network

MC-LAG = Multi Chassis LAG

NLRI = Network Layer Reachability Information

MPLS = Multiprotocol Label Switching

NVO = Network Virtualization Overlay

OAM = Operations Administration and Maintenance

OSPF = Open Shortest Path First

PE = Provider Edge

PoD = Point of Delivery

RFC = Request for Comments

RT – Route Target

RD – Route Distinguisher

SMET = Selective Multicast Ethernet Tag

TAP = Test Access Point

TCAM = Ternary Content Addressable Memory

UDP = User Datagram Protocol

VGA = Virtual Gateway Address

VM = Virtual Machine

VNI = VXLAN network identifier

VPLS = Virtual Private LAN Service

VPN = Virtual Private Network

VRF = Virtual routing and forwarding

VTEP = Virtual Tunnel End Point

VXLAN = Virtual Extensible LAN

WAN = Wide Area Network

Chapter 1

Data Center Overlay and EVPN

NVOs (Network Virtualization Overlays) have become popular in the DC and DCI network infrastructure context mainly because they are offering design agility for quicker, simpler service delivery, meaning efficiently crossing the underlay network boundaries with limited requirements for the underlay IP routing. The overlay's distributed state is typically offering a much higher scale, better growth over time than the traditional 3-tier LAN DC network design with reduced failure domain.

Using the overlay networking approach helps build unified LAN DC and DCI ecosystems with the common BGP EVPN overlay signalization protocols in both domains while offering various tunneling transport options for the data center interconnect.

With the newer EVPN-VXLAN overlay techniques, interconnecting data centers became much easier and faster to implement because it offers less dependency on the core-IP capabilities. For example, when comparing the EVPN-VXLAN to traditional VPLS-MPLS L2 networking, we can highlight that the newer overlay technology is not reserved anymore to telco's and larger enterprises, and now small to medium enterprises can use it directly at the DC fabric level, without additional CAPEX/OPEX spendings on dedicated gateways/licenses or without asking the WAN teams to deliver such connectivity for services extension and high availability.

Before focusing on our main topic of this book - the DCI – and to better understand the DC interconnect part, let's review the characteristics, requirements, and outcomes of the modern DC fabric where EVPN-VXLAN is used as a main technology for Layer 2 and Layer 3.

Network Virtualization and Multi-tenancy

Modern data centers are highly scalable and distributed systems that offer multi-tenancy and rich services at the servers and network node levels.

Network virtualization capabilities are popular in the data centers because they deliver additional security capabilities (IP and MAC level isolation) and node-level multi-tenancy.

Tenants are highly dynamic (in terms of creation/deletion/mobility) and have strong security and privacy requirements. They can use pure IP routing, pure bridging, or a combination of bridged and routed network services at the fabric level. Some tenants shouldn't be able to communicate with each other except on a strictly controlled manner (for example through a firewall).

In the case of the overlay IP Clos fabrics, they are typically only enabled at the edge of the fabric – at the server leaf and border-leaf level. In the newer overlay fabric design, the server leaf switch and border-leaf are handling all the network virtualization

Tenants operate in one or many virtual network instances. Virtual networks are distributed and are spread across data centers. They offer fully distributed architecture model instead of traditionally used parallel multi-VRF, tenant per tenant extensions to the aggregation layer.

In the case of the newer leaf-based overlay virtualization techniques, you can benefit from simpler, faster delivery of the services provisioning.

The intermediate spine to leaf underlay IP links, as well as the spine devices, are provisioned during the installation of the IP Clos and are not changed when a new tenant service is delivered. The changes are just done at the leaf level of the fabric, where the server/customer physically connects.

This new architecture makes data centers incrementally deployable without requiring major upgrade of the entire network.

Requirements of Data Center Overlay Networks

New applications are also driving new requirements for L2 and L3 network services. RFC7209, and RFC7365 provide a list of requirements and definitions for overlay networks in the data center.

The main requirements of a multi-tenant data center are:

- Isolation of network traffic per tenant
- Support for large number of tenants
- Extension of L2 connectivity across different PoDs within a DC or between different DCs

- Active/Active forwarding
- VM server mobility

Some of the requirements mentioned above are not completely new and were around in the DC networking industry for a long time. The way these requirements are fulfilled with overlay networks is solved in the protocol design and is done automatically within BGP EVPN signalization, instead of a monolithic spanning-tree combined with multi-VRF access/aggregation/ core designs.

The following list of the EVPN feature set is needed to ensure all requirements can be combined under the same overlay network in the data center.

Network Service Types

Network virtualization at the server leaf level (ERB – Edge Routed Bridging) can provide L2 Ethernet LAN-like service, L3 IP/VRF-like service, or both services for a tenant.

New L2 services can be enabled depending on end user requirements: VLAN-aware, VLAN-based and VLANbundle. Those services can coexist on the same server-leaf or border-leaf and enabled in specific MAC-VRF EVPN routing instances – aka EVIs.

L3 service must be distributed at the leaf level as routing performed by centralized L3 gateway devices is sometimes less efficient – mainly due to leaf-to-spine bandwidth utilization or due to the larger blast radius/failure domain.

Multi-homing

All-active redundancy mode is a strong requirement in data center networks, and it must use an algorithm that ensures that the frames are delivered in order for a given traffic flow.

Traffic forwarding must be optimized for a multi-homed group. It should not be forwarded between PE devices that are members of a multi-homed group unless the destination CE is attached to one of the multi-homed PEs.

Figure 1.1 is an example topology where leaf L2 is a member of two different ESI-LAGs with two different leaves – one ESI towards server1 and the other ESI to server2.

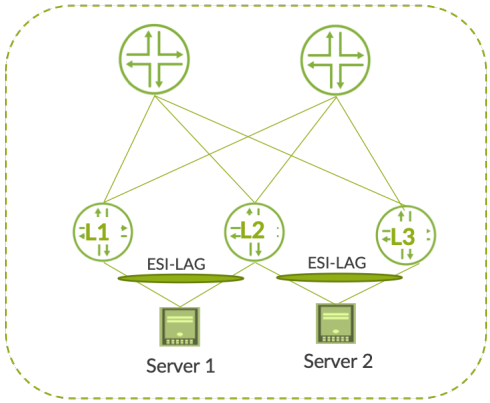


Figure 1.1 *Basic L2 multihoming scenario using ESI-LAG*

In this scenario the local bias will be acting for the flows sent to leaf2 and destined to server1 or server2. All ECMP paths to remote leaf nodes are typically used from leaf to spines but also from each spine to destination leaf nodes – when the destination MAC address is on a multihomed server connected via an ESI-LAG.

Ease of Provisioning

The PEs leaf nodes belonging to a given VPN or a given redundancy multi-homed group must be auto discovered. Different ESI types were defined by IETF and can help delivering the L2 multi-homing in much easier way, for example the ESI type-1 (not to confuse with EVPN route-type 1) generates the ESI 10 bytes value based on the LACP system-id information received from the server or the CE switch. Same goes for the ESI type-3 which can build the ESI value automatically based on the system MAC@ and local discriminator.

Otherwise, the ease of provisioning is brought by fabric managers such as Apstra or via the auto-EVPN options delivered across the fabric using the RIFT protocol as an underlay.

Control Plane MAC Learning

EVPN brings the control-plane BGP based Ethernet MAC learning instead of the ethernet flood-based L2 address learning. MAC learning makes MAC address mobility and consequently, VM Mobility, very efficient. VMs can move from one server to another server or across DCs in a seamless manner.

Control plane MAC learning brings other key advantages like fast convergence and ARP suppression.

Fast convergence

Virtualized applications increase the volume of MAC addresses. The network convergence time upon failure must be independent of the number of MAC addresses learned by the PE and it should be independent of the number of service instances.

Flood suppression

The data center network minimizes the amount of flooding of multi-destination frames and eliminates unnecessary flooding upon topology changes, especially in the case of a multi-homed site.

Multicast optimizations

By default, when ingress replication is in use, multicast traffic is flooded to remote PEs of a given bridge domain.

To optimize multicast L2 traffic in each bridge domain, PEs (leaf nodes) can act as IGMP/MLD proxies to make sure multicast traffic is only sent to interested receiver hosts in an efficient way. Each time a host is interested in a multicast group, it sends an IGMP/MLD Membership Report. Then, the IGMP/MLD router periodically sends Membership Queries.

The goal of IGMP / MLD proxy solution in EVPN is to reduce the flooding of IGMP/MLD messages and act as a distributed multicast router forwarding multicast traffic only to PEs that have hosts interested in the multicast group.

To achieve IGMP/MLD proxy, three new route types are described in RFC9251:

- RT6: Selective Multicast Ethernet Tag route (SMET) ensures that each PE sends its local IGMP membership requests and leave group state. Instead of flooding the IGMPv2 membership report coming from the receiver server into the fabric, it's transformed into the EVPN Type-6 route to advertise the intent to receive the given multicast feed.
- RT7 and RT8: Multicast Membership Report Sync route / Multicast Leave Sync route are used to coordinated IGMP states in the case of multi-homing where PEs share a given Ethernet Segment.

When the fabric is using distributed IP anycast gateways the edge-routed multicast is also sometimes required to preserve the leaf to spine bandwidth or reduce the latency when sources and receivers are connected to the same node. This is something EVPN fabrics are typically delivering through the OISM (Optimized Inter-Subnet Multicast) at the leaf and border-leaf level.

Security Considerations

With an EVPN control plane, the network can detect and properly handle duplicated MAC addresses and it can associate a MAC address with specific flags. For example, when the EVPN MAC address is detected as a duplicate (for example because of an Ethernet loop, introduced by wrong leaf/server cabling) it can be set as a blackhole MAC@ and discard the packets destined or sourced at that address.

Alternatively, in some cases the CFM based enhanced loop-detection can be used to react quicker to the loop and simply disable the interfaces involved in the loop.

Other important security features can be added to data center networks. For example, MAC Sec encryption on all leaf/spine links, L2-L3 traffic isolation, stateless ACLs, service chaining to a L4-L7 firewall, or the VXLAN Group Based Policy to deliver the micro segmentation.

NOTE This book will also examine additional security requirements in its scope of DCI.

EVPN: The Data Center Overlay Network

EVPN answers all data center overlay requirements previously described. The beauty of EVPN resides in the fact that with a single control plane you can offer L2 and L3 services as well as multi-homing, scaling, and convergence requirements.

NOTE EVPN is constantly evolving with the addition of new capabilities, this chapter focuses mainly on RFC7432, RFC8365, RFC9135, RFC9136.

EVPN Main Characteristics

Here are the main features offered by EVPN:

- Control-plane information is distributed with BGP EVPN is a new addition to BGP and benefits from all BGP capabilities in terms of auto-discovery, flexibility, and scaling. MAC addresses as well as IP prefixes can be advertised using the EVPN control plane.
- BGP auto-discovery is used to discover PE devices participating in a given VPN, and in a given redundancy group to discover the tunnel encapsulation type, the multicast tunnel type, and members.
- Broadcast and multicast traffic is sent using a shared multicast tree or with ingress replication.
- All-active multi-homing for ESI-LAG and DC-GW multi-homing is an important EVPN feature. With ESI-LAG, it allows a given CE to use all links connected to

multiple PEs and, at the DC-GW level, it allows a given leaf node to load-balance the traffic to multiple border-leaf nodes.

- Fast convergence is offered using MAC mass withdrawal in case of PE-CE link failure. A single route is used to notify remote PEs.
- VM Mobility mechanism is tracked with sequence numbers.
- The use of Route Targets allows different logical topologies (mesh, hub, spoke and extranet).
- Ethernet loop-detection is built in EVPN control plane.

EVPN and Different Encapsulation Options

BGP EVPN can be deployed with different encapsulation types: VXLAN, NVGRE, MPLS, MPLSoGRE, GENEVE.

In this book, we are focusing on VXLAN and MPLS as encapsulation options for EVPN. The MPLS encapsulation is considered for the DCI use case of EVPN while the VXLAN is considered inside the data center fabric as well as for the DCI use cases.

VXLAN has the advantage of using a UDP header which means it can be used to cross non-VXLAN-aware devices. For example, in the context of DCI, VXLAN packets could cross any IP network or Internet. Also, VXLAN doesn't require the use of Spanning Tree Protocol to avoid loops and it can provide much better availability and bandwidth with the use of multipathing.

Furthermore, the use of VLANs with their 12 bit VLAN ID, limits the number of broadcast domains to 4094. With the growing adoption of virtualization, this upper limit can be a challenge and VXLAN offers much better scalability.

Each VXLAN segment is identified through a 24-bit segment ID called VXLAN Network Identifier (VNI) allowing up to 16 million VXLAN segments. In the DC use-case each VLAN is mapped to a unique VNI so in practice the 4K VLAN-VNIs is used at the server-leaf level. For our scaled L2 requirement where same node must deliver more than 4K VLANs and when there's a need to deliver VLAN-overlapping, then there's a way to map all 4k * VLANs to a single VNI when using specific type of EVPN service – VLAN-bundle or when running the VLAN-rewrite operations ingress at the server connected port.

Here is the VXLAN packet as defined in RFC 7348 which is automatically set up on leaf and border-leaf nodes based on EVPN signalization. In the implementation chapters of this book (5-7), we share the pcap output related to the encapsulation used before and after the tunnel stitching.

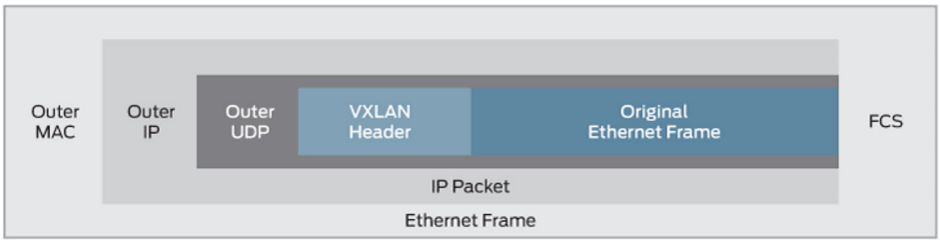


Figure1.2 *VXLAN Header used to transport data inside the DC and in DCI*

The VXLAN Header:

RRRRIRRR	Reserved	
VXLAN Network Identifier (VNI)		Reserved

This is an 8-byte field that has:

Flags (8 bits): where the I flag MUST be set to 1 for a valid VXLAN Network ID (VNI). The other 7 bits (designated "R") are reserved fields and MUST be set to zero on transmission and ignored on receipt.

VXLAN Segment ID/VXLAN Network Identifier (VNI): this is a 24-bit value used to designate the individual VXLAN overlay network on which the communicating VMs are situated. VMs in different VXLAN overlay networks cannot communicate with each other.

Reserved fields (24 bits and 8 bits): MUST be set to zero on transmission and ignored on receipt.

In the configuration and verification sections there'll be also some pcap wireshark outputs showing precisely how the VXLAN encap format looks like when leaving the server leaf and border-leaf.

Microsegmentation inside the EVPN-VXLAN fabric

The micro segmentation inside the data center fabric may help to additionally protect the workload by isolating them within the same VLAN-VNI into smaller group of workloads. The draft-smith-VXLAN-group-policy defines VXLAN Group Policy Option that allows a tenant system interface group identifier to be carried for the purposes of policy enforcement.

When it comes to the data center interconnect use case, the Group Based Policy information may be leveraged through automatic copy or rewrite. The data center gateway can copy the GBP information to the BGP control plane, setting it as a special community. For example, this is usefull when the DCI is using MPLS and where the information on GBP (specific to VXLAN transport) would be otherwise lost.

Outer UDP Header

This is the outer UDP header with a source port provided by the VTEP and the destination port being 4789 as assigned by IANA.

It is recommended that the UDP source port number is calculated using a hash of fields from the inner packet to enable a level of entropy for the ECMP/load-balancing of the VM-to-VM traffic across the VXLAN overlay. It should be in the dynamic/private port range 49152-65535.

Outer IP Header

This is the outer IP header with the source and destination IP addresses indicating the IP address of the VTEPs.

Inner VLAN Tag Handling

The VTEP by default SHOULD strip the VLAN tag unless configured otherwise, although in some cases the customer VLAN or the provider VLAN needs to be preserved, when EVPN-VXLAN is used in combination with Q-in-Q. For example, in the VLAN-BUNDLE service case, as described below, the VLAN tag remains.

L2 EVPN Service Types and MAC-VRFs

The EVPN standard describes several MAC-VRF service types to bring even more flexibility when provisioning Layer 2 services inside the data center fabric.

Here are the main characteristics of the MAC-VRFs also known as EVIs (EVPN Instances):

- They consist of one or more bridge tables
- They are identified by their corresponding Route Target (common between fabric nodes sharing same MAC-VRF) and Route Distinguisher (unique per each instance and per each node)
- Can enable different service-types for Layer 2 isolation and virtualization

Here are the three principal L2 service types offered in Junos and Junos Evolved as part of the MAC-VRF implementation – VLAN-based, VLAN-bundle, and VLAN-aware. Some of the main characteristics of these services are covered in the next section.

VLAN-based service-type

VLAN based service-type is an EVPN instance that consists of a single VLAN broadcast domain and single bridge-table per EVI/MAC-VRF. It is a one-to-one mapping between a VLAN-id: VNI and MAC-VRF/EVI – one VLAN mapped to one VNI inside the given MAC-VRF instance name.

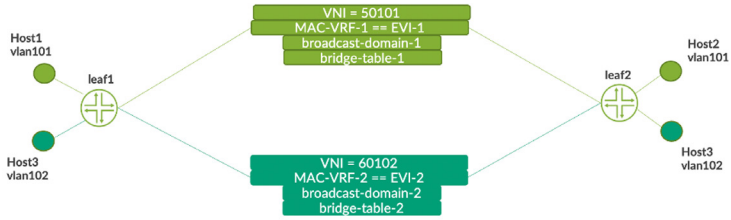


Figure1.3 EVPN VLAN-Based Service-type - Enabled at the MAC-VRF Level

This service is sometimes preferred but also means more auto-discovery routes in the fabric so higher RIB utilization when all VLAN-VNIs are enabled at all server leaf nodes. Nevertheless this option is sometimes used to offer complete isolation of two workloads – something we highlight in the network virtualization chapter.

VLAN Bundle service-type

A VLAN Bundle service-type is an EVPN instance that includes several VLAN broadcast domains sharing a single bridge table. MAC addresses must be unique across all VLANs for that EVI. It is a many-to-one mapping between VLANs and a MAC-VRF. In this case the VLANs coming from the servers of customers are typically not processed at the ingress port and are encapsulated into the VXLAN tunnel. This option helps to overlap the VLAN-ids but also to deliver higher VLAN scale of the data center fabric as even 4K VLANs can be mapped to the same VXLAN VNI.

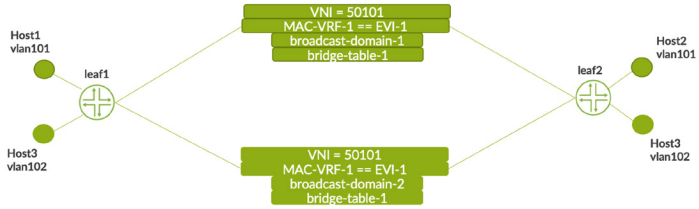


Figure1.4 VLAN Bundle Service-type

VLAN Aware service-type

A VLAN Aware service-type is an EVPN instance that includes several VLAN broadcast domains with each VLAN having its own bridge table and each VLAN mapped to dedicated VNI under the given MAC-VRF (EVI). Multiple bridge tables are maintained by a single MAC-VRF.

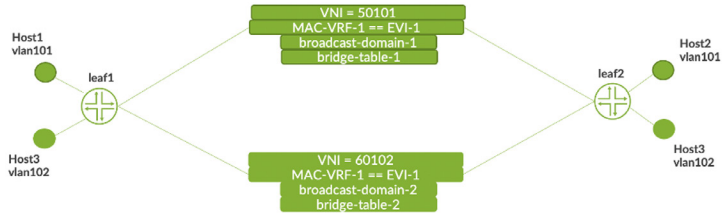


Figure1.5 VLAN Aware Service-type

This option is the most popular in enterprises that just want to define one Layer 2 MAC-VRF/EVI and add more VLANs later into it if needed.

It's also slightly more efficient from the RIB point of view when all VLAN-VNIs are enabled everywhere because we reduce the number of unique Auto-Discovery routes that are within the fabric.

BGP EVPN Ethernet Segments

Each Ethernet segment needs a unique identifier (ESI) in an EVPN.

ESI is encoded as a 10-octet integer and advertised as part of some EVPN route types when the servers/end hosts are multi-homed to two or more leaf nodes or when the border-leaf nodes are enabled with EVPN-VXLAN seamless stitching.

When a server is single-homed (server attached to one leaf node), the ESI value is 0 by default. An Ethernet segment multi-homed must have a unique value network wide. For a given server connected to two or more leaf nodes, the same ESI value is provisioned at each leaf. The same goes for the border-leaf enabled with the seamless stitching function on the given data center site – the same ESI value is provisioned on both border-leaf nodes, however, for the border-leaf nodes in another DC site, a specific different ESI value will be provisioned also on both borders.

There are five ESI types (not to confuse with EVPN route types) and the most popular are:

- Type 0: an arbitrary nine-octet ESI value configured by the operator. For example, the value could help identify the location of the ESI by having an octet that represents the DC, another one that represents the rack, then the interface number. This type of ESI is probably the most popular as of today as it offers the network admin full control over the segment identification
- Type 1: when LACP is used between PEs and CEs, this ESI type is a value auto generated from LACP parameters
- Type 3: value auto generated from the PE MAC address

BGP EVPN Routes

The EVPN standard is introducing different route-types to deliver unified control plane information within the same BGP address family and to optimize Ethernet and IP forwarding within the data center fabric as well as between the DCs. Depending on the way the services are deployed at the leaf nodes, some route types may not be used in a particular scenario. For example, if the DC fabric is just used for Layer 2 Ethernet extension purposes and tenant L2 virtualization, then the route Type-5 used for prefix advertisement is not going to be used. Another example is for the SMET route type-6 or Join/Leave sync route-type 7/8, they may not be used if there's multicasting needed in the fabric.

There are, however, certain route types which will always be used in the data center use case. For example, to build the baseline of the fabric the AD routes or IM routes will be used and the MAC/MAC-IP route type-2 will be always used when the given Layer 2 VLAN-VNI is extended between two leaf nodes.

The EVPN NLRI is carried in BGP using BGP Multiprotocol Extensions with an Address Family Identifier (AFI) of 25 (L2VPN) and a Subsequent Address Family Identifier (SAFI) of 70 (EVPN).

The following table summarizes different BGP EVPN route-types used in the present paper:

Table 1-1 BGP EVPN Route Types

Route Type	Route name	Usage
1	Ethernet AD (Auto Discovery) per ESI	MAC mass-withdraw, Split-Horizon using IP@ (for VXLAN encap.) Split-horizon label (for MPLS encap.)
1	Ethernet AD per EVI	Use for load balancing purposes.
2	MAC route	Advertising the MAC address using BGP
2	MAC-IP route	Advertising the MAC-IP binding (used for ARP/ND proxy/ suppression and for symmetric inter-irb routing
3	IM route – Inclusive Multicast route	Multicast Tunnel Endpoint Discovery, used for any ingress replication per VLAN-VNI
4	ES route – Ethernet Segment Route	Used to advertise the ESI 10 bytes info between the two or more nodes connected to same ESI and to elect the DF/nDF for the given ESI
5	IP Prefix Route	IP subnet advertisement using EVPN

6	SMET route	Selective Multicast Ethernet Tag. leaf node selectively sends or receives traffic based on the presence or absence of active receivers. IGMPv2/v3 or MLDv2 received from end host with the info on the intent to receive the multicast traffic is translated to BGP EVPN SMET route.
7	Multicast Join sync route	Used in multihoming (MH) scenario when receiver is connected to two leaf nodes and the nDF receives the IGMPv2 membership report then the Join Sync EVPN route type-7 is sent to the DF node elected for the given ESI so that he originates also the SMET route for the given multicast group.
8	Multicast Leave sync route	Used in multihoming (MH) scenario when receiver is connected to two leaf nodes and the nDF receives the IGMPv2 leave message then the Leave Sync EVPN route type-8 is sent to the DF node elected for the given ESI

RFC7432 describes the first four BGP EVPN route types and RFC 9136 introduces the IP Prefix advertisement option into EVPN. The multicast EVPN route types RT-6, RT-7, and RT-8 are introduced in RFC9251, and we already briefly described those routes.

To better understand the usage of each route, let's review the main purposes and explain when and why these route types are typically used.

EVPN Route Type 1 (RT-1): Ethernet auto-discovery route

Each leaf node in the data center fabric advertises RT-1 routes for each locally attached Ethernet segment for a given EVI.

When there is a connectivity failure to the attached segment, the PE withdraws the corresponding set of Ethernet A-D per ES routes.

If the leaf nodes (aka PEs) that receive the withdrawal have another A-D route for this segment, they will update the next hop accordingly. If not, they will invalidate the MAC entries for that segment.

This route type brings fast convergence, aliasing, backup path, and split horizon protection.

EVPN Route Type 2 (RT-2): MAC and MAC/IP advertisement route

This route determines reachability to unicast MAC Address and brings load balancing of unicast packets.

When a leaf node receives a packet from a server, it performs local data plane learning and learns the source MAC address. Then it advertises the MAC address using an EVPN RT-2 and it optionally adds the IP address in the advertisement.

A PE learns remote MAC addresses that sit behind other PEs using EVPN control-plane learning.

If the optional IP field is set, this route can be used to minimize the flooding of ARP or Neighbor Discovery. When a leaf node receives an ARP request for an IP address and it has the MAC address binding, it should perform ARP proxy by responding to the ARP request. Proxy ARP/ND, and ARP/ND suppression are benefits of RT-2.

Additionally, when the symmetric inter-irb is enabled in the data center fabric the MAC/IP RT-2 is used as well to perform the local IP routing operation at the ingress and egress leaf node.

EVPN Route Type 3 (RT-3): Inclusive Multicast Ethernet Tag route

This route describes handling of multi-destination packets and processing of unknown unicast, broadcast, and multicast packets. It's used to deliver the ingress replication mechanism and replicates the multicast feed to the egress leaf nodes within the given VLAN-VNI. When optimized multicast is used, the SMET and OISM flag capability is also added as part of the RT-3 advertisement.

EVPN Route Type 4 (RT-4): Ethernet Segment Route

This route is used for the designated forwarder election and ESI segment 10-byte value advertisement. This election will happen at ESI-LAG on the leaf nodes where servers or CE switches are connected or at the interconnect gateway level. Different DF/nDF election mechanisms can be used, the default and most common one is using the MOD based DF/nDF election, but preference-based election is also popular to control precisely which node is elected as the designated forwarder for the given ESI segment.

BGP EVPN Communities

RFC7432 describes new BGP EVPN communities, here are four of them:

- ESI Label extended community: this community is used for split horizon of multi-homed sites.
- ES-Import Route Target (carried with RT-4): this community enables all PEs connected to the same multi-homed site to import the ES routes. It is used for multi-homed Ethernet segment auto-discovery.
- MAC Mobility extended community: this community is used to detect MAC duplication issues.
- Default gateway extended community: PEs that act as a default gateway for a given EVPN instance can advertise their MAC address using a RT-2 advertisement carrying the default gateway extended community.If all PEs use the same default gateway MAC address for a given EVPN instance, such advertisement is not needed.

Integrated Routing and Bridging in EVPN

When L3 service is required for a bridge domain, an L3 interface called IRB interface is used. Each PE or leaf devices need IRB functionality to enable efficient and optimized routing of tenant traffic.

The PE acts as an IP default gateway with a MAC and an IP address configured on each IRB interface associated with its subnet.

An IP-VRF table is created along with one or more MAC-VRF tables. The IP-VRF is identified by its corresponding Route Target and Route Distinguisher. RT-2 are not only used to populate MAC-VRF instance but can also be used for host routes (/32 or /128) in the IP-VRF table. There are 2 routing scenarios: symmetric and asymmetric IRB (described in RFC 9135).

Router's MAC extended community

A new EVPN BGP extended community called *EVPN Router's MAC* was introduced for symmetric IRB scenarios. It is used to carry the PE's MAC address along with EVPN RT-2. In some cases this community can also be used along with RT-5 routes.

Route Type 5 – IP prefix advertisement using EVPN

There are cases where RT-2s are not suitable and the server connection to be delivered is just an IP connect. In this case, the EVPN-VXLAN based fabric can still use the same EVPN signalization and use it for IPv4/IPv6 prefix advertisement, similar in functionality to IPVPN services.

You might need to advertise aggregated IP prefixes instead of individual host routes to decouple the advertisement of the prefixes from the advertisement of a MAC address. RFC9136 describes a new EVPN route type: Route Type 5 – IP Prefix advertisement.

Route type 5 (RT-5)

Route Type 5 provides a clean and clear advertisement of IPv4 or IPv6 prefixes without MAC Address involved, which in larger scale fabric implementation can help to scale out the fabric. When more servers need to be connected, typically the ToR switch FIB is higher for IP than the TCAM for MAC@, so using the pure Type-5 EVPN extensively is often a good practice for higher scale EVPN fabrics.

IP-VRF to IP-VRF model

In some cases, IP Prefix routes may be advertised for subnets and IPs sitting behind an IRB. This use case is referred to as the *IP-VRF-to-IP-VRF* model. There are three different scenarios when it comes to routing from an IP-VRF to an IP-VRF:

- Interface-less model (most popular in the industry)
- Interface-ful with an SBD IRB model
- Interface-ful with an unnumbered SBD IRB Model

Juniper implements interface-less model, also called *Pure Type 5*.

Interface-less model

In this model, RT-5 is used to advertise IP prefixes along with a router's MAC extended community.

A VNI will be associated to the VRF and advertised with RT-5. In this case there's no requirement to have any specific server interface to be associated with the given VRF to advertise a prefix. In the context of DCI, the interface-less model is used when stitching Type-5 to Type-5 VXLAN tunnels – where in fact there's no IFL interfaces associated with the various prefixes and everything is handled from the forwarding point of view using the next hop recursiveness for the given destination IP prefix.

Interface-ful models

In these models, RT-5 advertisse IP prefixes whereas RT-2 advertises MAC-IP addresses of each SBD IRB interface.

RT-5 requires a recursive lookup resolution to an RT-2 route. VNI from RT-2's MPLS label 1 field is used when forwarding packets.

The Supplementary Broadcast Domain (SBD) is created for the recursive lookup with no attachment circuit (VLAN) and it has an IRB interface that connects the SBD to the IP-VRF.

Table 1.2 Fields Required for Each Model in the RT-5

Interface-less model	Interface-full with an SBD IRB model	Interface-full with an unnumbered SBD IRB Model
No SBD and no Overlay Indexes required.	Each SBD IRB has an IP and a MAC address where the IP is reachable RT-5 used to advertise IP prefixes and RT2 used to advertise MAC/IP of each SBD IRB.	Each SBD IRB has a MAC address only and no IP. RT-5 used to advertise IP prefixes and RT-2 used to advertise MAC of each SBD IRB
Route distinguisher Ethernet-tag-id = 0 IP Prefix length GW IP = 0 VNI = VRF VNI Router-MAC community VRF RT	Route distinguisher Ethernet-tag-id = 0 IP Prefix length GW IP = IRB IP of SBD VNI = 0 (RT-2 VNI will be used) No Router-MAC community VRF RT	Route distinguisher Ethernet-tag-id = 0 IP Prefix length GW IP = 0 VNI = 0 (RT-2 VNI will be used) Router-MAC community VRF RT

Multi-tenancy and Network Virtualization in the EVPN VXLAN Data Center

Previous chapters reviewed different route types and BGP communities, the building blocks of the EVPN control plane. They are also part of the seamless DCI solutions which is our focus in the coming chapters but depending on the type of DCI option, they may be modified to deliver more efficient and scalable network solutions.

The introduction of different EVPN route types and different EVPN services (VLAN-based, VLAN-aware, VLAN-bundle) also helps in delivering the multi-tenancy, network virtualization, and tenant isolation into the DC fabric.

The fabric network virtualization and tenant isolation capabilities of EVPN are leveraged at the server leaf switch and border-leaf level. The virtualization options, incorporated in EVPN-VXLAN fabric, offer solutions to the following requirements:

- Security and service chaining: depending on the type of services to be offered, they define if one service can talk directly to another within the given top of rack or must be fully isolated and inspected further at the L4-L7 firewall level
- Agility to easily expand the design: a higher number of leaf nodes and higher bandwidth is possible within the fabric, or between the data center sites, while maintaining the same level of security and services isolation

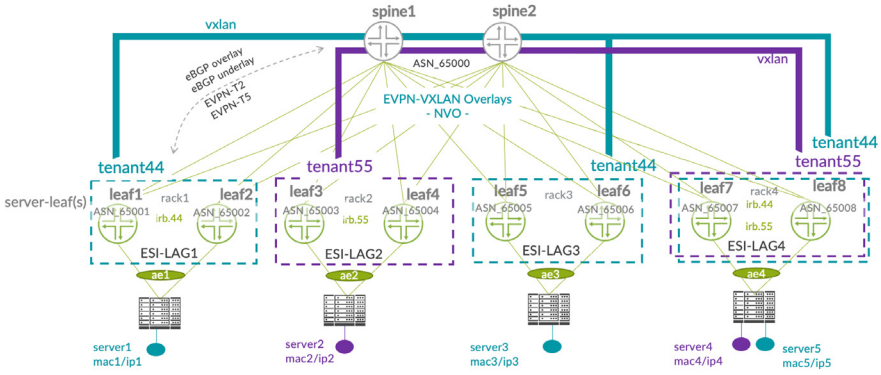


Figure1.6 *EVPN-VXLAN DC Fabric NVO - tenant distribution example within the DC fabric*

The example tenant distribution illustrated in Figure 1.6 shows tenant44 and tenant55 each having dedicated EVPN-VXLAN EVI (MAC-VRFs) Layer 2 contexts.

Based on the decision of the admin, two DC racks were dedicated to tenant44, one rack to tenant55, and the fourth rack was enabled for both tenant services. The tenant can be

also viewed here as a different service-id. In many DCs that type of tenant/service distribution is however not staying for long because many servers are simply moving from one rack to another or are added based on decommissioning, maintenance, memory upgrade cycles, etc... That's also why, when any changes are needed in the overlay fabric infrastructure, they can be easily adapted at the leaf to server port level, without any changes in the rest of the infrastructure.

For example, when tenant55/service55 must move to rack3, the fabric manager tool, or simply the admin, can quickly add that tenant only to that new pair of leaves, without changing anything in the rest of the fabric. This was not possible with the traditional LAN data center infrastructures where the tenant state was propagated manually on each node of the fabric instead of being fully distributed via the control plane protocol. In fact, EVPN as a standard was designed with built-in automation so the propagation of the information of a new connected tenant is automatically done via the BGP family EVPN advertisement to the rest of the fabric nodes. Based on the BGP EVPN control plane information (different EVPN route types) exchange between the leaf nodes, the VXLAN tunnels are automatically established between the fabric nodes, when they are sharing the same MAC-VRF EVPN instance(s).

Regarding the tenant network overlay infrastructure segmentations and virtualization options supported in Junos/Junos Evolved on the QFX family of products, consider these following network virtualization mapping options in Figure 1.7.

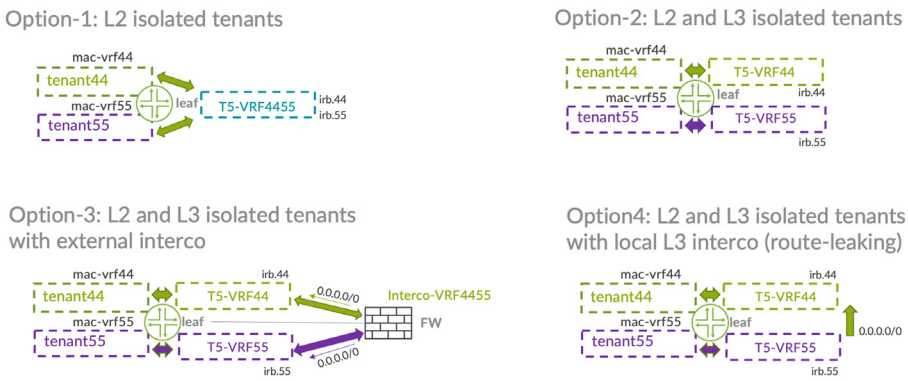


Figure1.7 *Tenant Network Virtualization Mapping Options in EVPN-VXLAN Fabric Overlays (NVO)*

- Option-1 isolates the tenant services at the L2 level but points to the same Type-5 instance through which the tenant44/tenant55 can communicate using local IP routing in the leaf.
- Option-2 is offering a 1:1 mapping of L2 MAC-VRFs to Type-5 L3 VRF, where each tenant or service gets an MAC as well as IP context, in this case there's no communication between the two services within the given node.
- Option-3 of the mapping is adding the possibility to communicate between the two services but via the external firewall for advanced L4-L7 inspection.
- Option-4 still offers a 1:1 mapping between the L2 and L3 context but, instead of fully isolating two services, it leaks a specific prefix between the two tenants – for example a default route.

The good thing about the proposed network virtualization options is that one set of tenant/services may follow the option-1 and the other set of services may follow another option.

Using the mapping concepts highlighted above, the following services options can be considered within the EVPN-VXLAN fabrics.

Table 1.3 Services Options Within EVPN-VXLAN Fabrics

Leaf service option	Service type	Description	Pros	Cons
1	L2 transparent services (VLAN-bundle)	Overlay circuit approach using VXLAN where any customer traffic is sent across the overlay	Easy to manage High user VLAN scale Overlapping VLANs	Limited intra tenant isolation No parallel IP services for the end-user
2	Combination of L2 & L3 virtualization	User traffic is processed ingress and isolated at L2-VPN and L3-VPN level for each service	Better security for end user – full isolation at MAC and IP level Option to seamlessly stitch at GW	Requires more automation tools to manage and deploy
3	IP Virtual Private Network using pure T5 virtualization	User gets the IPVPN virtualization type of service	No L2 services to be managed between the leaf nodes	More subnets to manage per tenant.

The services options suggested in Table 1.3 can be used within the given enterprise or telco-cloud EVPN-VXLAN fabric and used within the same node, this is visualized in Figure 1.8s fabric diagram where leaf1 communicates with leaf2 and the same physical 10GbE interface xe-0/0/1 is used to offer all three services options, each with different characteristics.

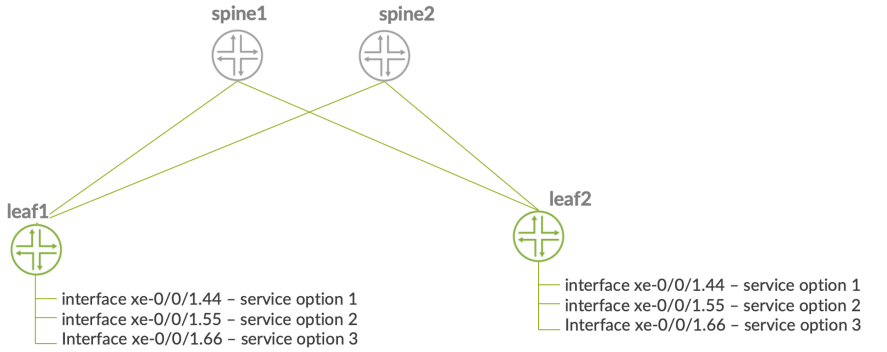


Figure 1.8 *EVPN-VXLAN fabric virtualization and services options*

Different network virtualization options offered by EVPN-VXLAN MAC-VRFs via different service-types (VLAN-based vs VLAN-aware).

In the example in Figure 1.9, for the VLAN-aware service type, the tunnel gets created between leaf1 and leaf2 even if leaf1 is not provisioned with any VLAN BD55. This is not the case for the VLAN-based EVPN service-type, where the auto-discovery routes are specifically allocated to each MAC-VRF EVI for each individual VLAN. The VLAN-based approach reduces the total number of VXLAN tunnels in the fabric, provided we have a very selective way of provisioning the tenant's networks on the fabric leaf devices. In the case of VLAN-aware, there is a common MAC-VRF for both tenant services. Even if the BD55 is not enabled on leaf1, the VXLAN tunnel will be pre-established between the leaf1 and leaf2.

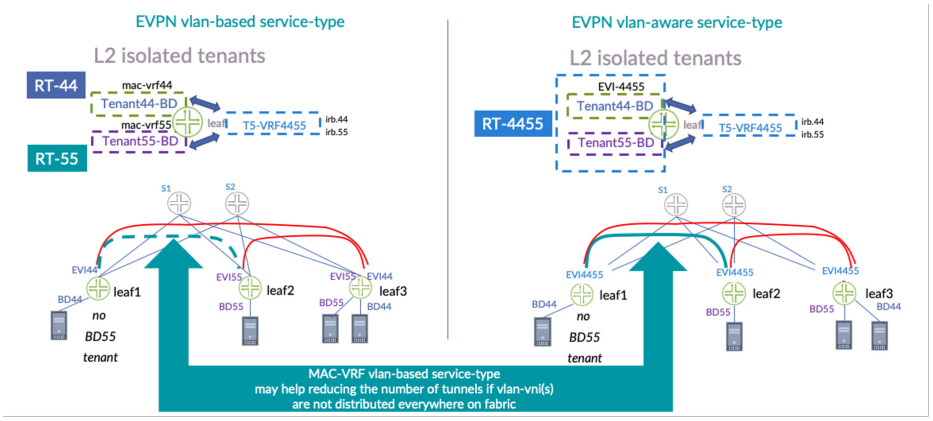


Figure 1.9 *Tenant Virtualization and EVPN Service-types Implications on VXLAN Tunnel Establishment*

On the other hand, if the EVPN-VXLAN fabric design requires all VLAN-VNIs to be enabled on all leaf nodes in the fabric, the selective MAC-VRF provisioning high- lighted above is not so efficient anymore and morover, auto-discovery routes will be present in the RIB of each leaf node. Indeed, when fabric design requires all VLANs to be present everywhere then the VLAN-aware service-type grouping multiple VLAN- VNIs under the same MAC-VRF instance will contribute to reduce the RIB (Routing Information Based) utilization, at the control plane level. Because in VLAN-based EVPN service-types, each MAC-VRF is just use one VLAN-VNI, then each new VLAN-VNI enabled on all leaves generates an independent auto-discovery route (AD route), slightly increasing the number of control plane entries distributed across the fabric.

Intra-Datacenter Architectures

Modern data center networks use multistage topologies inspired by Charles Clos. He designed a 3-staged non-blocking network to switch telephone calls. Let's review three data center topologies: 3 stage CLOS, 5 stage CLOS, and collapsed.

3 Stage IP Clos Fabric

CLOS networks are highly scalable, resilient, and high-performing. In a 3 stage CLOS topology there are two layers: the leaves and the spines. In Figure 1.10's diagram, L1, L2, and L3 are leaf devices and S1 and S2 are spine devices.

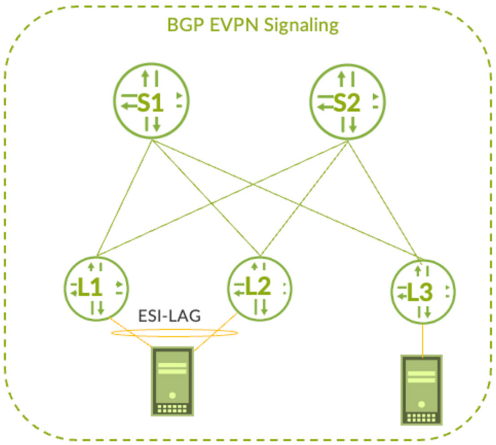


Figure 1.10 *3-stage IP Clos Fabric*

Each leaf is connected to all spines but the leaves are not connected to each other. The hosts are connected to the leaves and there's a maximum of three hops between any two hosts connected to the fabric: leaf – spine – leaf. The uplink bandwidth of each leaf can be increased by adding new spines.

At the leaf layer, if the aggregated bandwidth of the downlink interfaces is equal to the aggregated bandwidth of the uplink interfaces, then there is no oversubscription and the architecture is *non-blocking*. The maximum number of spines is the number of leave's uplink interfaces. The maximum number of leaves is the number of spines' interfaces.

5-stage IP Clos

When a 3-stage topology reaches its limit and there is a need for additional leaves or for cabling considerations, it can evolve into a 5 stage CLOS. See Figure 1.11.

A 5 stage CLOS interconnects two or more 3 stage PoDs with the use of an additional layer called *super spine*. The super spines are connected to the spines from each PoD. In the following diagram, SS1 and SS2 are the super spines interconnecting the two fabric PoDs.

There's a maximum of five hops between any two servers connected to the fabric: leaf > spine > super spine > spine > leaf. For a non-blocking architecture, there must be no oversubscription at both the leaf and spine layers. Evolving from a 3-stage to a 5-stage architecture is an easy and efficient way to increase the scalability of a data center and add more redundancy to it. For example, two different DC rooms can be interconnected using super-spine blocks of architecture.

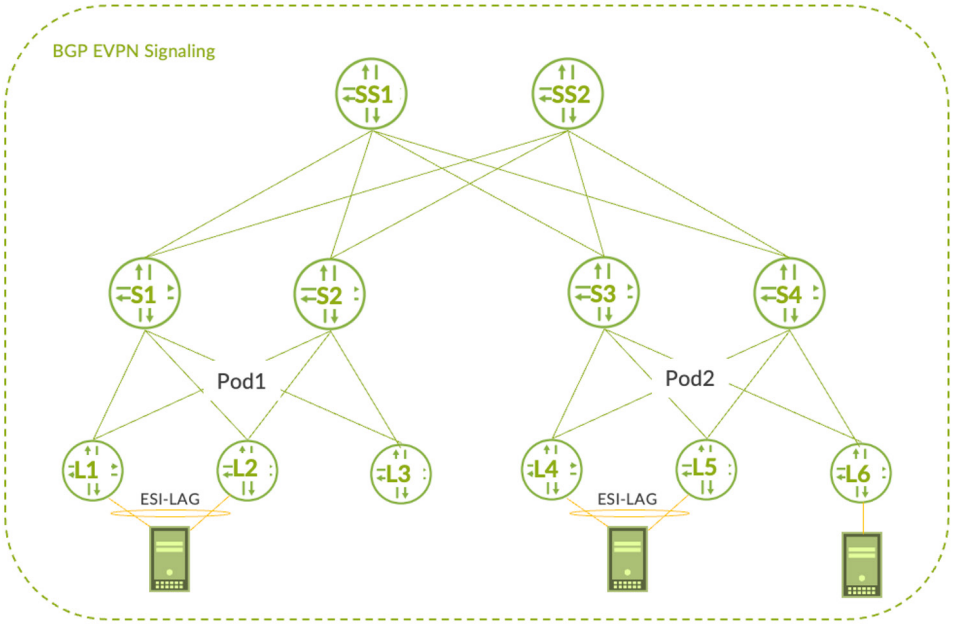


Figure 1.11 *5-stage IP Clos Fabric*

Similarly, to the 3-stage architecture, in each of the PoDs you can deploy a BO or ERB EVPN-XLAN design and have, additionally, in each PoD the border-leaf block connecting to the core IP network. In most of cases, the spines and super-spines will be used for IP forwarding and EVPN route-server purposes, with the exceptions of multicast assisted replication, when there's a reason to connect the existing L2 domain directly to the spines or when the spines in each PoD are used for the VXLAN to VXLAN stitching function, discussed in greater details in upcoming chapters.

Collapsed Core EVPN-VXLAN Design

For a very small DC edge site, the collapsed topology is ideal. It is the standard way of doing MC-LAG. You can benefit from the advantages of EVPN control plane with only two devices (or even more) and, when needed, connect it to remote locations also using EVPN.

Figure 1.12 Collapsed Spine with EVPN-VXLAN

When using collapsed core topology, the seamless stitching techniques are typically not required because there are not so many fabric VXLAN tunnels beside the one between the two leaf nodes, like L1 and L2 in Figure 1.12. In that ESI-LAG scenario, leaf L1 and L2 will have one VXLAN tunnel back-to-back and, when required, will establish VXLAN tunnels towards remote sites.

In the scenario where the collapsed core is connected directly to an MPLS core, the ESI-LAG will be used as a server multihoming technology, however the EVPN-MPLS will be used on the MPLS network and EVPN-VXLAN won't be required.

When connecting collapsed core topology to the core IP but without direct MPLS connect, the ESI-LAG is still used for active/active multihoming but with the VXLAN as encapsulation and EVPN BGP as signalization.

The nice part of the collapsed core topology is that it's not limited to two nodes and can be enabled with four nodes or more in full mesh topology bringing additional level of redundancy and bandwidth to servers or appliances.

NOTE In the case of collapsed spine topology, you typically don't use any of the seamless stitching techniques covered later in the book as the tunneling part starts from the collapsed spine to the remote location.

Bridge Overlay and Edge-Routed Bridging

EVPN-VXLAN architectures can be deployed to deliver L2 service only, this is called Bridged-Overlay (BO) or L2/L3 services, called in that case Edge-Routed Bridging (ERB).

In both cases the L2 VLAN-VNI stretching between top of racks can happen. However, in the case of ERB the first hop IP gateway is fully distributed and enabled at each leaf node, while in the case of BO, the IP first hop gateways are outside of the fabric, for

example, at the existing DC-GW devices in the given PoP location or when enabling the first hop IP gateway at the firewalls.

In the case of the BO, the traditional VRRP is used as the external DC-GW to deliver the IP gateway function to the servers connected to the server-leaf EVPN-VXLAN Layer 2 fabric.

In the case of the ERB design, we get the benefit of having a reduced blast radius – reduced failure domain thanks to the distributed anycast IP gateway model, while in the case of the BO fabric design, we maintain the centralized IP first hop gateway model.

In both cases, ERB and BO, when connecting to external devices, like existing firewalls, existing DC gateways, or load-balancers, it is typically done through the border-leaf block as highlighted in Figure 1.13.

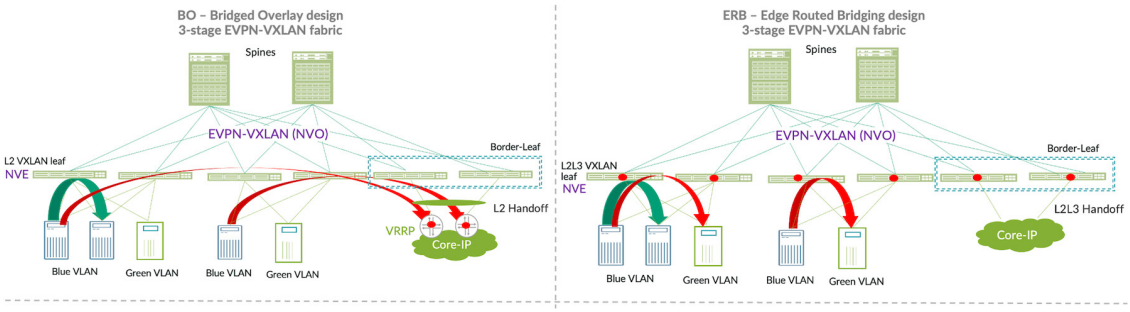


Figure 1.13 *3-stage IP Clos fabric - Bridged-Overlay vs Edge-Routed-Bridging design*

In both cases, the spines may play the role of IP forwarders and route-servers for EVPN and VTEP tunnel termination (NVE) will only happen at the server-leaf and border-leaf.

There are however situations where spines are also used in the role of the border spine – this design is called CRB and can also be a useful way of optimizing the EVPN-VXLAN DC fabric previsioning and scale. When most of the traffic is north-south, firewalls may connect to the spines directly. Or when the spines are used for assisted replication, as in the case of heavy deployments with multicast, with a lot of elephant multicast traffic. The spine will also become the NVE and will be used for VXLAN tunnel termination.

Underlay and Overlay Protocol Considerations

In an EVPN VXLAN data center environment, routing protocols are deployed in the underlay and the overlay. The first step is always required: get the underlay functional to get full IP reachability of the loopback IP addresses within the fabric and between the fabrics deployed in various locations.

Overlay Protocol

The overlay will be BGP-based as it handles EVPN signaling. For that, there are two choices: eBGP or iBGP.

- iBGP
  - all devices have the same overlay ASN
  - a different protocol is needed for the Underlay or the underlay is using eBGP with different ASN per node
- eBGP
  - Same routing protocol can be used for both Underlay and Overlay
  - Same BGP ASN is used for underay and overlay
  - No next-hop change is needed to preserve the original Next Hop
  - Each fabric node is identified by an ASN and it provides AS PATH loop detection
  - Follows RFC7938 recommendation

Underlay Protocols

The underlay will oversee advertising the loopback addresses used by the overlay and perform ECMP. With the use of ECMP on top of an efficient routing protocol, all links of a data center fabric are used making the data center network performant. For the underlay, an IGP can be chosen, mainly OSPF or ISIS, or an eBGP as well.

Note that eBGP has many advantages:

- Same protocol can be chosen for both underlay and overlay: each device will have a single ASN and it will be easier to configure.
- eBGP is also the protocol used to peer the Fabric with WAN routers, knowing only one protocol is sufficient.
- eBGP is more scalable than OSPF and ISIS and will be a better choice for very large scale fabrics.
- eBGP is less complex than a link-state IGP.
- The operator decides what route to advertise using simple routing policies.
- It is easier to troubleshoot as received and advertised routes can be monitored.
- Although eBGP is known for slower routing convergence compared to IGP, some mechanisms make it as fast as an IGP. For the underlay, eBGP peerings are configured on point-to-point fiber connections and a physical interface failure triggers a BGP reconvergence in milliseconds.

Sub-optimal Forwarding

Traffic forwarded to a dual-homed server (server2 in Figure 1.14) is load-balanced between both leaves (L2 and L3).

In some cases of multiple failures, like the example below, the traffic between L1 and L3 is sub-optimal and goes through another leaf of the fabric (L2 in this case). If the same ASN is configured on both spines S1 and S3, then a routing loop will be detected and the route leading to sub-optimal traffic will be rejected. But in the case of a single-homed server to L3 (server3), you need the sub-optimal traffic to happen. To address this dual failure scenario, it will be needed to authorize an AS loop or to configure different ASN on the spines.

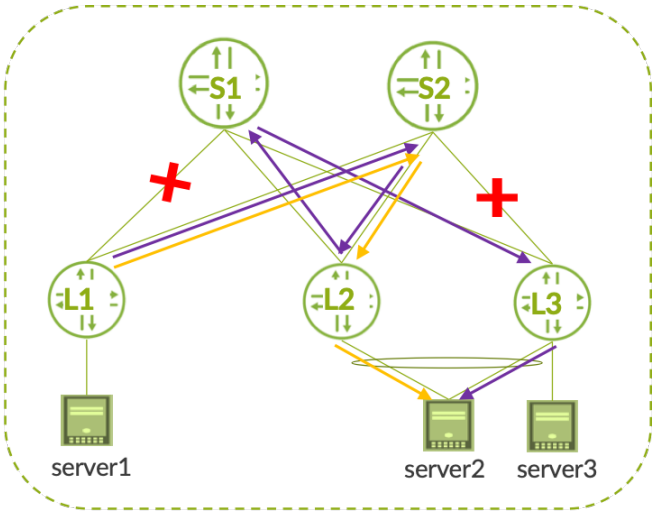


Figure 1.14 Sub-optimal Forwarding

MPLS Underlay

In the case of MPLS, an IGP is typically needed for loopback or label signaling (SPRING), OSPF and ISIS are the most deployed IGPs. As all nodes must belong to the same ASN, iBGP will be preferred for the EVPN MPLS scenario. This is something that is also demonstrated in the implementation section of the EVPN-VXLAN to EVPN-MPLS lab (Chapter 6).

Chapter 2

Network Overlays and DCI

The NVO's virtualization capabilities such as MAC-VRFs (EVIs) or T5 L3-VRFs enabled at the top of a rack leaf switch within the DC fabric, have become even more important when the given tenant service needs to be extended in multiple data center locations or PoDs. Extending the given tenant to a different DC site without touching the intermediate IP domains or intermediate spines/super-spines is offering additional agility, faster service delivery, and better data access redundancy model for the data center.

The interesting part of the network virtualization overlays for DCI (Data Center Interconnect) is that delivering additional service options is much easier. For example even if the existing IP core undelay is not multicast capable, the EVPN-VXLAN can be used to extended multicast between the DC sites for specific tenant/service, without any multicast requirements in the core IP network.

In the context of the DCI, the network virtualization can also help in the definition of which data is accessible between the DC sites. In Figure 2.1 the main DC site A has two PoDs but, due to data sensitivity, the replication of the data from PoD1 can only happen towards the DC site B and not to site C. Using the MAC-VRF network virtualization, the admin can decide to allocate one MAC-VRF to PoD1 and one MAC-VRF to PoD2 in site A, efficiently partitioning the two data PoDs from site B and site C connection.

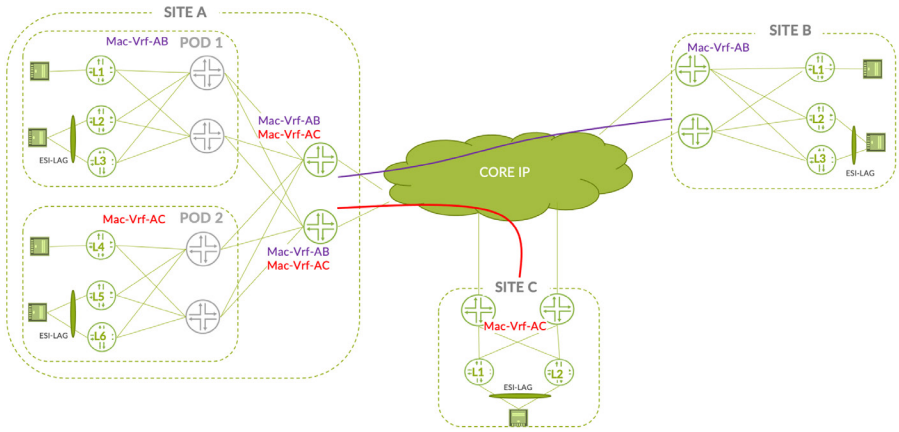


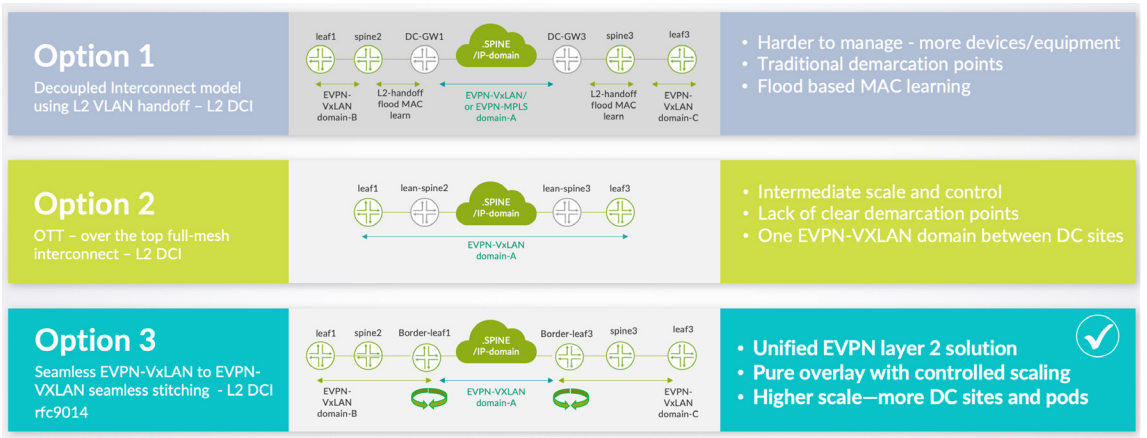
Figure 2.1 *EVPN-VXLAN Network Virtualization and Data Center Interconnect (DCI)*

The following chapter describes several DCI options improving the disaster recovery design, while preserving the same data isolation and data access models across multiple geographical locations.

The newer seamless stitching Layer 2 DCI options (option-3/option-4) will be covered in more details in the next chapters from the control plane and data plane perspective. The implementation part will be also detailed in the lab dedicated chapter.

Data Center Interconnect Options

DCI design can be solved in a different manner with a different level of effort and total cost. In Figure 2.2, the most popular DCI design options are highlighted.



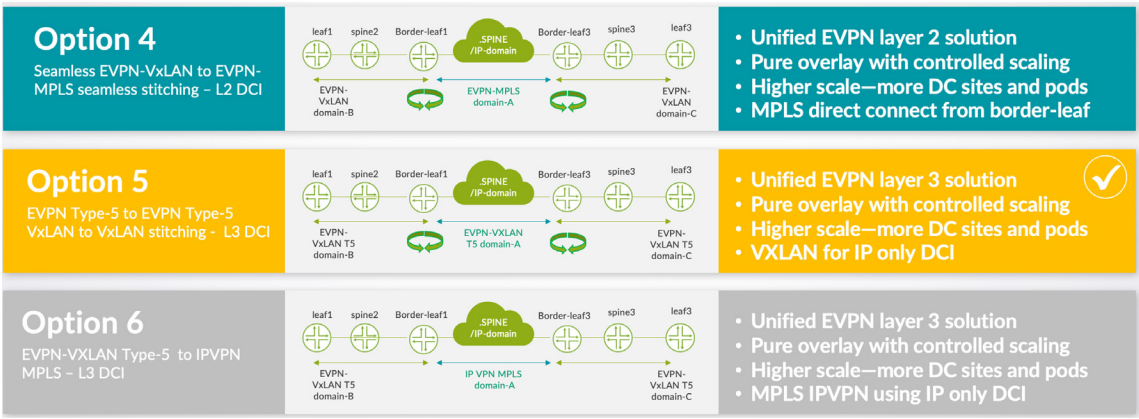


Figure 2.2 *DCI Design Options Summary*

The summary of DCI shown here highlights the main characteristics for each option and helps to identify the differences. For example, between options 1/2/3/4 and option-5/6, the main difference is that the first four design options are to be considered for L2 stretching requirements, and the last two options are for the IP-only based DCI. Between different options, it can be noticed that even if one option may appear easy to implement at the early phase, then over time when fabrics are growing or when number of DC sites is increasing, then the easy way of implementing may be problematic from scaling, operation, and the control point of view.

It is possible to mix several options to have a full flavor DCI. For example, you could mix options 3 and 5 to have both L2 and L3 stretching with VXLAN to VXLAN stitching, or options 4 and 6 to have both L2 and L3 stretching with VXLAN to MPLS DCI. Table 2.1 highlights some additional capabilities that should be considered during the design decision process.

Table 2.1 *DCI Option Decision Criteria*

	Option-1: Decoupled VLAN-handoff – L2	Option-2: Over-the-top (OTT) – L2	Option-3: Seamless EVPN-VXLAN to EVPN-VXLAN stitching – L2	Option-4 EVPN-VXLAN to EVPN-MPLS seamless – L2	Option-5: EVPN Type-5 to Type-5 VXLAN encapsulation – L3	Option-6: EVPN Type-5 to IPVPN MPLS encapsulation – L3
L2 stretch	✓	✓	✓	✓	✗	✗
End to end EVPN control-plane	✗	✓	✓	✓	✓	✓
Unified VXLAN forwarding	✗	✓	✓	✗	✓	✗

Controlled VTEP scale	✓	✗	✓	✓	✓	✓
Same subnet everywhere	✓	✓	✓	✓	✗	✗
Built-in DCI VLAN translation	✗	✗	✓	✓	✗	✗
VMTO	✗	✓	✓	✓	✓	✓
ARP/ND suppression	✗	✓	✓	✓	✓	✓
Flood reduction between sites	✓	✗	✓	✓	✓	✓
High scaled DCI for multisite	✓	✗	✓	✓	✓	✓
L3 full isolation	✗	✗	✗	✗	✓	✓

The first four options from Table 2.1 are offering L2 connectivity between the sites while option-5 and option-6 can be either deployed as the main DCI solution without any L2 extensions or added as an additional L3 service when one of the first four options is used. When IP routing of the tenants is not handled by the fabric (bridged overlay design), then the option-4 and option-5 are not typically considered for the DCI. In the next sections, we will be describing some of these options in more details.

DCI Option 1: VLAN Handoff

In the VLAN handoff scenario, each data center has its own control plane. Therefore, all EVPN control plane and data plane values (RT, RD, ESI and VNI) have local significance and there is a clear demarcation between the several sites.

At the external data center gateway (DC-GW) level, it is possible to control the list of VNIs that need to be stretched, those are mapped to a VLAN and handed over to the backbone. All MAC learning from the EVPN-VXLAN to the DC-GW is done using traditional flood and learn.

The backbone connecting the sites must be able to transport the extended VLANs, it has to offer L2 services. It's recommended to have several gateways on each site, so the backbone PE must handle also the L2 services and be aware of the VLANs enabled in the fabric, which typically takes longer to implement as a DCI solution when the PE nodes are not managed by the data center teams.

In terms of configuration, each extended VLAN must be added at the gateway level and transported by the backbone, that represents many configuration tasks with an increased risk of error.

When L3 is involved, it adds even more complexity. At the gateway level, each VRF must advertise its routes to the other site, and this can be done by configuring an external BGP peering inside each VRF. In this scenario, distributed routing, meaning a distributed gateway for stretched VLANs is very complex because we cannot benefit from the EVPN anycast gateway capability. For example, we would have to configure VRRP with a knob to make it active/active between datacenters.

The way handoff is enabled at the border-leaf nodes in each DC site is shown at Figure 2.3's simplified topology diagram.

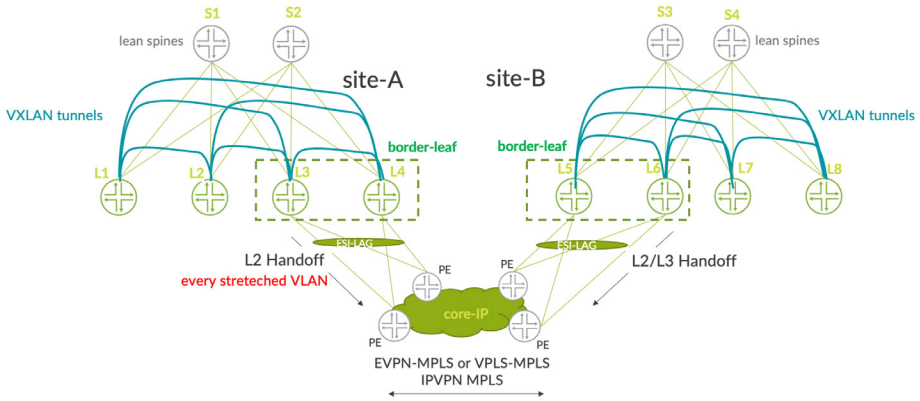


Figure 2.3 DCI Using L2 Handoff from the Border-leaf

You can see that the VXLAN tunnels always terminate at the border-leaf level and it's the role of the external DC-GW to deliver the DCI services.

DCI Option 2: Over-The-Top (OTT)

In the OTT scenario, all sites are seen as a single site or single domain from a control plane point of view. The main advantage here is simplicity: all devices are auto discovered with BGP and VXLAN tunnels built automatically between all leaves belonging to the same MAC-VRF/EVI.

L2 can be stretched very easily and L3 can be distributed with the use of anycast IRB in the same way it is done using a single site.

However, there are some drawbacks:

- Scaling: when adding a leaf on a site, it must build a VXLAN tunnel to all leaves of all sites for the same network. Each new device dramatically increases the scaling of the network.

- The need to have unique control plane and data plane values across all sites: RT, RD, ESI, and VNIs must be unique and have a global significance.
- Blast radius: if a configuration error is made or a problem occurs on one site, it could potentially impact all sites at the same time

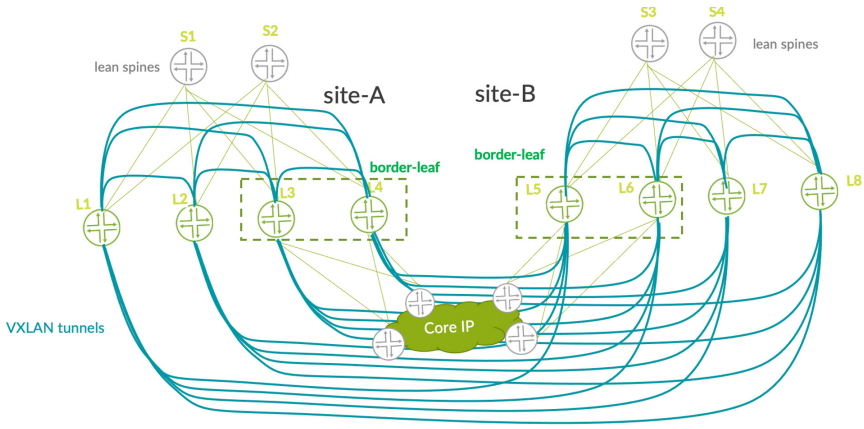


Figure 2.4 *OTT DCI EVPN-VXLAN Solution and VXLAN Full Mesh of Tunnels Between DC Sites*

When it comes to multi-PoD DC architecture the number of VXLAN tunnels between the PoDs may become massive over time when having all tenants provisioned on all leaf devices. Especially, when you are required to simply pre-provision all VLAN-VNIs on all leaf devices from the same fabric everywhere – enabling a flat L2 EVPN-VXLAN design. This situation is visualized in Figure2.5 where, on the left side, the fabric is creating massive number of tunnels when connecting the new leaf in PoD4 versus the new seamless EVPN-VXLAN stitching solution on the right side offering fully controlled and optimized interconnect solution for the multiPoD DC fabric.

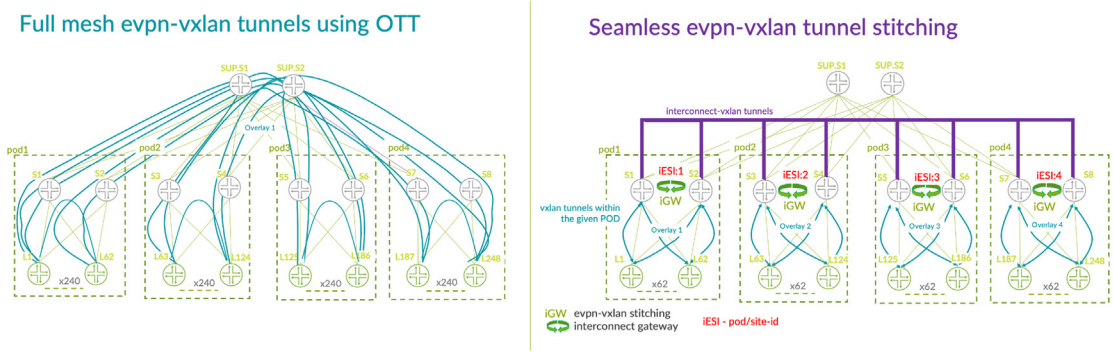


Figure 2.5 *Full Mesh EVPN-VXLAN Versus Seamless EVPN-VXLAN Stitching*

In fact, in the case of the seamless stitching solution, the leaf from the given PoD will terminate the interPoD tunnel at the interconnect gateway level (iGW) spine and use the EVPN signalized interco tunnel that had been established even before any additional new leaf gets connected.

From the tenant distribution point of view, the seamless EVPN-VXLAN stitching is giving an additional centralized tool where you can be more selective on which tenants L2 networks are stretched between the PoDs or DC sites.

DCI Option 3: Seamless EVPN-VXLAN to EVPN-VXLAN Stitching

In this DCI design option, you can enable the L2 VLAN-VNI extension between sites, in a selective manner within the given MAC-IP (EVI). Typically, this is done at the border-leaf level of the DC Fabric without interfering into the existing PE devices, and the admin of the fabric can decide which are the VLAN-VNIs to be extended.

With the VXLAN-to-VXLAN seamless stitching the local LAN VXLAN tunnels get terminated at the border-leaf and new DCI VXLAN tunnels are automatically established, based on EVPN control plane advertisements. In the VXLAN-to-VXLAN design option, you don't change the type of encapsulation but rather optimize the reach of the VXLAN domains. This increases the scale of the infrastructure and simplifies the number of operations needed for the L2 extensions. Once the interconnect VNIs are defined at each site, on the border-leaf block, the interconnect VXLAN tunnels are pre-established.

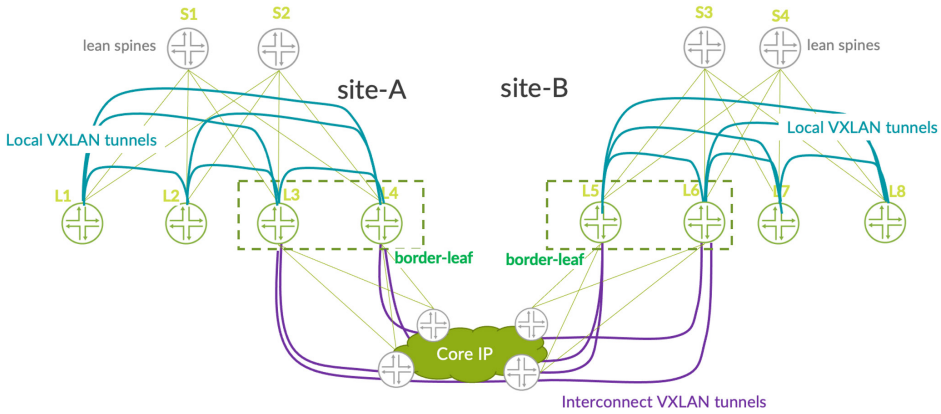


Figure 2.6 *Seamless Stitching - EVPN-VXLAN to EVPN-VXLAN - Reduced Number of Site-to-Site VXLAN Tunnels*

DCI Option 4: Seamless EVPN-VXLAN to EVPN-MPLS Stitching

Similarly, to the previous DCI option, the seamless EVPN-VXAN to EVPN-MPLS stitching optimizes the operational aspect of L2 DCI extension as well as improves the scaling when there are multiple DC sites to interconnect. This option of DCI is mainly applicable when there's an existing MPLS core network. Instead of having a dedicated DC-GW node, you have a more cost-effective solution, where the border leaf connects directly to the MPLS and integrates the border-leaf/DC-GW function in the same node.

The seamless EVPN-VXLAN to EVPN-MPLS example topology is highlighted below where border-leaf is connected on LAN side using EVPN-VXLAN and on DCI side using EVPN-MPLS, still using the same EVI (EVPN Instance). This is depicted in Figure 2.7's basic topology where the border-leaf nodes are just changing the encapsulation format from VXLAN to MPLS and follows the MPLS LSP to send the L2 packet to the remote destination.

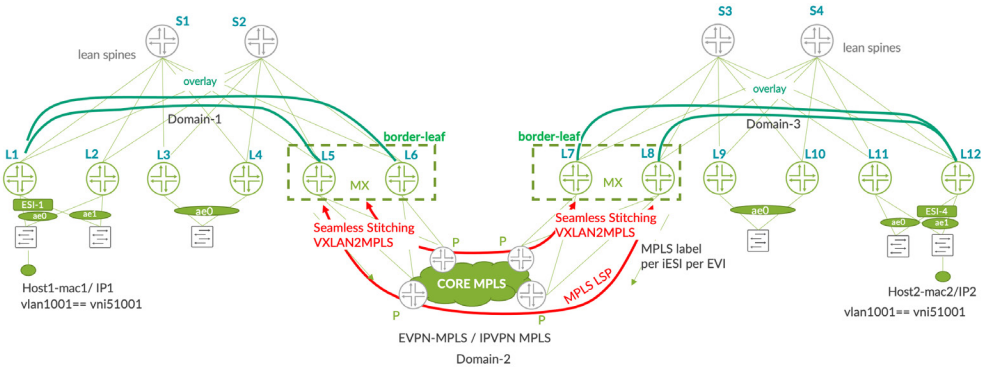


Figure 2.7 Seamless Stitching - EVPN-VXLAN to EVPN-MPLS - LSPs Between DC Sites

Option 5: EVPN-VXLAN T5 to EVPN-VXLAN T5 Seamless Stitching

When the fabric is configured as ERB with IP prefix advertisement services using pure T5 EVPN routes at the server leaf level, then the L3 context stretching is often needed in conjunction with L2. The goal is to extend T5 IP VRFs between data centers without having to build VXLAN tunnels from a leaf of one data center site to- wards all the leaves of remote data centers.

In the same manner as L2 stitching, the gateway device, the border-leaf at the edge of the data center, terminates the T5 VXLAN tunnels and builds new tunnels towards the edge of the remote data center. This provides all the advantages of seamless stitching: better scalability, control plane demarcation, better control, site identification.

In the case of the Type-5 to Type-5 stitching, option-5 is typically combined with option-3 where only a subset of VLAN-VNIs is stretched between DC sites, and the rest of reachability between the sites is delivered using IP prefixes.

Option 6: EVPN-VXLAN T5 to IPVPN MPLS

When we change the encapsulation from VXLAN to MPLS, you can use the commonly deployed IPVPN service on the MPLS network instead of EVPN to advertise the IP prefixes.

In that case, there is a clear demarcation at the border-leaf device where EVPN T5 routes are converted into IPVPN routes. Option 6 is typically used when either the DCI solution is not requiring any L2 extensions between the sites or is used in combination with option 4 when just a subset of VLANs are L2 stretched between sites, and the rest of site to site connectivity is delivered using the IP prefix based reachability, where the type-5 VXLAN tunnels are stitched to MPLS LSP.

Why Stitch Tunnels Between Overlay Networks?

With the growing number of nodes in the data center fabrics, the NVO architectures bring many flexibilities in terms of L2 extensions, thanks to the dynamic (EVPN-based) establishment of the VXLAN tunnels but at the same time, the dynamic nature of BGP EVPN may, in larger scale deployments, create a massive number of overlay tunnels. When the EVPN-VXLAN fabric from site-A gets connected to multiple sites running EVPN-VXLAN fabrics, the BGP EVPN-signalized mesh of tunnels can become more significant and harder to manage from the operational perspective.

That's why some form of tunnel establishment control between PoDs or sites is appropriate in larger scale deployments, typically where multiple DC sites have more than 64 fabric leaves and when the L2 or L3 NVO extension is required in parallel between all the sites.

In fact, the new seamless DCI EVPN-VXLAN to EVPN-VXLAN stitching techniques and EVPN-VXLAN to EVPN-MPLS are defined within RFC9014. They are allowing to efficiently reduce the number of tunnels between the data center sites but also to selectively control which broadcast domains are really stretched. Enabling the seamless connectivity at the interconnect-gateway, also known as the *iGW level*, within the existing border-leaf or border-spine, is also easier because the technology involved in stitching can be implemented inside the existing MAC-VRF of the given tenant.

Besides the controlled scale of the tunnels, the seamless tunnel stitching techniques can help in easier L2 extensions between small and larger DC sites. Remote pair of border-leaf devices will reduce the level of flooding when a subset of bridge domains gets extended.

When enabling the stitching at the spines or at the border-leaf of the given DC fabric, you can unify the scaling of the fabric interconnect, where the larger fabric scale is not impacting the smaller fabric.

From the virtualization point of view the new seamless DCI techniques also offer a method for additional VLAN-translation between the point-of-delivery within the DC, or between the DC sites, by introducing the translation-VNI capabilities between DC and DCI.

The site origin identification using the interconnect ESI, aka iESI, can help track the workloads mobility between the sites and apply specific policy regarding the acceptance of the workloads tagged with specific ESI value – using Junos advanced EVPN route policy-statements – aka EVPN route-maps.

Another important reason for tunnel stitching techniques is the fact that two transport techniques, for example VXLAN and MPLS, can be merged under the same control plane and enable the connection seamlessly between these domains inside the same virtual context. It helps control which type of workloads are interconnected between the data center sites, but it also simplifies the operational aspect of the service delivery. For example, a given service can have restricted operations in just one data center but is enabled in a dedicated separate MAC-VRF and the other service is enabled with a MAC-VRF having a multi-site profile as well.

Let's summarize the logic behind the new seamless DCI techniques for NVO architectures:

- Simplified and unified implementation of DCI Layer2 and Layer3 extensions.
- Better scaling of the control-plane (less EVPN routes between the domains) and data-plane (lower number of VXLAN tunnels).
- Efficient multicast flooding between sites or PoDs .
- New VNI translation options – customer VNI and provider VNI option.
- Different transport techniques seamless integration (VXLAN-to-VXLAN, VXLAN-to-MPLS).
- Local significance of control plane and data plane values (RD, RT, ESI, VNI…) limiting impact if configuration error is made on one site.

Scaling and Seamless EVPN-VXLAN Stitching

The seamless stitching interconnect GWs enabled at the border-spine or border-leaf level helps in unifying and reducing the scaling between the DC sites or PoDs for the following network performance KPIs:

- Flooding and unicast next hops: selected VNIs are extended, the border overlay next hops are used to reach the other DC site.
- VXLAN tunnels: interconnect tunnels used instead of full mesh VXLAN approach.

- RIB/FIB perspective: the leaf in each DC or in each PoD is not installing all the EVPN routes from the site to which it's not directly connected, efficiently reducing the RIB and in consequence the FIB utilization

This is visualized in Figure 2.8 where a data center was deployed with two type of PoDs; a larger one with many servers, and then after some time, a smaller PoD is connected with a lower number of servers to offer additional redundancy just for the most critical services.

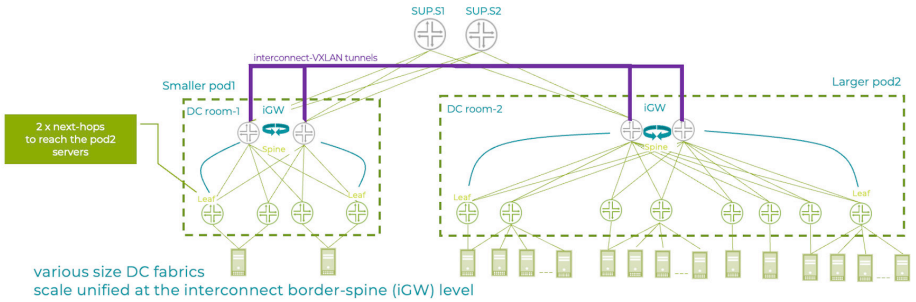


Figure 2.8 *Scaling and Seamless EVPN-VXLAN Stitching*

In this scenario, even if the larger PoD2 continues connecting the servers, the smaller PoD1 is not directly impacted. The leaf in the smaller PoD keeps consistent scaling from the unicast, flood next hop, or VXLAN tunnel perspective, highly reducing the TCAM utilization. The outcome of this approach is mainly around the reduced number of leaf scaling requirements thanks to the stitching happening at the higher end interconnect gateway level.

Main Use Cases for Seamless Tunnel Stitching

The main use cases for seamless EVPN-VXLAN stitching techniques are highlighted next and are mainly around scaling optimization for larger infrastructure for DC multi-PoD and DCI, but it can be extended to the multi-EVI use case offering virtual slicing of the fabrics. The other important use case here is the operational aspect of building the DC fabric using smaller PoDs and attaching them in steps to the fabric while building larger scale data centers with L2 stretch capabilities.

Another use case is the need of control plane demarcation in order to minimize the blast radius in case of problem on a datacenter or PoD.

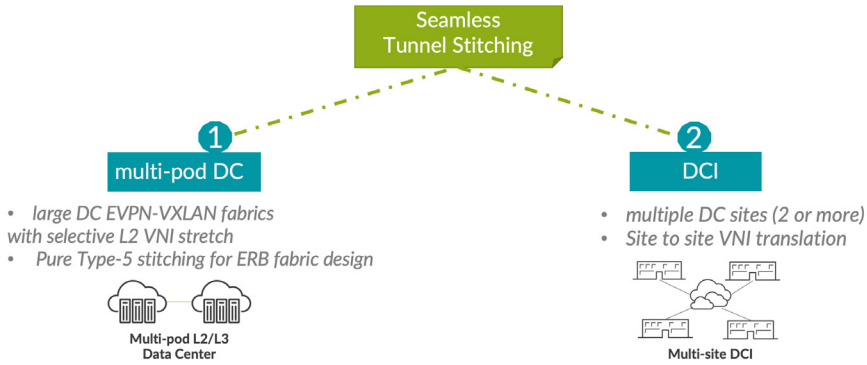


Figure 2.9 *Seamless Tunnel Stitching*

Use Case 1: Multi-PoD DC

This use case is for larger IP Clos fabrics with EVPN-VXLAN ERB, CRB or BO design, where the number of nodes is growing over time and where the total number of leaves in the fabric is between 240 and 480. The scaling of tunnels and operations in the fabric can be divided in eight PoDs respectively, using 62 leaf devices per PoD. Every pair of spines is enabled with the seamless stitching techniques and is terminating given PoD tunnels while originating the inter-PoD specific new tunnels to maintain the L2 broadcast domains between the point of delivery. The 248 * leaf devices use case with seamless stitching at the border-spine is illustrated Figure 2.10.

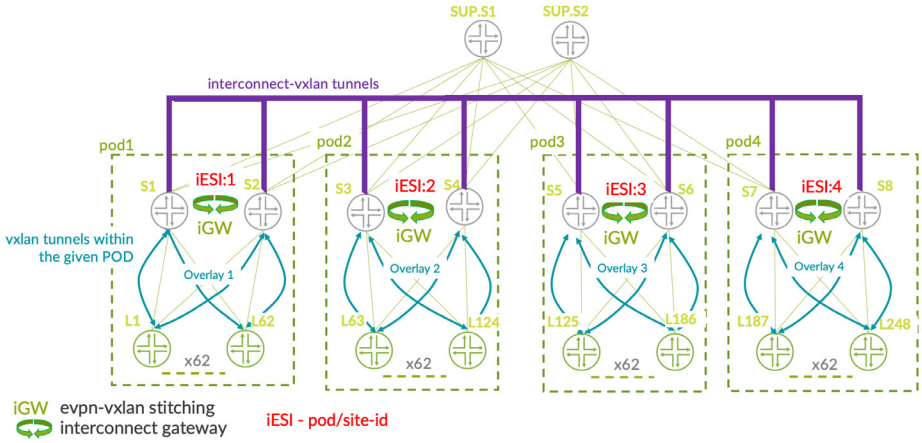


Figure 2.10 *Inter-PoD DC Fabric with 240 Leaves Using Seamless EVPN-VXLAN Tunnel Stitching at the Border-Spines*

In the NVO seamless tunnel stitching solution used in the multiPoD architecture shown in Figure 2.10, the leaf devices from one PoD to the other PoD of the same fabric are not establishing any direct VXLAN tunnel although they all can share the same broadcast domain for the given tenant network. This is achieved thanks to the Interconnect Gateway, the iGW spines role, at every point of distribution – PoD. This use case will be explained in more detail in chapter 4 where the seamless stitching at the iGW is incorporated at the border-leaf or super-spine level instead of the spines.

Use Case 2: Data Center Interconnect (DCI)

Another key seamless stitching use case is the data center interconnect (DCI) where, similarly to Figure 2.10, the full-mesh tunnels between sites are not used directly from the leaf in site-A to leaf in site-B but the tunnel is terminated at the iGW device and a new interconnect tunnel is crossing an intermediate IP domain in order to reach the remote site iGW, sharing the same interconnect instance. This use case is illustrated in the next Figure 2.11, where L5/L6 in site-A are terminating the local DC tunnels and originating new tunnels to connect across the existing IP domains to the site-B border-leaf. Site-B border-leaves terminate interconnect tunnels and originated new local site-B tunnels. This optimization mechanisms also happens seamlessly using standardized BGP EVPN signalization.

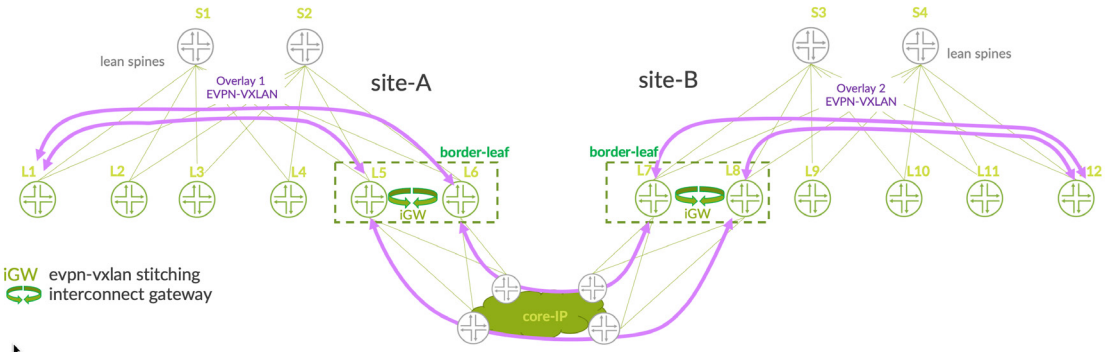


Figure 2.11 Use-Case Nr 2 – Seamless DCI Using NVO VXLAN Tunnel Stitching

As you can see, the DCI use case two main scenarios are possible when it comes to tunnel stitching:

- Scenario 1: Stitching from VXLAN LAN domain in Overlay 1 to VXLAN interconnect red highlighted in the above diagram
- Scenario 2: Stitching from VXLAN LAN domain in Overlay 1 to MPLS interconnect LSPs

In scenario 2, the border-leaf nodes connect directly to the MPLS backbone and become PE nodes. They are connected directly to remote site PE or connected via an existing P node to the MPLS domain for the reachability to remote sites.

Even if the main use case for the stitching of tunnels described in RFC9014 is for the Ethernet Layer 2, the tunnel stitching for the IP Layer 3 part in EVPN Type-5 VXLAN and EVPN Type-5 MPLS is also important and contributes to seamless introduction of IP services.

DCI and Security

Security can be enabled at different layers of the data center interconnection.

MACSec at the Border Leaf

If the border leaves have the hardware capacity, it is possible to encrypt all traffic by enabling MACSec. MACsec provides point-to-point security on Ethernet links and it is defined by the IEEE standard 802.1AE. MACSec provides data integrity to ensure that the traffic was not compromised while traversing the link and MACSec also provides encryption (see Figure 2.12).

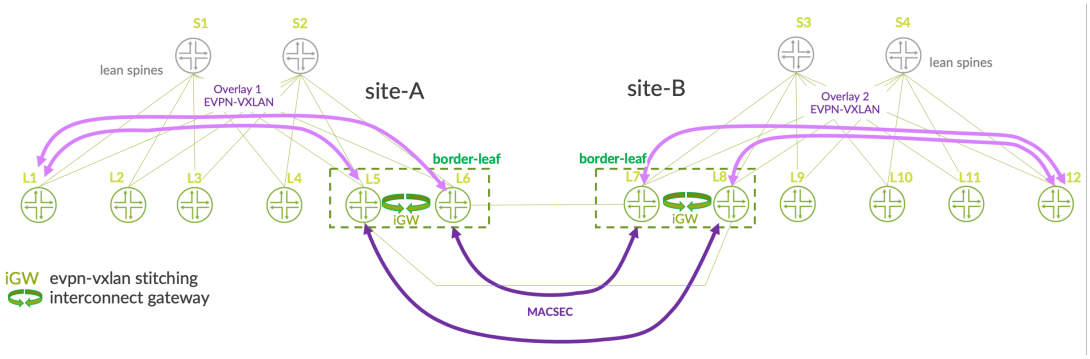


Figure 2.12 MACSec Encryption on the DCI Links

The advantage of MACSec over IPSec is that this encryption can be done directly by the border leaf switch at wire-speed. For example, in the Juniper portfolio, we offer QFX and PTX devices with line-rate MACSec encryption.

On the other hand, MACSec is limited to use cases where data center interconnection uses a direct link.

Secure EVPN with IPSec

Another way to encrypt traffic between datacenters is with IPSec. IPSec tunnels can be established over L3 networks and match more DCI use cases than MACSec.

If the encryption is done at the border of the data center, due to the need of high bandwidth between data centers, a high-end firewall will have to be deployed.

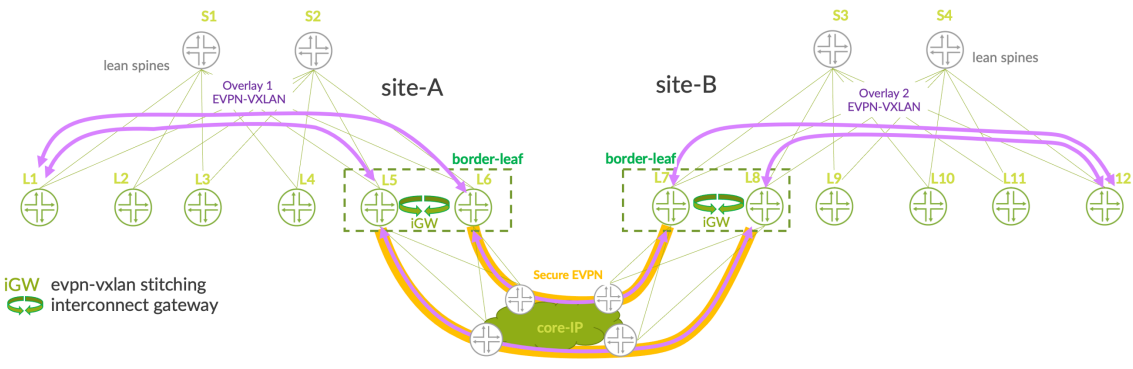


Figure 2.13 *Border leaf Doing Both EVPN VXLAN and IPSec*

This means the FW must be able to have both IPSec and EVPN VXLAN features. If that's not the case, then two separate devices will be needed: a border leaf dedicated to EVPN VXLAN and a firewall dedicated to IPSEC.

Instead of using IKEv2 for key exchange, the new draf-sajassi-bess-secure-EVPN describes a method that uses BGP for IPSec tunnels establishment, this method will also provide privacy, integrity and authentication.

It also answers the following DC requirements:

- Protect L2 and L3 tenant's data and control traffic.
- All tenant's traffic must be protected by IPSec: unicast, unknown unicast, broadcast, and multicast.
- BGP P2MP signaling for setup and maintenance.
- Granularity of Security Association Tunnels : per PE, per tenant, per subnet, per L3 flow, per L2 flow, per AC pair.
- Support single or multiple policies and DH groups for all SAs.

Control Plane Protocol Security

The EVPN control plane is based on BGP, consequently, securing the BGP peering is key. The newest mechanism is TCP-AO and this obsoletes TCP MD5, as defined in RFC5925.

TCP-AO is an authentication protocol for TCP-based routing protocols and can be used for other TCP-based protocols as well. Unlike IPsec, TCP-AO does not support encryption.

TCP-AO provides the following features:

- Rekeying during a TCP connection.
- Automatic replay protection for long-lived connections.
- Per-connection traffic keys as unique as the TCP connection itself.

Group Based Policy (GBP) and DCI

Group based policy provides micro and macro-segmentation for EVPN VXLAN fabrics. Hosts or group of hosts are assigned an SGT (Security Group Tag) either manually or with a radius server. A security policy based on security group tags is configured on the leaves to perform micro and macro-segmentation.

As described in the previous chapter, draft-smith-VXLAN-group-policy leverages VXLAN header to assign a group identifier. The identifier of the source host is carried in the VXLAN encapsulated packet and used at the egress leaf to apply the security policy.

For GBP to be useable in the DCI context, the SGT must remain in the VXLAN header while crossing DCI links. This cannot be achieved in the VLAN hand-off use case as the VXLAN header is removed. It is naturally done in the OTT implementation as the VXLAN header is forwarded without any change.

In the context of seamless EVPN VXLAN stitching, the gateway must copy the SGT value to the newly VXLAN interconnect packet.

Seamless EVPN VXLAN to MPLS stitching is not natively compatible with SGT because the VXLAN header is removed by the gateway and replaced by a MPLS header. A BGP community could be used to advertise the SGT information to the remote site.

Inline Firewall for DCI

An inline firewall in the path of each DCI link can bring L4 to L7 security. For example, Juniper SRX firewalls do VXLAN tunnel inspection providing the following features:

- Stateful inspection
- L4/L7 application filtering

- IDP / IDS / IPS
- UTM

Let's review two use cases here.

Use case 1: DCI is over the top (OTT) and both sites share the same EVPN VXLAN control plane. In that case, the firewall inspects VXLAN encapsulated traffic. See Figure 2.14.

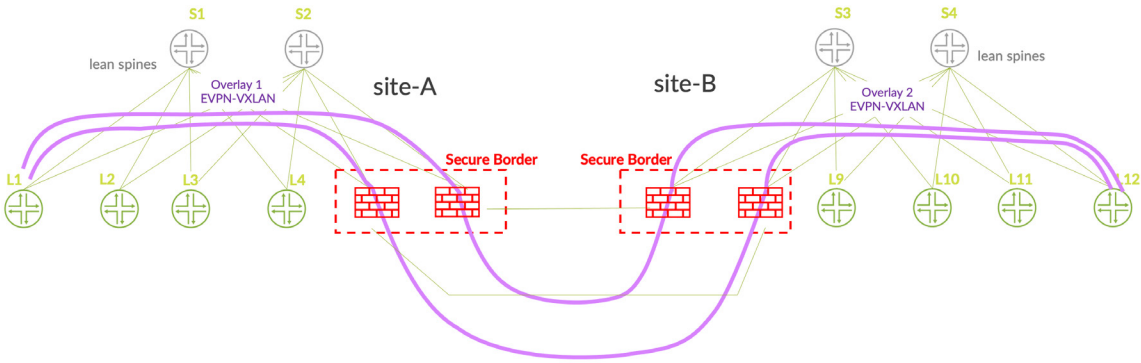


Figure 2.14 *Secure Border Using Inline Firewall with OTT DCI*

Use case 2: a border leaf performs DCI EVPN seamless stitching and a firewall is added between the border leaves to provide L4-L7 security. In that case as well, the firewall inspects VXLAN encapsulated traffic and, moreover, you benefit from the seamless stitching feature provided by the border leaf. See Figure 2.15.

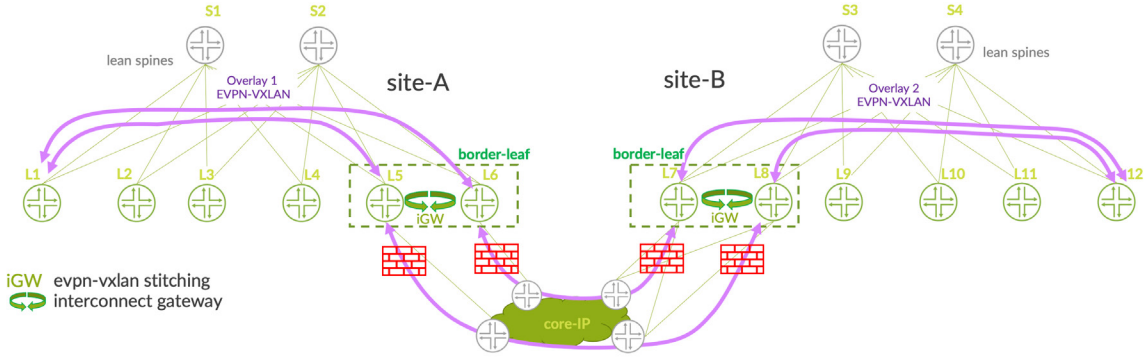


Figure 2.15 *Seamless Stitching and Inline FW for DCI Traffic*

Service Chaining to a Firewall

EAST-WEST Traffic

When routing between T5-VRFs, it is interesting to provide advanced security features like AppFW, IDP/IDS/IPS, URL Filtering, etc.

To provide this service chaining-like feature, the firewall cluster is physically connected to the border leaves. A routing protocol like eBGP is recommended between each security zone towards its associated T5-VRF. The FW advertises a default route to the VRF in order to filter all the traffic egressing each VRF. See Figure 2.16.

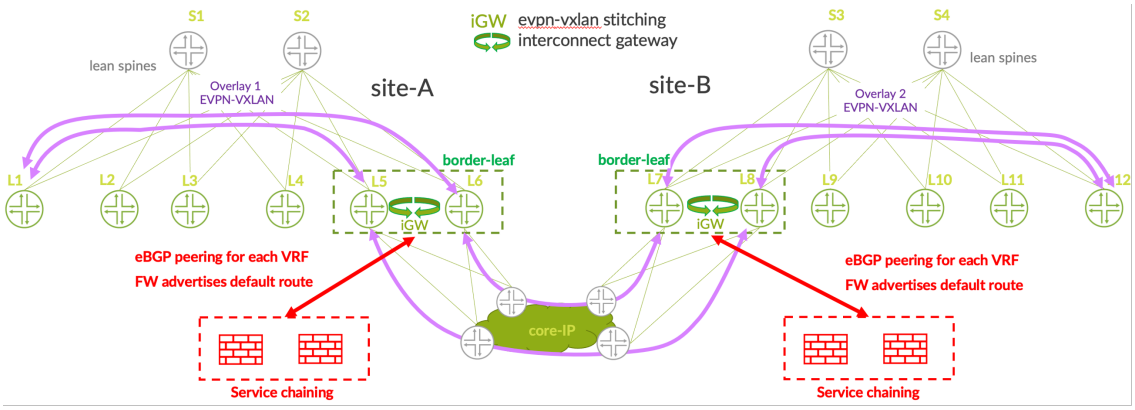


Figure 2.16 Service Chaining to a Firewall for East-West Traffic

NORTH-SOUTH Traffic

When traffic ingress data centers from remote locations, it often requires to provide advanced security features with L4-L7 firewalls. This firewall service must be redundant as well. It means that it is deployed in each data center and that traffic must be symmetrically handled.

In order to guarantee symmetrical routing, there are several configuration options.

Option 1: active / passive implementation

All traffic ingresses and egresses the same data center. This also means sub-optimal routing for the passive site where all traffic of the data center ingresses and egresses the other data center. The requirement is to have enough bandwidth between the data centers and to stretch all L2 and L3 services between the data centers. The default route advertised by the active firewall cluster is preferred over the passive cluster.

Option 2: use of host routes.

For ingress traffic, the fabric advertises /32 routes to the firewalls and the firewalls advertise those /32 routes to the remote sites. This way, ingress traffic is redirected to the appropriate firewall of remote locations. If the number of hosts is very important, this option can be difficult to implement due to RIB/FIB limitation on firewalls and re- mote devices. For egress traffic, each FW is the default route for the local data center..

Option 3: load balance VRFs between data centers.

In that case, some VLANs are active in a data center and backup in the other one. For ingress traffic, firewalls advertise active networks to the remote sites. For egress traffic, firewalls advertise a default route in the local data center and a BGP community can be set to influence routing in the appropriate data center.

Chapter 3

Deep Dive into EVPN Seamless Stitching

Many aspects related to tunnel stitching were designed to be done automatically via BGP EVPN signalization. However, it's always good to know exactly what's happening underneath from the theoretical point of view as well as for mindful design and implementation of more complex scenarios with many DC sites involved.

In this chapter, we will review in more details the control-plane and data plane aspects involved in the seamless EVPN-VXLAN to EVPN-VXLAN tunnel stitching as well as EVPN-VXLAN to EVPN-MPLS.

Control Plane and Data Plane for Seamless T2 Tunnel Stitching – VXLAN to VXLAN

When it comes to the EVPN-VXLAN to EVPN-VXLAN tunnel stitching, the interconnect domain-2 shown in Figure 3.1 can be pre-established at the border-leaf so the VXLAN tunnels between the border-leaf1 in site-A and border-leaf3 in site-B may already have the tunnels in place even before leaf1 and leaf7 in each site get deployed.

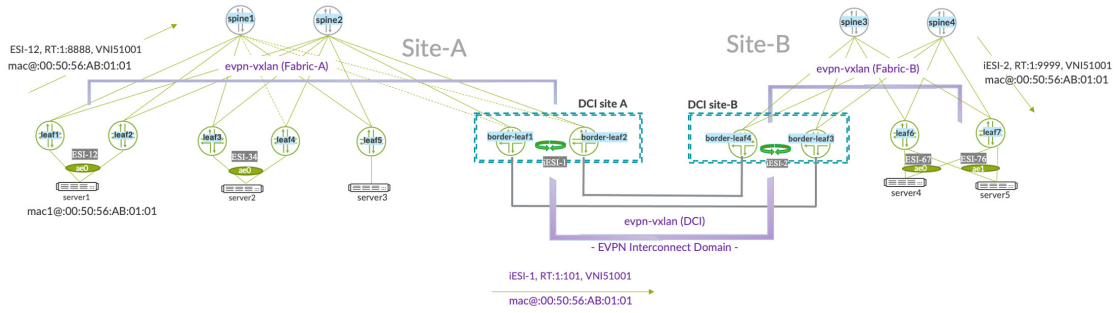


Figure 3.1 *Reference Topology for VXLAN-VXLAN to EVPN-VXLAN Seamless Stitching*

Figure 3.2 is a briefing on how the mac1@: 00:50:56:AB:01:01 originated in site-A at leaf1 is advertised from left to right, crossing spine, then the border-leaf1 and border-leaf3 which are here enabled with the VXLAN to VXLAN tunnel stitching and then finally received at the leaf7.

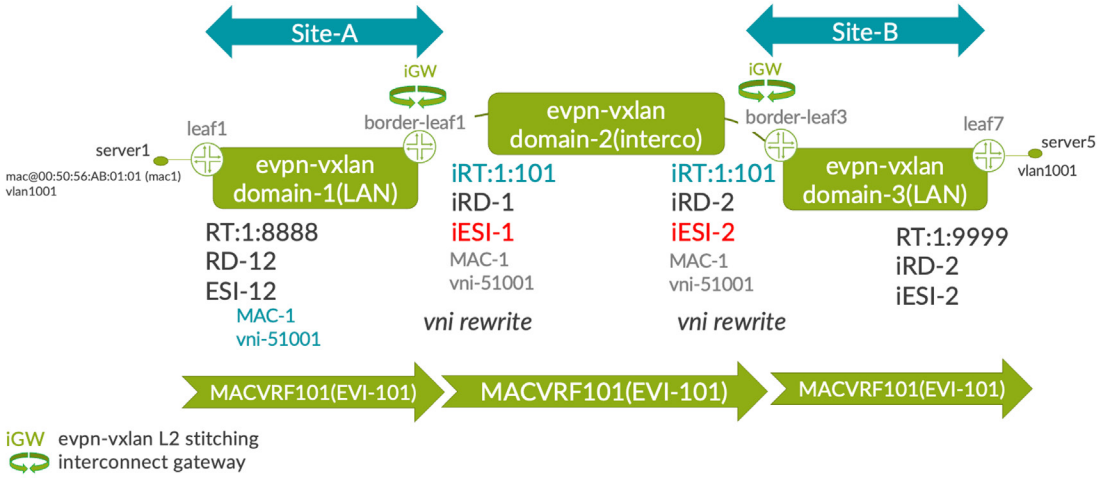


Figure 3.2 Seamless EVPN-VXLAN T2 Tunnel Stitching - Summary of Control Plane Steps

Based on the automatic translation of EVPN routes Type-1, Type-3, Type-4, and Type-2 for the mac@1 at border-leaf1 in site-A, the border-leaf3 will establish a VXLAN tunnel to border-leaf1. The translation of route-target, route-distinguisher, and ESI value will happen seamlessly for the EVPN routes at the border-leaf.

The original EVPN route received from leaf1 are not used at the remote DC sites because the border-leaf is advertising new EVPN routes for the L2 VNI segments and MAC addresses learned from the local fabric leaf nodes.

The same operation of EVPN route translation will continue to happen for mac@1 from the example topology at the border-leaf3 in site-B, where before advertising the Type-2 route to leaf7, it will rewrite the EVPN route attributes, such as route-target to the site-B local site target:1:9999, the protocol next-hop to its local IP@, the route distinguisher to its local value, and the ESI value to its local site-B iESI-2.

Based on that control plane information, leaf7 will establish a tunnel to its local site border-leaf3, instead of an over-the-top direct tunnel to leaf1. This is something that can be depicted on the block level diagram (Figure 3.2), where VLAN1001 is extended between the site-A and site-B using VNI51001.

By using the EVPN route translation at border-leaf1/3, the tunnels between the leaf nodes in two different DC sites are never directly established and are always using the IP

next hops of local site border-leaf to reach the MAC addresses from the remote data center site. This is significant improvement for larger scale deployments with many leaf nodes, as it helps to control the number of route processing cycles at the leaf nodes and reduces the number of next hops installed in the PFE (only the local border-leaf IP next hop will be installed for remote sites mac-addresses reachability).

The border leaf level VNI rewrite operation can be useful in case the operator of the data centers would like to have a different provider VNI value for data center interconnect purposes whilst preserving the same bridging and same broad- cast domain end-to-end from the forwarding point of view. In case this is enabled at the configuration level, a new VNI value is advertised by the border-leaf for the MAC@ learned from the leaf.

The translations that occur at the border leaf are seamless and don't require explicit policies.

When it comes to the control plane aspects and Junos/Junos Evo, Figure 3.3's block diagram highlights the sequence in which the logical tunnels are created, based on the received EVPN control plane information.

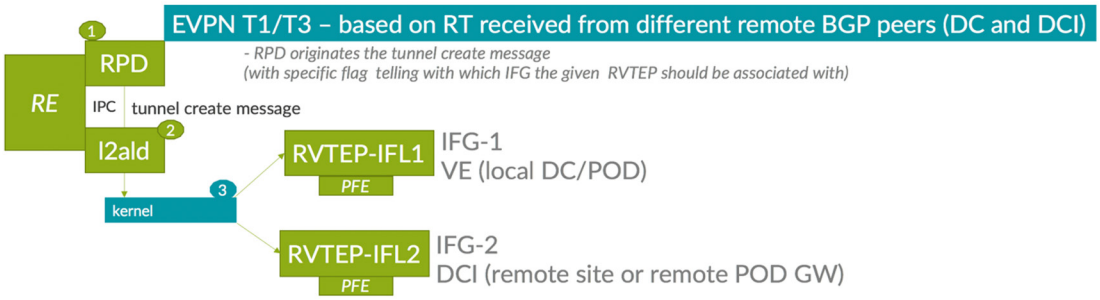


Figure 3.3 *Junos/Junos Evo L3 Routing and L2 Bridging Daemon Interaction for the VXLAN Tunnel Creation*

The first step of the state machine for the VXLAN bridging tunnel creation involves processing of the EVPN route at the RPD routing daemon level, which then triggers the 'tunnel create' message sent to the Layer 2 bridging daemon – l2ald, which then sends, via the kernel, a call for the local fabric tunnel creation, DCI tunnel creation and installation of the flooding mesh-groups at the PFE level, which is needed for the processing of BUM traffic.

Because the border-leaf EVI is processing the routes from the local fabric as well as routes from the remote site, the RT-1/RT-2/RT-3 coming from two different domains will contribute to the creation of the LAN VXLAN tunnels as well as the DCI VXLAN tunnels. This is something we will explain in more detail in the next chapter.

When stitching from one VXLAN tunnel to another at the border-leaf, the operation of decapsulating and encapsulating packets is taking place at the PFE level. The data plane part of the stitching also involves handling the broadcast and multicast packets when dealing with L2 DCI to maintain the loop-free ethernet solution. In Junos/Junos Evo this is implemented using the concept of mesh-groups, associated LAN, WAN tunnels and local node interfaces.

Figure 3.4 reveals the way remote site and local site VXLAN tunneling gets associated with different mesh-groups aka interface-groups (IFG) to respect the split-horizon rule, which is not allowing to send back the same BUM traffic to the interface on which it was originally received.

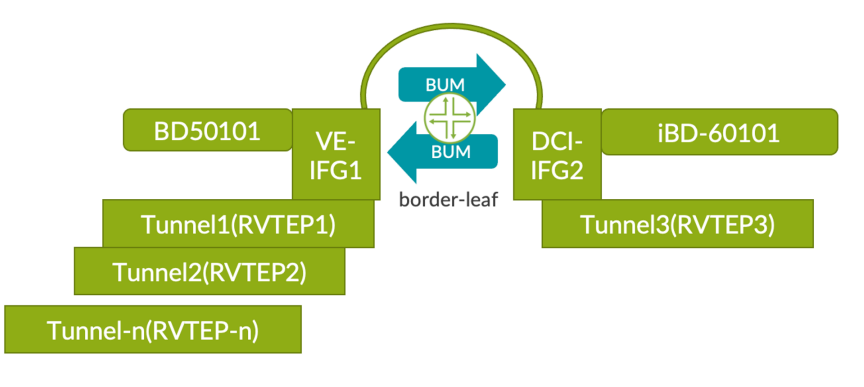


Figure 3.4 *Layer 2 Data-Plane and Tunnel Stitching – Tunnels to Mesh-Groups Mapping*

For mesh groups for the given bridged-domain, a flood-route is installed, which points to a composite next hop which has all the interface next hops where the packet needs to be flooded. For each VLAN/VNI, a mesh group has different flood NH groups and a flood route. The concept of mesh groups is specific to Layer 2 traffic and is not used in the case of pure IP DCI solutions (DCI option-5/option-6).

The mesh-group installation per MAC-VRF and per VLAN name can be verified at the QFX5130 and QFX5700 border-leaf level using the following command.

Verification 1: flood mesh-group next-hops at MAC-VRF / VLAN-name level

```
root@border-leaf3> show mac-vrf forwarding flood instance MACVRF101 VLAN-name VLAN1001
Name: MACVRF101
CEs: 0
VEs: 3
VLAN Name: VLAN1001
Flood Routes:
  Prefix    Type        Owner             NhType        NhIndex
  0x30004/51 FLOOD_GRP_COMP_NH __ves__        ulst          57068  >> where to flood when BUM comes
from Fabric
  0x30006/51 FLOOD_GRP_COMP_NH __wan_flood__    ulst          57067 >> where to flood when BUM comes
```

```
via DCI
  0x40002/51 FLOOD_GRP_COMP_NH __re_flood__     comp          57021
root@border-leaf3>
```

We can then also check which physical interface is used for the given mesh-group name next hop. For example, here we show how to verify the outgoing DCI flood interface at the QFX5130 border-leaf3 level. The composite NH index 57067 learnt from previous command for the L2 flooding mesh-group called __wan_flood__ points to the DCI interface et-0/0/26 at border-leaf3 – used to connect to DC A from DC B.

Verification 2: mesh-group recursive next-hop lookup

```
root@border-leaf3> start shell
[VRF:none] root@border-leaf3:~#
[VRF:none] root@border-leaf3:~# cli-pfe
root@border-leaf3:pfe>
root@border-leaf3:pfe> show evo-pfemand nh-recursive index 57067 >>  where to flood when traffic comes
from DCI
57067(unilist, Protocol:vpls, Ifl:0 .local..0, Token:0)
 57059(indirect, Protocol:vpls, Ifl:4294967295 N/A, Token:0)
  57055(composite, Protocol:vpls, Ifl:4294967295 N/A, Token:0)
   57019(composite, Protocol:vpls, Ifl:4294967295 N/A, Token:0)
    57016(composite, Protocol:vpls, Ifl:4294967295 N/A, Token:132770)
     332498(indirect, Protocol:ipv4, Ifl:1031 et-0/0/26.0, Token:100002)
      22001(software, Protocol:ipv4, Ifl:1031 et-0/0/26.0, Token:100002)
       55002(unicast, Protocol:ipv4, Ifl:1031 et-0/0/26.0, Token:100002)
    57017(composite, Protocol:vpls, Ifl:4294967295 N/A, Token:132770)
     332497(indirect, Protocol:ipv4, Ifl:1031 et-0/0/26.0, Token:100002)
      22001(software, Protocol:ipv4, Ifl:1031 et-0/0/26.0, Token:100002)
       55002(unicast, Protocol:ipv4, Ifl:1031 et-0/0/26.0, Token:100002)
 57063(indirect, Protocol:vpls, Ifl:4294967295 N/A, Token:104099)
  57011(discard, Protocol:vpls, Ifl:4294967295 N/A, Token:104099)
root@border-leaf3:pfe>
root@border-leaf3:pfe> show evo-pfemand nh-recursive index 57068 >> DCI interfaces to flood when BUM
comes from fabric
57068(unilist, Protocol:vpls, Ifl:0 .local..0, Token:0)
 57060(indirect, Protocol:vpls, Ifl:4294967295 N/A, Token:0)
  57056(composite, Protocol:vpls, Ifl:4294967295 N/A, Token:0)
   57052(composite, Protocol:vpls, Ifl:4294967295 N/A, Token:0)
    57050(composite, Protocol:vpls, Ifl:4294967295 N/A, Token:132771)
     1522216(indirect, Protocol:ipv4, Ifl:1030 et-0/0/24.0, Token:100003)
      22003(software, Protocol:ipv4, Ifl:1030 et-0/0/24.0, Token:100003)
       55003(unicast, Protocol:ipv4, Ifl:1030 et-0/0/24.0, Token:100003)
 57064(indirect, Protocol:vpls, Ifl:4294967295 N/A, Token:104099)
  57011(discard, Protocol:vpls, Ifl:4294967295 N/A, Token:104099)
root@border-leaf3:pfe>
```

At the unicast data plane level, when the VXLAN packet is received at the border-leaf, it gets decapsulated locally and sent either to the local server port (for example when there are any servers or appliance locally connected to the border) or it is decapsulated from ingress VNI and encapsulated in the new DCI VXLAN tunnel using the same or a new VNI value (if VNI translation is set) inside the VXLAN tunnel header.

Control Plane and Data Plane for Seamless T5 Tunnel Stitching – VXLAN to VXLAN

In some cases, the DCI solution may not have a requirement for the L2 VLAN stretching, and the site-to-site reachability is required only between the IP prefixes or an ERB EVPN-VXLAN design was deployed inside the fabric. In this case DC site A and site B border-leaf nodes can terminate the pure Type-5 LAN tunnels and originate a new Type-5 DCI tunnel, rewriting the RMAC (router-mac) but conserving the original routing VNI value. Similarly, to the L2 seamless stitching, in this case we also offload the leaf from the significant number of tunnels. Because in case of pure Type-5 EVPN-VXLAN there's no ESI used then, the overlay IP ECMP (hierarchi- cal IP ECMP) will be used for load balancing of the traffic from remote to local site – the same prefix will be reachable via two Type-5 tunnels from boder-leaf3 in DC site B – one towards boder-leaf1 and other to border-leaf2. For example, because IRB.10 and IRB.20 subnets are only enabled in DC A, then they will be advertised towards the remote DC B only as IP prefixes using EVPN route type-5. The MAC/MAC-IP route-type 2 informa- tion won't be advertised to the remote sites for the site specific VLANs.

This is something illustrated in Figure 3.5's diagram on where the RMAC and routing-VNI gets re-originated for the prefixes unique for the given DC site.

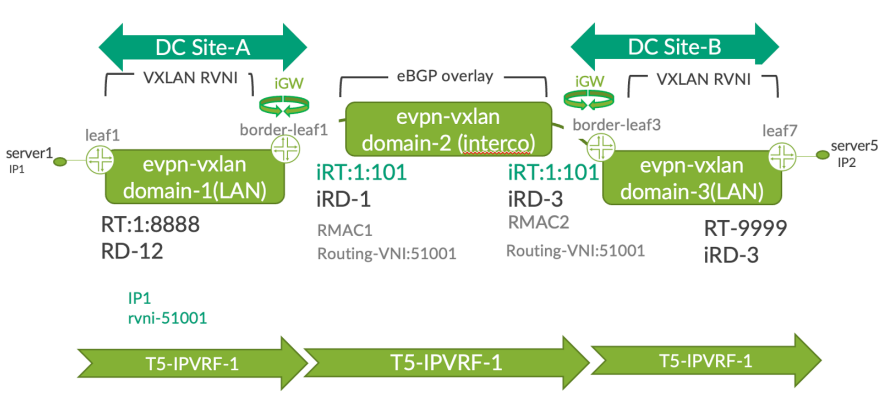


Figure 3.5 *Pure Type-5 Seamless Tunnel Stitching - Block Diagram*

You can see in Figure 3.5 that the iESI in the case of pure Type-5 to Type-5 stitching is not used for load balancing purposes and only the overlay IP ECMP will be used to load balance the traffic destined to the given DC site. The original router-mac (RMAC) from leaf1, the route-distinguisher, and the route-target are changed at the border-leaf level but the routing VNI info is by default preserved, copied from the original EVPN T5 route. The Figure 3.5 is illustrating the pure Type-5 scenario, aka *interface-less*, where there're no need to also advertise the Type-2 MAC routes for load-balancing purposes. The interface-less T5 mode is in fact the most popular industry implementation of EVPN prefix-advertisement standard – RFC9136.

When it comes to Junos and Junos Evo implementation of Type-5 stitching, Figurer 3.6 can help illustrate the interaction between the Kernel and routing daemons.

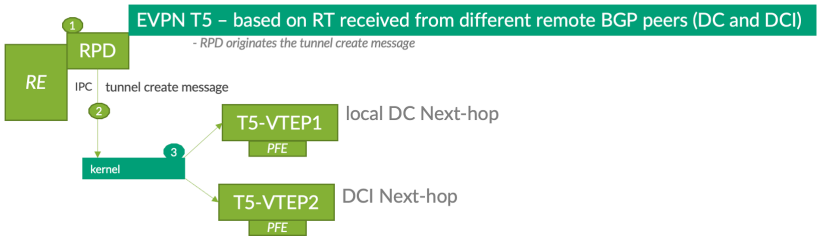


Figure 3.6 *Pure Type-5 VXLAN Tunnel Stitching - RPD Daemon And Kernel State Machine*

When comparing the block diagram for "Type-5 stitching" and the one for "type-2 stitching" you can observe that in the case of Type-5, stitching the local border-leaf will not create any additional logical interfaces, aka *IFLs*, and the L2 daemon called l2ald is not engaged in the creation of the Type-5 tunnels anymore.

Control Plane and Data Plane Operation for EVPN-VXLAN to EVPN-MPLS Stitching

When there's a requirement of connecting the border-leaf directly to the MPLS core network, instead of going through a dedicated external DC-GW, then the stitching from VXLAN to MPLS directly at the border becomes an attractive option. It reduces the number of nodes to manage and improves the site-to-site latency.

In this case, the VXLAN encap/decap takes place at the border-leaf on the LAN side and the MPLS encap/decap on the WAN side. An MPLS label associated with the given EVI (MAC-VRF) will be used before forwarding the packet over the LSP (Label Switched Path). The original VNI information from the LAN EVPN-VXLAN fabric side is replaced with the vlan-id info at the border-leaf (when using the VLAN-aware service-type, for example) and the remote leaf performs the similar operation of decapsulating the MPLS LSP with the vlan-id information remapped again into the local site VNI.

To simplify the discussion around data plane and control plane driven operations when dealing with VXLAN to MPLS, the following block level diagram in Figure 3.7 is shown where border-leaf1/border-leaf3 is enabled with the DCI tunnel stitching function. Similar to the previous example, the border-leaf will be terminating the local fabric VXLAN tunnels. However, instead of originating a VXLAN tunnel for interconnect purposes, it'll use an MPLS LSP, so the given MAC-VRF/EVI interconnect will allocate an MPLS label associated with that EVI/iESI. In Figurer 3.7, the border-leaf3 will be using the MPLS LSP with the MPLS label 101 to reach the server1 MAC@.

When using the iESI from DC site-A, the MPLS aliasing label will also be acting for load balancing purposes, when the given site has two or more border-leaf nodes.

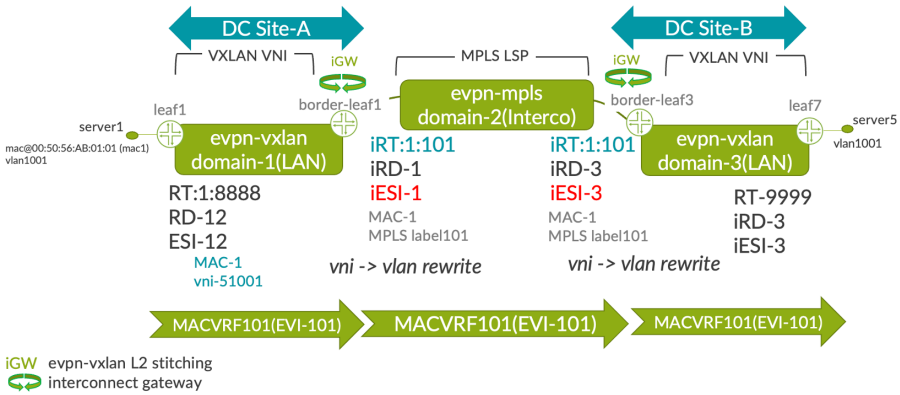


Figure 3.7 *Seamless EVPN-VXLAN to EVPN-MPLS Stitching - Data Plane and Control-Plane Operations*

The data plane operation highlighted above is possible thanks to the BGP EVPN control plane part where the LAN route-distinguishers (RD) and WAN (DCI) route-distinguishers (called *iRD*),

LAN Route-Target (RT) and WAN Route-Target (iRT) within the same EVI (MAC-VRF) are also part of the MAC-VRF configuration at the border-leaf in each DC site. The ESI change takes place at the border-leaf node, same as for the VXLAN-to-VXLAN stitching– all the LAN Fabric MAC@ will be represented on the remote site border-leaf via the site identifier iESI, which is specific to each site per EVI.

The border-leaf nodes will advertise type 1 EVPN routes (AD routes) per ESI and AD route per EVI to DC and DCI side. DC fabric wide RD and RTs (shared with server leaf nodes) and iRD and iRT for DCI (shared with remote site border-leaf nodes). In the context of VXLAN to MPLS, the AD route per EVI towards the DCI side is advertised with aliasing label, which will be used by remote border-leaf for load balancing of unicast traffic.

The IM route type-3 will also have the vlan-id instead of the VNI used toward leaf nodes on the fabric side. The type 4 route (ES-Import route) in the DCI context is advertised with the new I-ESI (interconnect ESI) and MPLS label, which is used for split-horizon function between multi-homing border-leaf nodes. The DF/nDF election is also done by default using the MOD algorithm but can be also set with hard coded preference-based values.

Handling at the data plane level of the BUM packets is consistent with the one described in the VXLAN-to-VXLAN section above, where VXLAN IFL interfaces and MPLS LSP are associated with specific mesh-group IDs for which the split-horizon rules will be enabled, to avoid L2 Ethernet flooding loops. In the section dedicated to implementation, we further break down the control plane and data plane aspects of EVPN-VXLAN to EVPN-MPLS stitching for bridging.

In some cases, the EVPN-VXLAN fabric is deployed in parallel of bridging part, mentioned before, also with the IP Prefix advertisement, with prefixes specific to the given DC site location. In this case, there's no need to leverage in parallel the bridging between the sites and only the full IP reachability is required between the DCs. This is something highlighted in Figure 3.8 where each DC fabric is deployed with ERB design and generates site specific prefixes (that are not used, also on a different DC site), which needs to be reachable between the sites and potentially reachable also from the DC users seating behind the core IP. In this case the border-leaf nodes in each fabric will simply advertise the IP prefixes received from distributed leaf nodes (like, for example, QFX5120 or QFX5130) as EVPN-VXLAN Type-5 routes but transform them to IPVPN-MPLS advertisements. This way any existing remote PE node in the IPVPN-MPLS network will get the reachability to the DC fabric. The border-leaf doesn't have to advertise all the EVPN Type-5 prefixes from the given site location and only a summary route for the given tenant will be sent to the rest of the PE nodes. This is something visualized in Figure 3.8.

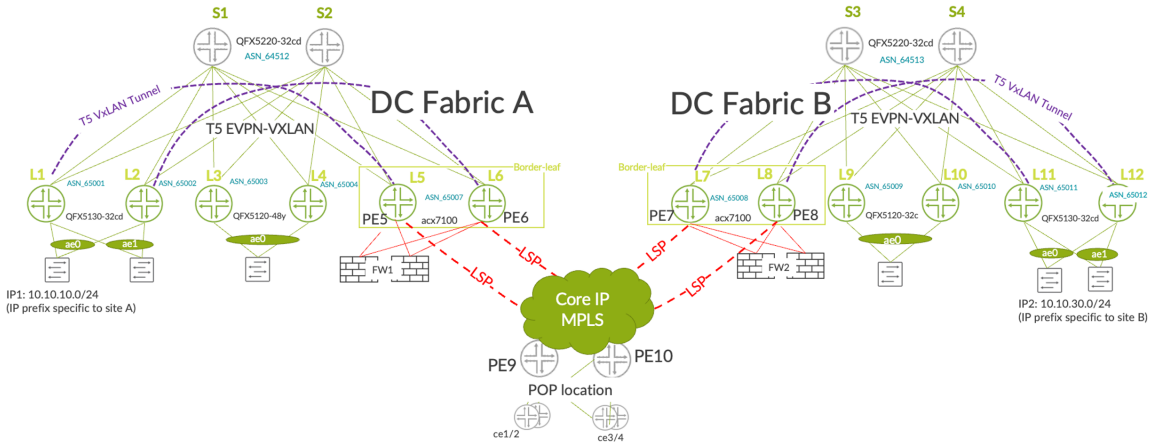


Figure 3.8 *EVPN-VXLAN Type-5 to IPVPN-MPLS Internetworking*

The deployment highlighted above is interesting when the organization already deployed MPLS in their core infrastructure and data centers services, in different locations, are reachable by the users using IPVPN-MPLS. In this case, in order to simplify the DCI deployment, same approach is taken and simply the site specific prefixes – here the

10.10.10.0/24 for site A and 10.10.30.0/24 for site B, are advertised from the border-leaf PE nodes to the IPVPN-MPLS existing infrastructure. So before they are reachable not only between the DC sites but also for all the existing POP locations for the given tenant/service id. The operation behind the stitching from VXLAN Type-5 tunnels to MPLS LSP is also explained in the draft-ietf-bess-EVPN-ipvpn-interworking document.

In order to protect the network from any IP routing loops, the domain-path, aka *D-PATH attribute*, specifically set for DC-to-DC communication can be optionally introduced.

The situation when part of the IP prefixes are site specific, and others are stretched between the two site,s may also happen. In this L2 DCI stretched scenario, the PoP (Point of Presence) locations for most optimal IP reachability will also get the host IP routes via the IPVPN in order to reach the right DC location directly in an optimal way.

Chapter 4

DCI and Multipod: Underlay Architecture Options

This chapter focuses on different underlay options for two or more PoDs or data center sites interconnection. Because of the number of sites to interconnect, due to the expected east-west oversubscription ratio or because of the existing site-to-site IP underlay routing, the decision on the type of underlay architecture will be made based on:

- Where exactly the seamless stitching will take place – at the border-spine or at the border-leaf level
- What type of underlay transport approach will be used – dark fiber, super-spines, or existing IP domain

Border-Spine Stitching with Super-Spines Underlay Transport

When the PoD-to-PoD workloads have higher requirement for the bandwidth – for example due to intensive data replications – a dedicated lean super-spine block may be required with 100G or 400G forwarding capacity.

Even if there is still core IP connectivity for end users to access the DC workloads, site-to-site replications or server clustering can use dedicated block of super-spines.

In the context of EVPN-VXLAN, we call this block *lean super-spines* as it's only delivering high bandwidth, IP forwarding, low latency, without any processing of EVPN routes or without any tunnel termination/origination.

The super-spine peers use eBGP with both PoD1/PoD2 spines to deliver full IP reachability between the S1/S2 in PoD1 and S3/S4 in PoD2. The route server role for overlay eBGP peerings is optional as spines from each PoD can simply establish a full mesh of overlay multihop eBGP peering between themselves. For more than two pods or two sites the iBGP full-mesh is however more convenient from provisioning point of view.

The seamless stitching of tunnels would still happen in that case at the spine level in each PoD. The leaves from PoD1 would not enable a direct tunnel to the leaf devices in PoD2. Host-1 to Host-2 provisioned within the same VNI 51001 would communicate via the interconnect gateways local to the PoD. In the example shown in Figurer 4.1, the vlan-id used at the leaf level is 1001 but, with the built-in translation capabilities, both VLAN and VNI can be different in each PoD, while still delivering the same L2 broadcast domain between the PoDs.

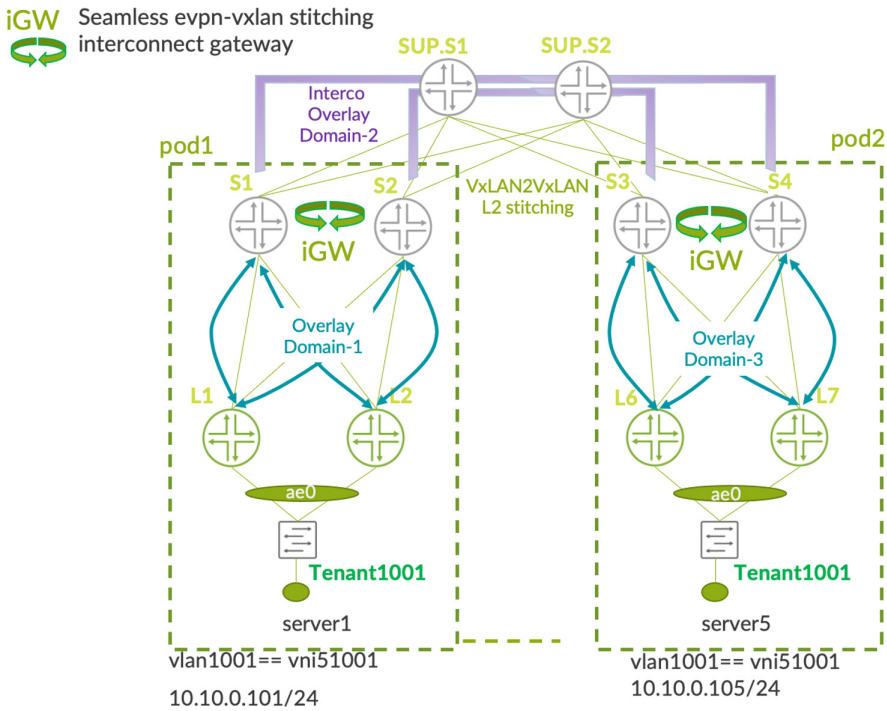


Figure 4.1 *Inter-PoD EVPN-VXLAN Stitching At Border-Spine With Super-Spine Underlay*

With the inter-PoD design shown in Figure 4.1, the super-spines usually use 100Ge/400Ge PoD-to-PoD connection for any low latency and data replications requirements. This design can still use the border-leaf block from each PoD for the end user access to the data via the WAN.

Border-Spine Stitching with Intermediate IP Domain Underlay Transport

Sometimes, when significant data is going north-south, and only a small amount of data gets replicated between the DC sites or PoDs, the spines are used directly as border nodes connecting the end users to the core-IP. At the same time, the edge routers from that core network are used for inter-PoD or DC site to site connectivity. With this design we also reduce one hop so the firewalls can be directly connected to the border spines in this scenario. See Figure 4.2.

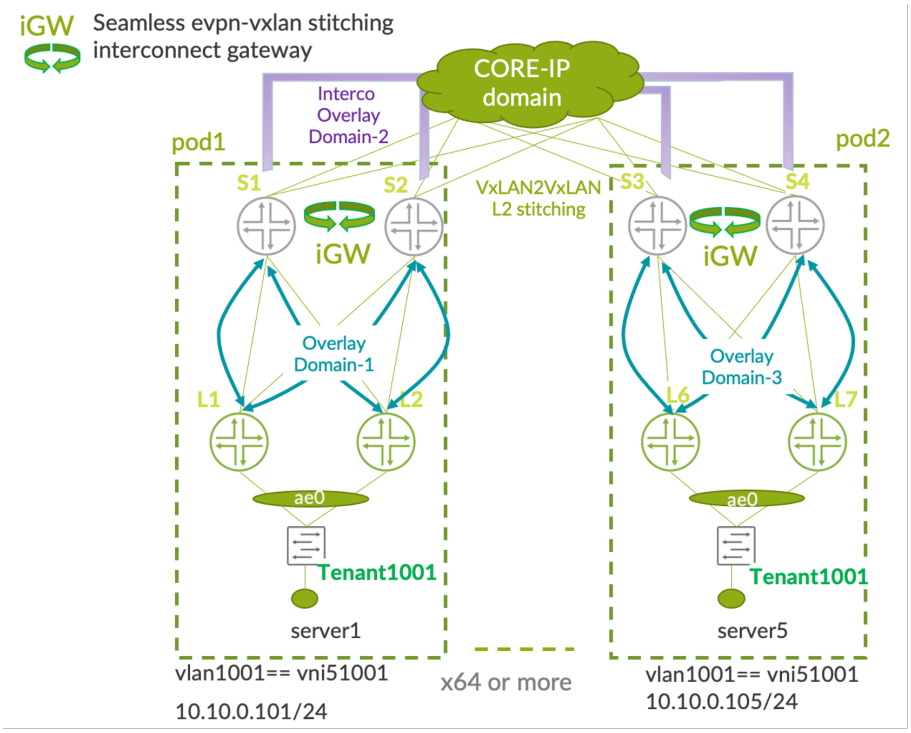


Figure 4.2 *Inter-PoD EVPN-VXLAN Stitching Using Intermediate IP Domain*

With the design shown In Figurer 4.2, you also skip adding a dedicated block of border-leaf nodes, which for significant north-south data volumes, would require higher number of leaf-spine 100Ge links overall in the fabric. Compared to the previous example, in this case you typically don't use any additional border-leaf block as everything is combined from PoD-to-PoD as well as core IP to the DC connect at the border-spine level. The intermediate domain 2 connected to the border-spines will be using the edge routers for IP forwarding purposes and all the overlays between the border-spine S1/S2 and S3/S4 will create a full mesh of eBGP multi-hop peerings.

Border-Spine Stitching with Back-To-Back Dark Fiber Transport

There are situations where the two PoDs are either in two different DC rooms, or the two DC buildings are interconnected using dark fiber. That's where the back-to-back connection in full mesh or partial mesh between the spines from each PoD can be used to help delivering fast data replications, without any super-spine layer. See Figure 4.3.

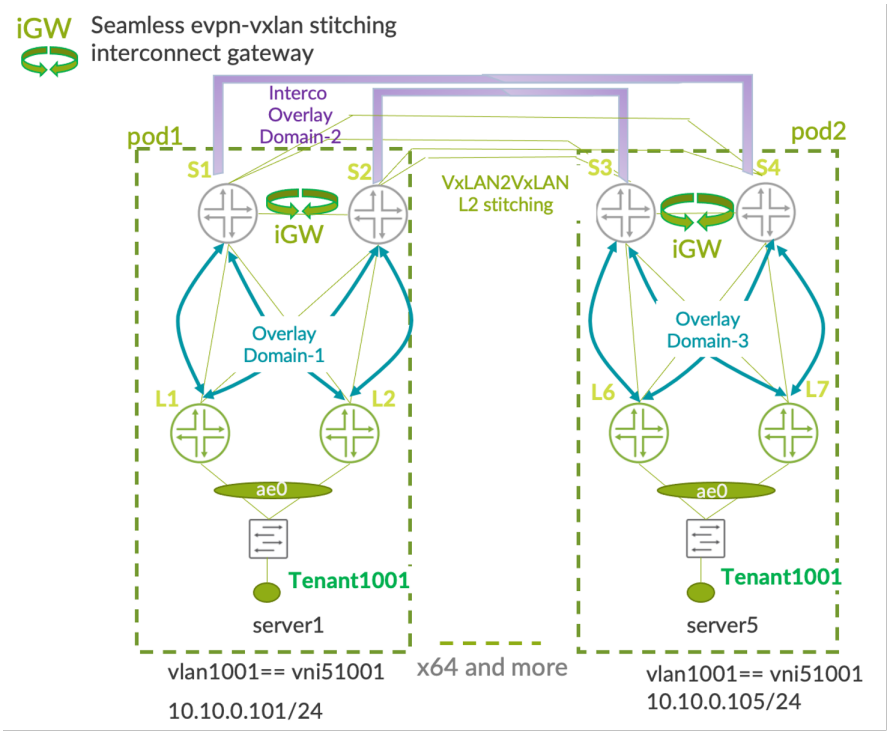


Figure 4.3 *Inter-PoD EVPN-VXLAN Stitching Using Full-Mesh Dark-Fiber*

With this design, we similarly put in place the stitching function at the border-spine level. However, the BGP interconnection for underlay is directly done with the remote site spine nodes. The link between the spine nodes is optional but sometimes preferred for higher redundancy requirement.

Because the design doesn't have any intermediate super-spine or core-IP edge-routers involved, the latency between the PoDs is lower when comparing to the two previous design options.

Typically, with dark fiber the link failure detection is also direct between the two PoDs. However, in the EVPN context, where most BGP peerings are using 3x300ms BFD, it's better to keep the same timers also in the back-to-back dark fiber scenario, for fast convergence consistency between the two PoDs.

Border-leaf Design and Intermediate Core IP Underlay

When the given DC site is deployed using just lean spines (for example using QFX5220-128c or QFX5210-64c) offering IP forwarding and route-server function, then for the L2 and L3 seamless stitching, each PoD will allocate a pair of border-leaf devices (QFX5130, QFX5700, MX240, MX304, ACX7100, and PTX10001-36mr). This is probably the most popular enterprise DC deployment option for data center interconnect where many east-west flows are used within the fabric and the same border-leaf is used to interconnect the data center sites as well as the access to the workloads via the core-IP. In this type of design each border-leaf usually also connects to a firewall cluster to secure the user connections coming via the core-IP and selectively control only the traffic between the DCs via the firewall clusters.

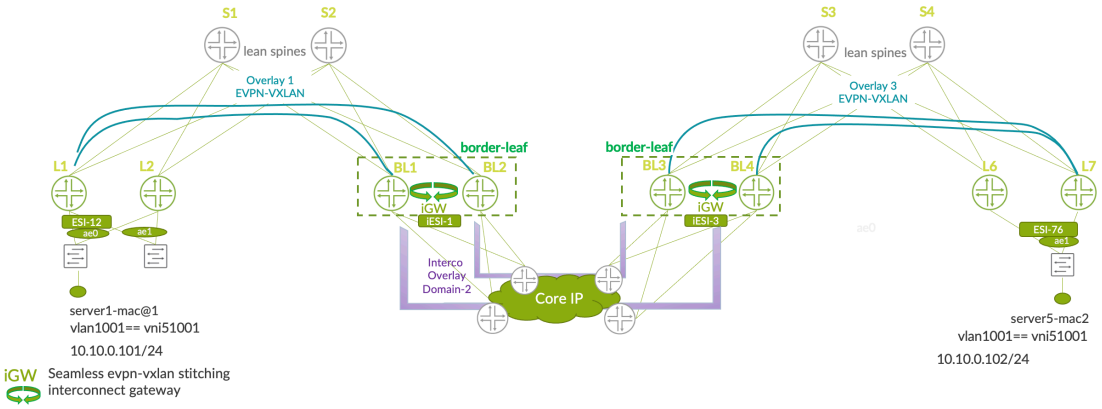


Figure 4.4 DCI EVPN-VXLAN Stitching Using Border-leaf – VXLAN to VXLAN Option

This DCI deployment option is especially popular when the fabric PoD is deployed using the edge routed architecture (ERB), where all tenant virtualization is delivered at the edge and higher segmentation ratio between the workloads is used in general within the fabric.

Comparing border-leaf design with previously discussed border-spine, the main difference is related to the latency and bandwidth used for the north-south connection, as well as for the replication/synchronization between the DCs. So long as the border-leaf is having enough links to the lean spines to cover the site-to-site replication and north-south connection, or when simply most of the traffic stays within the DC site, then this design option is more interesting compared to the others previously mentioned. Mainly because the site local forwarding functions delivered by the lean spines are not mixed with the DCI function, the design offers a better demarcation point between the two distinct functions, and reduces the blast radius for any of the lean spines or border-leaf nodes operations such as upgrades or node replacements.

When using the core IP to connect between the border-leaf block from each DC site, you can occasionally have a situation where the IP connectivity to the remote border-leaf VTEPs is offered via the existing PE IPVPN MPLS service. This is still something possible to consider as an underlay connectivity option, although we recommend in this case extending the core IP domain of the IGP down to the border-leaf and just advertise the local border-leaf loopbacks into the existing IGP domain.

Border-leaf Design with Direct MPLS Connect

When the data center already has an MPLS direct connect to the backbone network, offering quick reachability to the rest of the data centers from the newly deployed fabric, sometimes it's just easier to connect the border-leaf nodes directly to the MPLS P routers. In this case the fabric workloads are still interconnected end-to-end using EVPN control plane. However, at the border-leaf nodes (MX series) the fabric wide VXLAN domain is stitching to MPLS LSP from the transport point of view. See Figure 4.5.

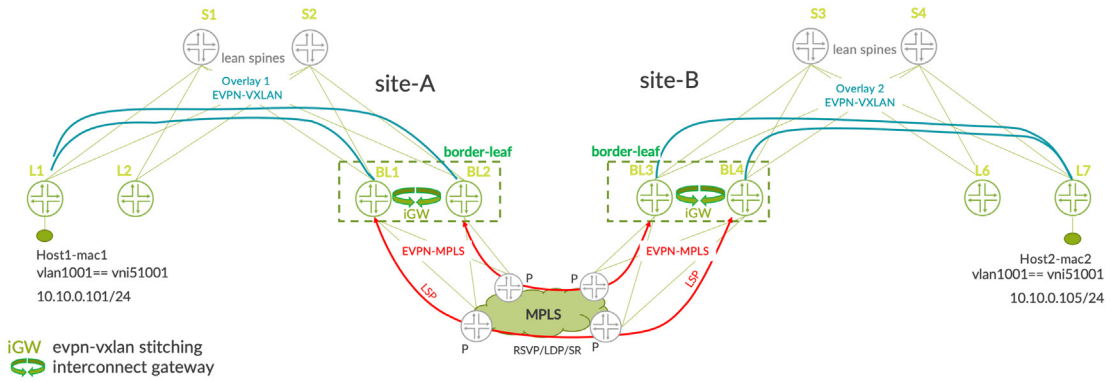


Figure 4.5 *DCI EVPN-VXLAN Stitching Option Using Border-leaf VXLAN-to-MPLS Option*

The good thing in this case of the seamless stitching techniques is that the interconnect option is enabled within the same tenant MAC-VRF and brings the same deterministic L2 extension capabilities, instead of the default full mesh VXLAN overlay between the DC sites leveraging the MPLS for the underlay.

In the case of VXLAN to MPLS stitching, besides the automatic change of transport type, from VXLAN in the fabric to MPLS for DCI, most of the control plane EVPN stays the same. However, the border-leaf, instead of generating VNI specific IM routes to the remote sites, will originate the bridge-domain VLAN related routes and associate them with the EVI MPLS label. This means that every EVI instance at the border-leaf will also originate a MPLS aliasing label for active/active load-balancing purposes, from DC site-A to DC site-B. At the border-leaf node, the split-horizon rules will be triggered for any BUM traffic from the core to make sure it is not flooded back to the originating domain.

Chapter 5

Seamless EVPN-VXLAN to EVPN-VXLAN Stitching – Implementation and Verification

To better understand the low-level implementation and verification aspects of the seamless EVPN-VXLAN to EVPN-VXLAN stitching, the simplified lab topology in Figure 5.1 is used.

Use Case

In the proposed example, there is a bigger data center on site-A and a smaller on site-B, interconnected via the VXLAN to VXLAN DCI tunnel stitching as part of the scaling and admin operations optimization project.

The two DC sites are connected using 100Ge links. Note that the crossed links towards both remote site border-leaf devices are not used here in the lab topology but are recommended for even better redundancy.

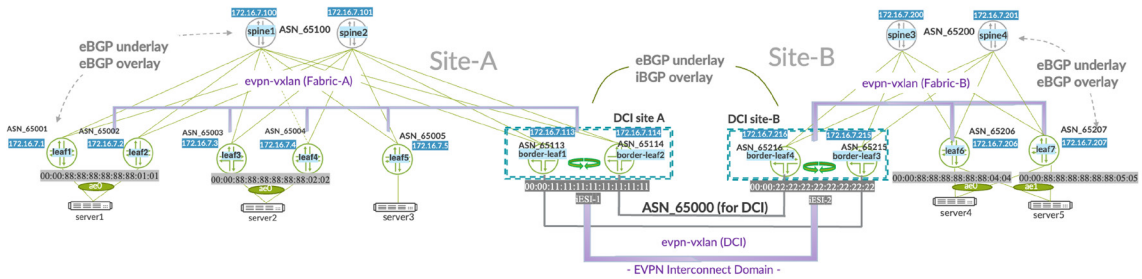


Figure 5.1 *Seamless EVPN-VXLAN Tunnel Stitching - Lab Topology*

Table 5.1 summarizes the main configuration components used for the provisioning of the DCI infrastructure shown in Figurer 5.1 using the VLAN-aware EVPN Instances (EVI) called MACVRF101 and eBGP underlay/overlay peering.

Table 5.1 BGP Overlay/Underlay ASN, Loopback, RD, RT Information Inside the Fabric

Device name	Loopback/ Router-id	RD – route distinguisher for the LAN	MAC-VRF (EVI) name	RT - LAN route-target for LAN	RD – route distinguisher for interconnect	RT – Interco. route-target	Under-lay BGP ASN#	Over-lay BGP ASN#
DC Site-A								
spine1	172.16.7.100	-	-	-	-	-	65100	65100
spine2	172.16.7.101	-	-	-	-	-	65100	65100
leaf1	172.16.7.1	172.16.7.1:1	MACVRF101	target:1:8888	-	-	65001	65001
leaf2	172.16.7.2	172.16.7.2:1	MACVRF101	target:1:8888	-	-	65002	65002
leaf3	172.16.7.3	172.16.7.3:1	MACVRF101	target:1:8888	-	-	65003	65003
leaf4	172.16.7.4	172.16.7.4:1	MACVRF101	target:1:8888	-	-	65004	65004
leaf5	172.16.7.5	172.16.7.5:1	MACVRF101	target:1:8888	-	-	65005	65005
border-leaf1	172.16.7.113	172.16.7.113:1	MACVRF101	target:1:8888	172.16.7.113:101	target:1:101	65113	65113
border-leaf2	172.16.7.114	172.16.7.114:1	MACVRF101	target:1:8888	172.16.7.114:101	target:1:101	65114	65114
DC Site-B								
spine3	172.16.7.200	-	-	-	-	-	65200	65200
spine4	172.16.7.201	-	-	-	-	-	65200	65200
leaf6	172.16.7.206	172.16.7.206:1	MACVRF101	target:1:9999	-	-	65206	65206
leaf7	172.16.7.207	172.16.7.207:1	MACVRF101	target:1:9999	-	-	65207	65207
border-leaf3	172.16.7.215	172.16.7.215:1	MACVRF101	target:1:9999	172.16.7.215:101	target:1:101	65215	65215
border-leaf4	172.16.7.216	172.16.7.216:1	MACVRF101	target:1:9999	172.16.7.216:101	target:1:101	65216	65216

When considering DCI interconnections, each site is enabled with a site id iESI at the MAC-VRF level. Because in the suggested example we use only one MAC-VRF(EVI) called MACVRF101 the following DC site-id ESIs are used listed in Table 5.2.

Table 5.2 ESI - Site Identifiers Used At The Interconnect Gateway Border-leaf Nodes

iESI site identifier for DC site-A	
MACVRF101	00:00:11:11:11:11:11:11:11:11
iESI site identifier for DC site-B	
MACVRF101	00:00:22:22:22:22:22:22:22:22

The ESI for DCI site identification shown in Table 5.2 are used only at the border-leaf level in each DC and are not enabled at the server leaf nodes. As explained in the previous chapters, these ESI values are rewritten for EVPN Route type-2 at the border-leaf level. This way, the remote site border-leaf nodes are using a single iESI value for a given MAC-VRF to reach all the MAC addresses located on the local site. The leaf nodes in each site use local border-leaf ESI value to reach the remote MAC addresses, effectively reducing the number of next hops installed at the PFE level.

The following VLANs and VNIs are enabled inside the EVI MACVRF101 and extended between the two DC sites to connect all servers together. The servers (server1/2/3 in DC1 and server4 and server5 in DC2) are running standard VMware virtual machines so multiple VLANs must be enabled to each of the servers, however only the VLAN1001 and VLAN1002 will be extended between the two DC sites, because only that service has the site level redundancy.

Table 5.3 VLAN-VNI information for DC-A and DC-B

EVI/MAC-VRF name	VLAN-name	vlan-id	VNI
VLAN/VNIs shared between DC site-A and site-B			
MACVRF101	VLAN1001	1001	51001
MACVRF101	VLAN1002	1002	51001
VLAN/VNIs specific to DC site-A			
MACVRF101	VLAN10	10	5010
MACVRF101	VLAN20	20	5020
VLAN/VNIs specific to DC site-B			
MACVRF101	VLAN30	30	5030
MACVRF101	VLAN40	40	5040

The vlan-id information shown in Table 5.3 is used to provision each leaf node locally. However, vlan-id information is stripped at the server leaf nodes before sending the traffic into the VXLAN tunnel towards the border.

Because the DC sites are also deployed with the anycast IP gateway model (aka *EVPN-VXLAN ERB*), the following IRB interfaces IP addressing are used at the leaf nodes. The border-leaf nodes don't have the same IRB interfaces enabled, while typically in a production environment some IRB interfaces would be present at the border-leaf level, mainly to connect the external firewall cluster or other appliances. The IRB interfaces are placed in the same IP VRF called T5-VRF-1. See Table 5.4.

Table 5.4 IRB interfaces IP addressing details

IP VRF name	IRB interface	IP address of the IRB interface	Corresponding VLAN name	Corresponding local VLAN id
IRB interfaces enabled at DC site-A and site-B				
T5-VRF1	irb.1001	10.10.0.1/24	VLAN1001	1001
T5-VRF1	irb.1002	10.10.1.1/24	VLAN1002	1002
IRB interfaces specific to DC site-A				
T5-VRF1	irb.10	10.10.10.1/24	VLAN10	10
T5-VRF1	irb.20	10.10.20.1/24	VLAN20	20
IRB interfaces specific to DC site-B				
T5-VRF1	irb.30	10.10.30.1/24	VLAN30	30
T5-VRF1	irb.40	10.10.40.1/24	VLAN40	40

On the server-facing interfaces (aka "PE-CE" ports) the following servers are connected and extended between the two data centers in VLAN1001. See Table 5.5.

Table 5.5 Servers MAC and IP Information

Server name	server MAC@	server IP@	VLAN-name	vlan-id	VNI	ESI-LAG	LACP system-id	Physical interface
Servers in DC site-A								
server1	00:50:56:-AB:01:01	10.10.0.101/24	VLAN-1001	1001	51001	00:00:88:88:88:88:88:88:01:01	00:01:88:88:01:01	et-0/0/50 leaf1 et-0/0/4 leaf2
server2	00:50:56:-AB:01:02	10.10.0.102/24	VLAN-1001	1001	51001	00:00:88:88:88:88:88:88:02:02	00:01:88:88:02:02	et-0/0/50 leaf3 et-0/0/50 leaf4

server3	00:50:56:-AB:01:03	10.10.0.103/24	VLAN-1001	1001	51001	00:00:00:00:00:00:00:00:00:00 (single homed)	(single homed)	et-0/0/50 leaf5
Servers in DC site-B								
server4	00:50:56:-AB:01:04	10.10.0.104/24	VLAN-1001	1001	51001	00:00:88:88:88:88:88:88:04:04	00:01:88:88:04:04	xe-0/0/14:0 leaf6 xe-0/0/14:0 leaf7
server5	00:50:56:-AB:01:05	10.10.0.105/24	VLAN-1001	1001	51001	00:00:88:88:88:88:88:88:05:05	00:01:88:88:05:05	xe-0/0/12:0 leaf6 xe-0/0/12:0 leaf7

Table 5.6 tracks the verification outputs shown in the next section. For example, the ESI-LAG value highlighted above is used within each fabric but when stitching at the border-leaf, it is replaced with the site identifier iESI values. So, depending on the device where the commands are executed, the ESI is changing for the given end host MAC address.

For the VXLAN-to-VXLAN stitching lab purposes, the VLAN1001 is our main focus in the next sections, even if the VLAN1002 is also enabled in the same MACVRF101 EVI and stretched between the two data center sites.

The data shared in the previous tables are also included in the following reference lab topology, Figure 5.2, where you can see the server MAC and IP information for the VLAN1001 which is mapped inside each fabric and between the DC sites to VNI 51001.

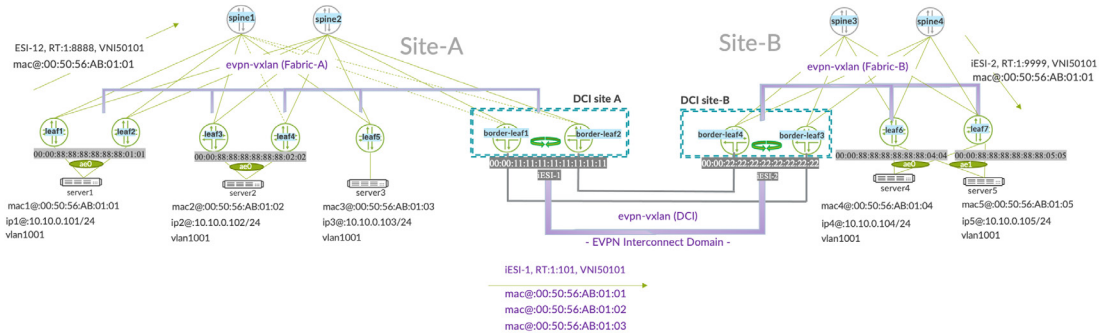


Figure 5.2 *Seamless EVPN-VXLAN Stitching Lab Diagram - Local Site ESI and Interconnect ESI Values*

In the upcoming sections the focus will be on the configurations and verifications used at the border-leaf nodes. We are also showing the selected leaf devices mac-VRF configurations and verifications for reference purposes.

Underlay and Overlay BGP Peering Provisioning and Verification

Before we can start provisioning the MAC-VRFs with any VLAN-VNIs inside, the baseline underlay and overlay eBGP routing must be fully operational and deliver the IP reachability between the loopbacks of all the nodes.

The underlay routing can also be achieved using ISIS or OSPFv2. However, to reduce the number of routing protocols used in the network, we decided to use eBGP for underlay and overlay purposes inside the fabric and iBGP overlay between the data center sites. For EVPN overlay, BGP is the only routing protocol option. For the DCI sometimes the underlay is just an IGP such as ISIS or OSPFv2 however in our case we just used the eBGP underlay to deliver site to site border-leaf loopback reachability.

In the present example, eBGP is used for the underlay so the IP@ of the loopbacks from global routing table are exported within the underlay BGP group to be later used as BGP peering and protocol next hop (PNH) for the overlay (EVPN) as well as source IP address for the VTEP tunnels (BGP PNH and VTEP.SRC.IP == Loopback0.0 IP address from the global routing table).

In smaller/medium size DC fabrics, usually the 16-bit BGP ASN numbers are fine, but all Junos devices can also support provisioning the 32-bit ASN numbers with much more private BGP ASN numbers available per DC site. In this configuration example the 16-bit BGP ASN numbers are used. Every node gets a unique underlay BGP ASN number, so every node-to-node peering in the underlay as well as in the overlay will be eBGP type – external BGP.

To reduce any sub-optimal forwarding, both spines in each DC site use the same ASN# number in the underlay and overlay. Server leaf nodes each get a unique ASN# for underlay and overlay inside the fabric and only a common overlay BGP ASN 65000 is used for DCI purposes. With this approach when there's a third DC site to be built in the future it'll simply connect to the existing overlay common BGP ASN number and get reachability to the existing sites. The way ASN BGP numbers are allocated for this Day One book lab is highlighted in Figure 5.3.

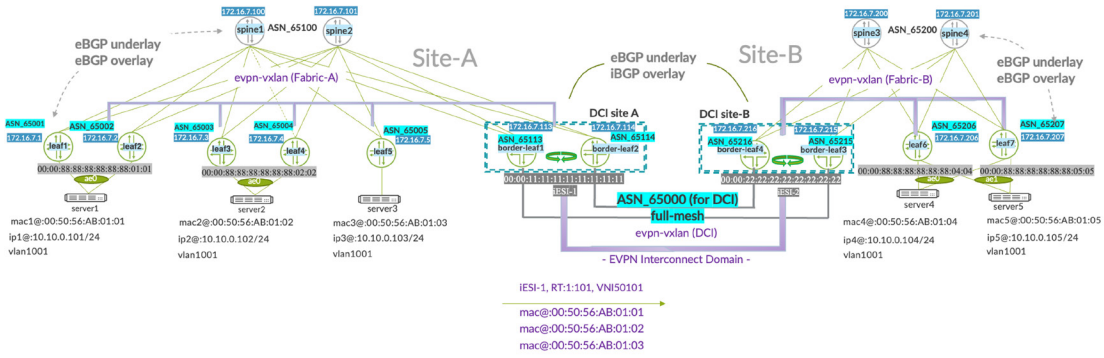


Figure 5.3 BGP Underlay/Overlay Peerings and ASN Allocations

As highlighted in Figure 5.3, the eBGP underlay and iBGP overlay approach is extended at the DCI level. However, when there's an existing core IP routing, the underlay IP routing may be different. For example, when the core-IP is already running an IGP. As long as the underlay IP routing solution delivers full IP reachability between the loopback addresses of the interconnect gateways, then an IGP can be also a valid design option.

In our lab, for simplicity and consistency with the fabric design, we just used eBGP underlay and eBGP overlay inside the fabric and eBGP underlay and iBGP overlay between the DC sites.

Loopbacks of the server leaf nodes and their end-to-end reachability is mainly needed within the fabric as they are not needed between the DC sites. Indeed, the interconnect gateway (border-leaf in our lab topology) changes the routes originated at the server leaf nodes by changing the PNH IP with the one used locally at the iGW. Thereby, the site-to-site underlay loopback reachability is mainly required for the interconnect gateways - border-leaf nodes.

Here's an example of the underlay provisioning between the leaf5 qfx5120 and spine1/spine2 in site-A.

Config. 1 server leaf5 eBGP underlay & overlay peering

```
set protocols bgp group underlay type external >> we explicitly say what type of BGP peering we'll be
using
set protocols bgp group underlay export my_underlay_export >> this route policy-statement is for local
loopback adv.
set protocols bgp group underlay local-as 65005 >> unique BGP ASN for underlay
set protocols bgp group underlay multipath multiple-as
set protocols bgp group underlay neighbor 192.168.220.1 peer-as 65100 >> spine1 underlay peering using
the interface et-0/0/20 IP@
set protocols bgp group underlay neighbor 192.168.222.1 peer-as 65100 >> spine2 underlay peering using
the interface et-0/0/20 IP@
set protocols bgp group overlay type external
set protocols bgp group overlay multihop >> we need to specify it explicitly because our peering is
using loopback interfaces
set protocols bgp group overlay local-address 172.16.7.5 >> corresponding to local loopback0.0 IP
address
set protocols bgp group overlay family evpn signaling >> at the server leaf only the EVPN address-
family is used
set protocols bgp group overlay local-as 65005 >> unique BGP ASN for overlay
set protocols bgp group overlay multipath multiple-as
set protocols bgp group overlay neighbor 172.16.7.100 peer-as 65100 >> spine1 overlay peering to his
loopback
set protocols bgp group overlay neighbor 172.16.7.101 peer-as 65100 >> spine2 overlay peering to his
loopback
set protocols bgp group overlay vpn-apply-export >> this is needed when implementing any export
policy-statements
set policy-options policy-statement my_underlay_export term term1 from route-filter 172.16.7.0/24
prefix-length-range /32-/32
set policy-options policy-statement my_underlay_export term term1 then accept
```

The policy-statement called `my_underlay_export` is used to advertise the local loopback0.0 IP@ into the underlay routing. The 192.168.220.1 and 192.168.222.1 IP addresses are the spine enabled IP addresses which are reachable via the interfaces et-0/0/48 and et-0/0/49 from the server leaf5.

Config. 2 leaf5 100Ge interfaces configuration for underlay IP to connect to spine1/spine2

```
set interfaces et-0/0/48 mtu 9216
set interfaces et-0/0/48 unit 0 family inet address 192.168.220.2/24  >> underlay interface
IP@ used to connect to spine1

set interfaces et-0/0/49 mtu 9216
set interfaces et-0/0/49 unit 0 family inet address 192.168.222.2/24  >> underlay interface
IP@ used to connect to spine2
```

To make sure we load-balance the traffic from the leaf to both spines using IP ECMP, the following configurations are required at the routing-options level for the underlay IP routing, where the policy-statement called `LB` is exported at the routing-options forwarding-table level.

Config. 3 server leaf5 IP ECMP underlay routing-options config

```
set routing-options router-id 172.16.7.5  >> it's better to use the value corresponding to local
loopback IP@ and RD used in VRFs
set routing-options forwarding-table export LB
set routing-options forwarding-table chained-composite-next-hop ingress evpn  >> this part is for
qfx5120 leaf when T5 instance is used
## the following policy-statement is used to deliver the IP ECMP underlay towards spines
set policy-options policy-statement LB term term1 from protocol evpn
set policy-options policy-statement LB term term1 then load-balance per-packet
set policy-options policy-statement LB term term1 then accept
set policy-options policy-statement LB term term2 then load-balance per-packet
set policy-options policy-statement LB term term2 then accept
```

The similar eBGP underlay and overlay configurations can be repeated on the other leaf nodes where only the local-address, BGP local-as and underlay eBGP peering IP addressing is changing. For example, here's the BGP configuration of the server leaf1.

Config. 4 server leaf1 eBGP underlay/overlay configuration

```
set protocols bgp group underlay type external
set protocols bgp group underlay export my_underlay_export
set protocols bgp group underlay local-as 65001
set protocols bgp group underlay multipath multiple-as
set protocols bgp group underlay neighbor 192.168.15.1 peer-as 65100
set protocols bgp group underlay neighbor 192.168.16.1 peer-as 65100
set protocols bgp group overlay type external
set protocols bgp group overlay multihop
set protocols bgp group overay local-address 172.16.7.1
set protocols bgp group overlay family evpn signaling
```

```
set protocols bgp group overlay local-as 65001
set protocols bgp group overlay multipath multiple-as
set protocols bgp group overlay bfd-liveness-detection minimum-interval 100
set protocols bgp group overlay bfd-liveness-detection multiplier 3
set protocols bgp group overlay neighbor 172.16.7.100 peer-as 65100
set protocols bgp group overlay neighbor 172.16.7.101 peer-as 65100
set protocols bgp group overlay vpn-apply-export
set policy-options policy-statement my_underlay_export term term1 from route-
filter 172.16.7.0/24 prefix-length-range /32-/32
set policy-options policy-statement my_underlay_export term term1 then accept
```

That loopback IP@ advertised from each node in the fabric via underlay routing is, in fact, used for BGP overlay multi-hop peerings with family EVPN. Moreover, it is also used as the IP@ for the VxLAN tunnel termination within the given site fabric.

It means, within the given EVPN-VXLAN fabric, that leaf1 and leaf5 are using their local loopback IP addresses for the VXLAN tunnels. This address is also the protocol next hop used in the overlay BGP advertisement for all EVPN route types within the fabric. However, for the MAC@ reachable in remote destinations from site B, local leaf1 and leaf5 use the BGP protocol next hop of the local site border-leaf1/border-leaf2 instead of the protocol next hop of leaf6/leaf7 directly.

From the lean spine perspective, BGP configuration also involves underlay and overlay peering. However, because there is not any MAC-VRF configuration on these spines, the VXLAN tunnel termination won't take place and the spines will be acting as pure IP forwarders and route-servers.

From the BGP point of view, the lean spines need to be provisioned with the explicit `multihop no-nexthop-change` knob to change the default behavior of eBGP peering which changes the BGP protocol next hop with his local IP value.

At the border-leaf level we use a same BGP ASN value for underlay and overlay but unique per node, on the fabric side when peering eBGP to the spines – ASN 65113 for underlay and overlay at border-leaf1 and ASN 65114 for underlay and overlay at border-leaf2.

The `multihop no-nexthop-change` knob, in the context of seamless DCI, is not needed at the border-leaf level because it changes the EVPN routes protocol-next hop with his local loopback IP address by default.

For example, here is the peering configuration on border-leaf1 (in DC site-A) to border-leaf3 (in DC site-B).

Config. 5 border-leaf1 underlay/overlay peering

```
set protocols bgp group underlay type external
set protocols bgp group underlay export my_underlay_export  >> policy-statement to advertise the local
loopback0.0
set protocols bgp group underlay local-as 65113  >> underlay BGP ASN in DC site A
set protocols bgp group underlay multipath multiple-as
```

```
set protocols bgp group underlay neighbor 192.168.53.1 peer-as 65100  >> underlay peering to spine1
set protocols bgp group underlay neighbor 192.168.53.1 import my_underlay_import
set protocols bgp group underlay neighbor 192.168.63.1 peer-as 65100  >> underlay peering to spine2
set protocols bgp group underlay neighbor 192.168.63.1 import my_underlay_import
set protocols bgp group underlay neighbor 192.168.63.1 peer-as 65100
set protocols bgp group underlay neighbor 192.168.228.2 peer-as 65215  >> underlay peering to border-
leaf3 in DC site B
set protocols bgp group overlay type external
set protocols bgp group overlay multihop
set protocols bgp group overlay local-address 172.16.7.113
set protocols bgp group overlay family evpn signaling
set protocols bgp group overlay local-as 65113  >> same overlay BGP ASN inside the fabric – site A
set protocols bgp group overlay multipath multiple-as
set protocols bgp group overlay neighbor 172.16.7.100 peer-as 65100  >> overlay peering to spine1
set protocols bgp group overlay neighbor 172.16.7.101 peer-as 65100  >> overlay peering to spine2
set protocols bgp group overlay vpn-apply-export
```

The border-leaf2 underlay/overlay eBGP is consistent with border-leaf1 – both are using the different overlay ASN number which helps identifying routes inside the fabric.

Config. 6 border-leaf2 underlay/overlay peering with same ASN for both border-leaf nodes

```
set protocols bgp group underlay type external
set protocols bgp group underlay export my_underlay_export >> policy statement to advertise the local
loopback0.0
set protocols bgp group underlay local-as 65114 >> underlay BGP ASN at the DC site A
set protocols bgp group underlay multipath multiple-as
set protocols bgp group underlay neighbor 192.168.54.1 peer-as 65100 >> underlay peering to spine1
set protocols bgp group underlay neighbor 192.168.54.1 import my_underlay_import
set protocols bgp group underlay neighbor 192.168.64.1 peer-as 65100 >> underlay peering to spine2
set protocols bgp group underlay neighbor 192.168.64.1 import my_underlay_import
set protocols bgp group underlay neighbor 192.168.231.2 peer-as 65215 >> underlay peering to border-
leaf4 in DC site B
set protocols bgp group overlay type external set protocols bgp group overlay multihop
set protocols bgp group overlay local-address 172.16.7.114 set protocols bgp group overlay family evpn
signaling
set protocols bgp group overlay local-as 65114 >> overlay BGP ASN inside the fabric – site A
set protocols bgp group overlay multipath multiple-as
set protocols bgp group overlay neighbor 172.16.7.100 peer-as 65100 >> overlay peering to spine1
set protocols bgp group overlay neighbor 172.16.7.101 peer-as 65100 >> overlay peering to spine2
set protocols bgp group overlay vpn-apply-export
```

The following policy statements for import were used in DC site A on both border-leaf nodes in site A. The import policy was used to make sure the remote site loopbacks (used to terminate the VXLAN tunnels) are not learned also via the local spines. If we were not restricting it through the import policy of the underlay BGP group, there could be some suboptimal path also via the local spine nodes.

```
set policy-options policy-statement my_underlay_import term term1 from route-
filter 172.16.7.215/32 exact
set policy-options policy-statement my_underlay_import term term1 from route-
filter 172.16.7.216/32 exact
set policy-options policy-statement my_underlay_import term term1 then reject
set policy-options policy-statement my_underlay_import term term2 then accept
```

The export policy is simply there for underlay to advertise the local loopback IP@ to the rest of the fabric and to the remote location. Advertising to the remote location the local fabric loopbacks of the server leaf nodes is optional because besides the border-leaf to border-leaf loopback reachability between the sites the server leaf from site A to server leaf from site B won't have any direct VxLAN tunnel established.

```
set policy-options policy-statement my_underlay_export term term1 from route-
filter 172.16.7.0/24 prefix-length-range /32-/32
set policy-options policy-statement my_underlay_export term term1 then accept
```

When it comes to DCI overlay peering from border-leaf nodes, a new overlay BGP group called DCI is introduced and a full mesh of iBGP peering. We decided to use full mesh of iBGP because only two DC sites were used in our example, however in case of multiple sites, typically an in-path or off-path Route-Reflector would be used.

At the border-leaf nodes in site A we used the following overlay DCI BGP peering configuration:

Config. 7 iBGP EVPN overlay configuration at border-leaf1

```
set protocols bgp group DCI type internal
set protocols bgp group DCI local-address 172.16.7.113
set protocols bgp group DCI family evpn signaling
set protocols bgp group DCI local-as 65000
set protocols bgp group DCI multipath
set protocols bgp group DCI neighbor 172.16.7.215
set protocols bgp group DCI neighbor 172.16.7.216
set protocols bgp group DCI neighbor 172.16.7.114
set protocols bgp group DCI vpn-apply-export
```

Config. 8 iBGP EVPN overlay configuration at border-leaf2

```
set protocols bgp group DCI type internal
set protocols bgp group DCI local-address 172.16.7.114
set protocols bgp group DCI family evpn signaling
set protocols bgp group DCI local-as 65000
set protocols bgp group DCI multipath
set protocols bgp group DCI neighbor 172.16.7.215
set protocols bgp group DCI neighbor 172.16.7.216
set protocols bgp group DCI neighbor 172.16.7.113
set protocols bgp group DCI vpn-apply-export
```

The border-leaf nodes in DC site B are using consistent approach for iBGP full mesh config and peering to his local border-leaf as well as remote location border-leaf nodes in site A.

Config. 9 iBGP EVPN overlay configuration at border-leaf3

```
set protocols bgp group DCI type internal
set protocols bgp group DCI local-address 172.16.7.215
set protocols bgp group DCI family evpn signaling
set protocols bgp group DCI local-as 65000
set protocols bgp group DCI multipath
set protocols bgp group DCI neighbor 172.16.7.113
set protocols bgp group DCI neighbor 172.16.7.114
set protocols bgp group DCI neighbor 172.16.7.216
set protocols bgp group DCI vpn-apply-export
```

Config. 10 iBGP EVPN overlay configuration at border-leaf4

```
set protocols bgp group DCI type internal
set protocols bgp group DCI local-address 172.16.7.216
set protocols bgp group DCI family evpn signaling
set protocols bgp group DCI local-as 65000
set protocols bgp group DCI multipath
set protocols bgp group DCI neighbor 172.16.7.113
set protocols bgp group DCI neighbor 172.16.7.114
set protocols bgp group DCI neighbor 172.16.7.215
set protocols bgp group DCI vpn-apply-export
```

You can see in the above configuration examples that even if our site to site connect is using a partial mesh, we enabled a full mesh because we wanted to make it ready for any additional links between the DC sites.

MAC-VRF Implementation and Verification for DC and DCI

Once the BGP underlay and overlay are in place, you can tackle the MAC-VRF (aka EVI) configurations to introduce new VLAN-VNIs and associate server interfaces to it.

To standardize and simplify the provisioning of EVPN, the new MAC-VRF implementation is introduced and is used on all QFX series devices. With the new mac-VRF instance type, all the three standard EVPN service types (VLAN-aware, VLAN-bundle, VLAN-based) can be explicitly defined and enabled at the same node using different routing-instance names.

In our lab, MAC-VRF (EVI) with service-type vlan-aware and multiple VLANs is used on server leaf switch and border-leaf nodes.

In the section dedicated to EVPN-VXLAN to EVPN-MPLS, we're using MX series routers for the stitching from VXLAN to MPLS and therefore the legacy instance-type virtual-switch EVI configuration is used. This is the equivalent of VLAN-aware EVPN service-type and can interoperate with the QFX series running the new MAC-VRF instance-type configurations.

The following L2 configuration is used to provision leaf1 when connecting server1 the EVPN-VXLAN fabric in site-A.

Config. 7 leaf1 mac-VRF (EVI) configuration

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-
instances MACVRF101 protocols evpn encapsulation VXLAN  >> the only encap. option for qfx5110/
qfx5120 leaf
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-
community  >> needed with local IRB interfaces
set routing-instances MACVRF101 protocols evpn extended-vni-list 5010  >> VNI local to fabric A
set routing-instances MACVRF101 protocols evpn extended-vni-list 5020  >> VNI local to fabric A
set routing-instances MACVRF101 protocols evpn extended-vni-
list 51001  >> VNI local to fabric A and stretched to fabric B
set routing-instances MACVRF101 protocols evpn extended-vni-
list 51002  >> VNI local to fabric A and stretched to fabric B
set routing-instances MACVRF101 vtep-source-
interface lo0.0  >> loopback0.0 interface is only across all MAC-VRFs
set routing-instances MACVRF101 service-type vlan-aware  >> explicit EVPN service type definition
set routing-instances MACVRF101 interface ae0.0  >> ESI-
LAG interface association with the given MAC-VRF
set routing-instances MACVRF101 route-
distinguisher 172.16.7.1:1  >> each server and border node will use unique value
set routing-instances MACVRF101 vrf-
target target:1:8888  >> the local site A route target shared with other nodes in DC A
set routing-instances MACVRF101 vlans vlan10 vlan-id 10  >> VLANs are defined inside MAC-
VRF this vlan-id is not L2 extended to site B
set routing-instances MACVRF101 vlans vlan10 l3-
interface irb.10  >> the IRB IP interface associated with VLAN in the given MAC-VRF
set routing-instances MACVRF101 vlans vlan10 vxlan vni 5010
set routing-instances MACVRF101 vlans vlan1001 vlan-
id 1001  >> VLAN extended between DC site A and DC site B at the border-leaf1/2
set routing-instances MACVRF101 vlans vlan1001 l3-interface irb.1001
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 51001
set routing-instances MACVRF101 vlans vlan1002 vlan-
id 1002  >> VLAN extended between DC site A and DC site B at the border-leaf1/2
set routing-instances MACVRF101 vlans vlan1002 l3-interface irb.1002
set routing-instances MACVRF101 vlans vlan1002 vxlan vni 51002
set routing-instances MACVRF101 vlans vlan20 vlan-id 20  >> VLANs are defined inside MAC-
VRF this vlan-id is not L2 extended to site B
set routing-instances MACVRF101 vlans vlan20 l3-interface irb.20
set routing-instances MACVRF101 vlans vlan20 vxlan vni 5020
```

The leaf1 interface configuration connected to server1 is using the enterprise-style configuration with local ESI value – significant within the DC1.

Config. 8 ESI-LAG configuration to connect server1

```
set interfaces ae0 description ep-style
set interfaces ae0 mtu 9100
set interfaces ae0 esi 00:00:88:88:88:88:88:88:01:01  >> same value to be used at leaf2 when connecting
to server1 using ESI-LAG
set interfaces ae0 esi all-active  >> only the all-active ESI is used for L2 multihoming using EVPN-
VXLAN
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:01:88:88:01:01  >>  unique value per ESI-
LAG
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
```

```
set interfaces ae0 unit 0 family ethernet-switching vlan members 10  >> VLAN local to DC1
set interfaces ae0 unit 0 family ethernet-switching vlan members 20  >> VLAN local to DC1
set interfaces ae0 unit 0 family ethernet-switching vlan members 1001-1002  >> VLANs stretched between
DC1 and DC2
set interfaces et-0/0/50 ether-options 802.3ad ae0  >> associating physical interface towards the
server1 with ESI-LAG
set chassis aggregated-devices ethernet device-count 2  >> to be enabled if not yet used globally at
the given server-leaf
```

When the server leaf is used with QFX5120 platforms, the following global configurations are also required to enable the MAC-VRF based EVI implementation.

Config. 9 leaf1 global forwarding configurations for MAC-VRF and scaling

```
set forwarding-options evpn-vxlan shared-tunnels  >> this is required for MAC-
VRFs at qfx5110 or qfx5120
set forwarding-options vxlan-routing next-hop 32768  >> we increase the total number of next-
hops from default, this is mainly needed when IRB interface runs IPv4 and IPv6 in parallel
set forwarding-options vxlan-routing interface-
num 8192  >> it's an optional knob to consider when many IPv4/IPv6 tenants
set forwarding-options vxlan-routing overlay-ecmp  >> when same server leaf is using Type-
5 instances this is needed
```

When deploying the IRB anycast interfaces, use the same anycast gateway MAC across all other server nodes.

Config. 10 IRB anycast IP gateway configuration used at server-leafs in DC A

```
set interfaces irb unit 10 family inet address 10.10.10.1/24  >> same IP@ is used across all server leaf
nodes
set interfaces irb unit 10 mac 00:00:01:01:01:01  >> same MAC@ is used across all server leaf nodes
set interfaces irb unit 20 family inet address 10.10.20.1/24
set interfaces irb unit 20 mac 00:00:01:01:01:01
set interfaces irb unit 1001 family inet address 10.10.0.1/24
set interfaces irb unit 1001 mac 00:00:01:01:01:01
set interfaces irb unit 1002 family inet address 10.10.1.1/24
set interfaces irb unit 1002 mac 00:00:01:01:01:02
```

NOTE These IRB interfaces will be enabled inside the IP Type-5 VRF later with the Type-5 tunnel stitching scenario. In the case the fabric is just deployed as an L2 EVPN-VXLAN fabric, the IRB interfaces are not required, and the IP default gateways (like an MX or SRX external DC GW) can be connected to the border-leaf nodes.

For completeness of the explanation, we also introduce the MAC-VRF and interface configurations of the leaf2 also connected to server1.

Config. 11 leaf2 mac-VRF (EVI) configuration

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 5010
set routing-instances MACVRF101 protocols evpn extended-vni-list 5020
```

```
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001
set routing-instances MACVRF101 protocols evpn extended-vni-list 51002
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 service-type vlan-
aware  >> same EVPN service type at the one enabled at leaf1
set routing-instances MACVRF101 interface ae0.0  >>  ESI-LAG connected to server1
set routing-instances MACVRF101 route-
distinguisher 172.16.7.2:1  >> unique value related to the local lo0.0 IP@
set routing-instances MACVRF101 vrf-
target target:1:8888  >> shared RT value with other server leaf in DC A
set routing-instances MACVRF101 vlans vlan10 vlan-id 10
set routing-instances MACVRF101 vlans vlan10 l3-interface irb.10
set routing-instances MACVRF101 vlans vlan10 vxlan vni 5010
set routing-instances MACVRF101 vlans vlan1001 vlan-id 1001
set routing-instances MACVRF101 vlans vlan1001 l3-interface irb.1001
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 51001
set routing-instances MACVRF101 vlans vlan1002 vlan-id 1002
set routing-instances MACVRF101 vlans vlan1002 l3-interface irb.1002
set routing-instances MACVRF101 vlans vlan1002 vxlan vni 51002
set routing-instances MACVRF101 vlans vlan20 vlan-id 20
set routing-instances MACVRF101 vlans vlan20 l3-interface irb.20
set routing-instances MACVRF101 vlans vlan20 vxlan vni 5020
```

For the ESI-LAG interface configuration needed to connect server1, you can notice that it is the same configuration as leaf1: same ESI value, same LACP system-id and same list of VLANs.

Config. 12 leaf2 ESI-LAG configuration to connect server1

```
set interfaces ae0 mtu 9100
# same ESI value as the one used at leaf1 to connect to server1
set interfaces ae0 esi 00:00:88:88:88:88:88:88:01:01
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
# same LACP system-id us set as the one at leaf1
set interfaces ae0 aggregated-ether-options lacp system-id 00:01:88:88:01:01
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members 10
set interfaces ae0 unit 0 family ethernet-switching vlan members 20
set interfaces ae0 unit 0 family ethernet-switching vlan members 1001-1002
set interfaces et-0/0/4 ether-
options 802.3ad ae0  >> used to associate the physical interface towards server1 with an ESI-LAG
set chassis aggregated-devices ethernet device-
count 2  >> should be increased in production network for more ESI-LAGs
```

Similarly, to leaf1 you must enable the following global forwarding options.

Config. 13 global forwarding options required at qfx5110/qfx5120 leaf nodes

```
set forwarding-options evpn-vxlan shared-tunnels  >> this knob is a pre-requisite when enabling
MAC-VRF/ reboot needed
set forwarding-options vxlan-routing next-hop 32768
set forwarding-options vxlan-routing interface-num 8192
set forwarding-options vxlan-routing overlay-ecmp  >> used when deploying the ERB anycast IRB gateway
- T5 instances
```

The leaf2 connected with leaf1 to server1 is also enabled with an IP anycast gateway:

Config. 14 IRB anycast IP interface configuration

```
set interfaces irb unit 10 family inet address 10.10.10.1/24  >> same IP anycast used at leaf1 in DC A
set interfaces irb unit 10 mac 00:00:01:01:01:01  >> same MAC@ used at leaf1 in DC A
set interfaces irb unit 20 family inet address 10.10.20.1/24
set interfaces irb unit 20 mac 00:00:01:01:01:01
set interfaces irb unit 1001 family inet address 10.10.0.1/24
set interfaces irb unit 1001 mac 00:00:01:01:01:01
set interfaces irb unit 1002 family inet address 10.10.1.1/24
set interfaces irb unit 1002 mac 00:00:01:01:01:02
```

With IRB interfaces local to the server leaf, all ARP/ND resolutions are performed locally. In case the same server leaf is enabled with an IP Type-5 instance for prefix-advertisement (ERB design), then the subnet associated with the IRB and, in some cases, also the Type-5 host routes are advertised to the fabric and used at the border-leaf. When a given server leaf gets a MAC-IP Type-2 route as well as the Type-5 host route, then the Type-5 host-route prefix route is always preferred for forwarding by the server leaf.

The border-leaf nodes in each DC site are used as the interconnect gateways (iGW) unifying the border-leaf and gateway role within the same device. That also reduces efficiently the total cost and maintenance of the DCI infrastructure.

The tunnel stitching implementation is done inside the local site MAC-VRF called MACVRF101 where a new block of configuration called 'interconnect' is dedicated to data center interconnect purposes. Here's how DC site A border-leaf1 and border-leaf2 seamless stitching configuration is enabled to stretch the VNI 51001 and 51002, with an explicit list of VNIs that we want to extend between the sites.

Config. 15 border-leaf1 - EVPN-VXLAN to EVPN-VXLAN seamless stitching configuration

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001 >> VNI defined for LAN fabric
purposes in an explicit way
set routing-instances MACVRF101 protocols evpn extended-vni-list 51002
## the new configuration block called 'EVPN interconnect' is used for seamless stitching DCI using
qfx5130/qfx5700
set routing-instances MACVRF101 protocols evpn interconnect vrf-target target:1:101 >> DCI dedicated
route-target shared with remote border
set routing-instances MACVRF101 protocols evpn interconnect route-distinguisher 172.16.7.113:101 >>
unique iRD per border-leaf
set routing-instances MACVRF101 protocols evpn interconnect esi 00:00:11:11:11:11:11:11:11:11 >> the
DC site identifier ESI unique per site
set routing-instances MACVRF101 protocols evpn interconnect esi all-active >> all-active is the only
option but is typically preferred in DC
set routing-instances MACVRF101 protocols evpn interconnect interconnected-vni-list 61001 >> we
defined which VNI to extend between DCs
set routing-instances MACVRF101 protocols evpn interconnect interconnected-vni-list 61002 set routing-
instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 service-type vlan-aware >> we keep the EVPN service type consistent
with what was enabled at server leaf
```

```
set routing-instances MACVRF101 route-distinguisher 172.16.7.113:1 >> unique regular RD for the fabric
LAN purposes
set routing-instances MACVRF101 vrf-target target:1:8888 >> LAN fabric route-target shared with server
leaf nodes
set routing-instances MACVRF101 vlans vlan1001 vlan-id 1001 >> VLAN to VNI mapping
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 51001 >> local site A VNI value shared with
server leaf nodes
set routing-instances MACVRF101 vlans vlan1001 VXLAN translation-VNI 61001 >> explicit but optional VNI
translation for DCI purposes
set routing-instances MACVRF101 vlans vlan1002 vlan-id 1002 set routing-instances MACVRF101 vlans
vlan1002 vxlan vni 51002
set routing-instances MACVRF101 vlans vlan1002 VXLAN translation-VNI 61002
## at the global protocol EVPN level we also explicitly provision the peer IP loopback address of the
border-leaf from same DC site
set protocols evpn interconnect-multihoming-peer-gateways 172.16.7.114 >> required for loop free
forwarding
```

Because at DC site A we have two border leaf nodes the same configuration pattern is also used at the border-leaf2:

Config. 16 border-leaf2 - EVPN-VXLAN to EVPN-VXLAN seamless stitching configuration

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001 >> VNI defined for LAN fabric
purposes in an explicit way
set routing-instances MACVRF101 protocols evpn extended-vni-list 51002
## the new configuration block called 'EVPN interconnect' is used for seamless stitching DCI using
qfx5130/qfx5700
set routing-instances MACVRF101 protocols evpn interconnect vrf-target target:1:101 >> DCI dedicated
route-target shared with remote border
set routing-instances MACVRF101 protocols evpn interconnect route-distinguisher 172.16.7.113:101 >>
unique iRD per border-leaf
set routing-instances MACVRF101 protocols evpn interconnect esi 00:00:11:11:11:11:11:11:11:11 >> the DC
site identifier ESI unique per site
set routing-instances MACVRF101 protocols evpn interconnect esi all-active >> all-active is the only
option but is typically preferred in DC
set routing-instances MACVRF101 protocols evpn interconnect interconnected-vni-list 61001 >> we defined
which VNI to extend between DCs

set routing-instances MACVRF101 protocols evpn interconnect interconnected-vni-list 61002 set routing-
instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 service-type vlan-aware >> we keep the EVPN service type consistent with
what was enabled at server leaf
set routing-instances MACVRF101 route-distinguisher 172.16.7.113:1 >> unique regular RD for the fabric
LAN purposes
set routing-instances MACVRF101 vrf-target target:1:8888 >> LAN fabric route-target shared with server
leaf nodes
set routing-instances MACVRF101 vlans vlan1001 vlan-id 1001 >> VLAN to VNI mapping
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 51001 >> local site A VNI value shared with
server leaf nodes
set routing-instances MACVRF101 vlans vlan1001 VXLAN translation-VNI 61001 >> explicit but optional VNI
translation for DCI purposes
set routing-instances MACVRF101 vlans vlan1002 vlan-id 1002 set routing-instances MACVRF101 vlans
vlan1002 vxlan vni 51002
set routing-instances MACVRF101 vlans vlan1002 VXLAN translation-VNI 61002
## at the global protocol EVPN level we also explicitly provision the peer IP loopback address of the
border-leaf from same DC site
set protocols evpn interconnect-multihoming-peer-gateways 172.16.7.114 >> required for loop free
forwarding
```

When it comes to DC site B, the border-leaf configurations are following the same principle as site-A. However, they have a new interconnect iESI 00:00:22:22:22:22:22:22:22:22 used for load-balancing the L2 traffic on border-leaf3/ border-leaf4. Because the iESI is used within the MACVRF for interconnect DCI purposes, the DF/nDF election also takes place.

In this example, we keep the default MOD election but in Junos/Junos Evo, preference-based DF/nDF election can be optionally used. This is especially useful when there are more MACVRFs enable. With preference-based election, we can control which of the two border-leaf nodes will be responsible for multicast forwarding for which set of VLANs.

Config. 17 border-leaf3 - EVPN-VXLAN to EVPN-VXLAN seamless stitching configuration

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001
set routing-instances MACVRF101 protocols evpn extended-vni-list 51002
set routing-instances MACVRF101 protocols evpn interconnect vrf-target target:1:101  >> iRT used to
advertise routes to remote DC site A
set routing-instances MACVRF101 protocols evpn interconnect route-distinguisher 172.16.7.215:101  >>
used to advertise EVPN routes at DCI
set routing-instances MACVRF101 protocols evpn interconnect esi 00:00:22:22:22:22:22:22:22:22  >> the
iESI specific to DC site B
set routing-instances MACVRF101 protocols evpn interconnect esi all-active
set routing-instances MACVRF101 protocols evpn interconnect interconnected-VNI-list 51001
set routing-instances MACVRF101 protocols evpn interconnect interconnected-VNI-list 51002
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 service-type vlan-aware  >> same EVPN service type enabled in site B
and site A
set routing-instances MACVRF101 route-distinguisher 172.16.7.215:1  >> regular RD used when
advertising EVPN routes to local leaf nodes
set routing-instances MACVRF101 vrf-target target:1:9999  >> the fabric specific route-target is
different in site B from the one in site A
set routing-instances MACVRF101 vlans vlan1001 vlan-id 1001
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 51001
set routing-instances MACVRF101 vlans vlan1001 VXLAN translation-VNI 61001
set routing-instances MACVRF101 vlans vlan1002 vlan-id 1002
set routing-instances MACVRF101 vlans vlan1002 vxlan vni 51002
set routing-instances MACVRF101 vlans vlan1002 VXLAN translation-VNI 61002
## at the global protocol EVPN level we also explicitly provision the peer IP loopback address of the
border-leaf from same DC site
set protocols evpn interconnect-multihoming-peer-gateways 172.16.7.216  >>required to for optimal loop
free forwarding
```

Config. 18 border-leaf4 - EVPN-VXLAN to EVPN-VXLAN seamless stitching configuration

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001
set routing-instances MACVRF101 protocols evpn extended-vni-list 51002
set routing-instances MACVRF101 protocols evpn interconnect vrf-target target:1:101  >> iRT used to
advertise routes to remote DC site A
set routing-instances MACVRF101 protocols evpn interconnect route-distinguisher 172.16.7.216:101  >>
```

```
unique iRD value for DCI purposes
set routing-instances MACVRF101 protocols evpn interconnect esi 00:00:22:22:22:22:22:22:22:22  >>the
iESI specific to DC site B is also a site identifier
set routing-instances MACVRF101 protocols evpn interconnect esi all-active
set routing-instances MACVRF101 protocols evpn interconnect interconnected-VNI-list 51001  >> we
replicate the list of extended VNIs
set routing-instances MACVRF101 protocols evpn interconnect interconnected-VNI-list 51002
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 service-type vlan-aware
set routing-instances MACVRF101 route-distinguisher 172.16.7.216:1
set routing-instances MACVRF101 vrf-target target:1:9999  >> the regular RT value in site B is
different from site A
set routing-instances MACVRF101 vlans vlan1001 vlan-id 1001
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 61001
set routing-instances MACVRF101 vlans vlan1001 VXLAN translation-VNI 61001  >> same as on site A we
translate the local site VNI 51001 to the DCI VNI 61001
set routing-instances MACVRF101 vlans vlan1002 vlan-id 1002
set routing-instances MACVRF101 vlans vlan1002 vxlan vni 51002
set routing-instances MACVRF101 vlans vlan1002 VXLAN translation-VNI 61002
## at the global protocol EVPN level we also explicitly provision the peer IP loopback address of the
border-leaf from same DC site
set protocols evpn interconnect-multihoming-peer-gateways 172.16.7.215  >> required to for optimal
loop free forwarding
```

As you can see, there are no IRB routing interfaces defined at the border-leaf and mapped to different VLANs, this is because we don't have in our design any server also connected to border-leaf nodes.

The translation-VNI function can be interesting when the goal is to maintain the same broadcast domain between the sites and when the remote site is using a different VNI value for a given VLAN. In this case, we can normalize the VNI value at the border-leaf level.

Verification at the border leaf level

At the Border-leaf2, let's check the EVPN routes reflected by spine2 for server1 with mac-address 00:50:56:ab:01:01. This MAC shows up as local fabric and original ESI is used for reachability within the fabric on site A.

```
root@border-leaf2> show route receive-protocol bgp 172.16.7.101 evpn-mac-address 00:50:56:ab:01:01
table MACVRF101.evpn.0
MACVRF101.evpn.0: 101 destinations, 102 routes (101 active, 0 holddown, 0 hidden)
  Prefix              Nexthop           MED    Lclpref    AS path
  2:172.16.7.1:1::51001::00:50:56:ab:01:01/304 MAC/IP
*                     172.16.7.1                          65100 65001 I.  >> MAC Route Type 2

  2:172.16.7.1:1::51001::00:50:56:ab:01:01::10.10.0.101/304 MAC/IP.    >> MAC/IP Route Type 2
*                     172.16.7.1                          65100 65001 I
```

To verify the BGP attributes associated with the local DC site server1 MAC@, use the following Junos command at the border-leaf2:

```
root@border-leaf2> show route receive-protocol bgp 172.16.7.101 evpn-mac-address 00:50:56:ab:01:01
table MACVRF101.evpn.0 detail
MACVRF101.evpn.0: 101 destinations, 102 routes (101 active, 0 holddown, 0 hidden)
* 2:172.16.7.1:1::51001::00:50:56:ab:01:01/304 MAC/IP (1 entry, 1 announced)
    Import Accepted
```

```
    Route Distinguisher: 172.16.7.1:1
    Route Label: 51001  >> corresponding to the VNI value mapped to VLAN1001
    ESI: 00:00:88:88:88:88:88:88:01:01  >> ESI configured at the leaf1/leaf2 to connect server1
    Nexthop: 172.16.7.1
    AS path: 65100 65001 I  >> AS Path from leaf to border leaf
    Communities: target:1:8888 encapsulation:VXLAN(0x8)
* 2:172.16.7.1:1::51001::00:50:56:ab:01:01::10.10.0.101/304 MAC/IP (1 entry, 1 announced)
    Import Accepted
    Route Distinguisher: 172.16.7.1:1
    Route Label: 51001
    ESI: 00:00:88:88:88:88:88:88:01:01  >> the ESI used at the leaf1/leaf2 connecting server1
    Nexthop: 172.16.7.1
    AS path: 65100 65001 I
    Communities: target:1:8888 encapsulation:VXLAN(0x8)
```

To check how this control-plane EVPN advertisement is decoded inside the wireshark packet analyzer, we simply enable the port mirroring at the border-leaf2. The MAC@ 00:50:56:ab:01:01 of the server1, when received at the border-leaf2 from the spine2 is as shown in Figure 5.4.

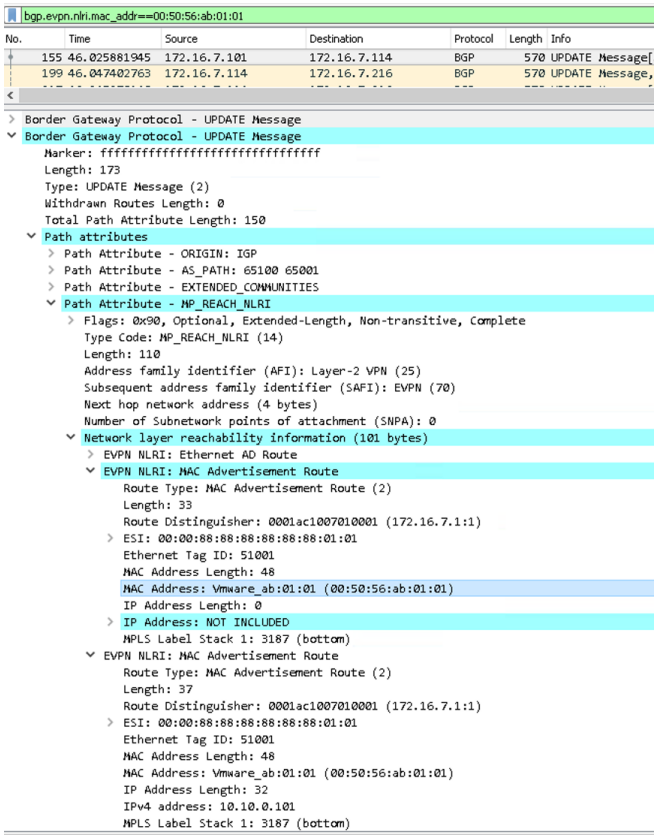


Figure 5.4 *Wireshark Decoded EVPN Route-Type-2 MAC/MAC-IP Received at The Border-leaf2 From Spine2 in DC-A , Originated at leaf1*

Now on Border leaf 2, let's check the routes advertised to Border leaf 4 for server 1 (mac@ 00:50:56:ab:01:01), when the site A originated MAC@ all behind the iESI 00:00:11:11:11:11:11:11:11:11.

```
root@border-leaf2> show route advertising-protocol bgp 172.16.7.216 evpn-mac-address
00:50:56:ab:01:01 table MACVRF101.evpn.0 detail
MACVRF101.evpn.0: 101 destinations, 102 routes (101 active, 0 holddown, 0 hidden)
* 2:172.16.7.114:101::51001::00:50:56:ab:01:01/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
    Route Distinguisher: 172.16.7.114:101
    Route Label: 61001  >> original VNI value is rewritten at the border-leaf for DCI purposes
    ESI: 00:00:11:11:11:11:11:11:11:11  >> ESI rewritten with Interconnect ESI value
    Nexthop: Self
    Flags: Nexthop Change
    AS path: [65000] I  >> AS Path rewritten with own ASN at border-leaf2/border-leaf1
    Communities: target:1:101 encapsulation:VXLAN(0x8)  >> the new route-target associated with the
EVPN MAC route
* 2:172.16.7.114:101::51001::00:50:56:ab:01:01::10.10.0.101/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
    Route Distinguisher: 172.16.7.114:101
    Route Label: 61001
    ESI: 00:00:11:11:11:11:11:11:11:11
    Nexthop: Self
    Flags: Nexthop Change
    AS path: [65000] I
    Communities: target:1:101 encapsulation:VXLAN(0x8)
```

The advertisement mentioned here with Junos output is also decoded with wireshark to precisely observe the attribute changes that happen at the border-leaf2 (172.16.7.114) when the server1 mac@ is advertised towards border-leaf4 (172.16.7.216). See Figure

You can see, for example, the ESI changes from
00:00:88:88:88:88:88:88:01:01
to the new interconnect ESI
00:00:11:11:11:11:11:11:11:11.
The same goes for the AS-PATH as well as next hop, both BGP attributes are changed for the EVPN route-type-2.

On Border-leaf2, we also verify the MAC-IP table. You can see the border-leaf2 is using the site B site identified iESI 00:00:22:22:22:22:22:22:22:22 to reach server4/server5.

From border-leaf2 and border-leaf1 perspective, server1/server2 continue to be reachable via the original server ESIs. Because server3 is single homed to leaf5, the ESI information is equal to all-zero, as per standard, and only the RVTEP IP@ information is registered.

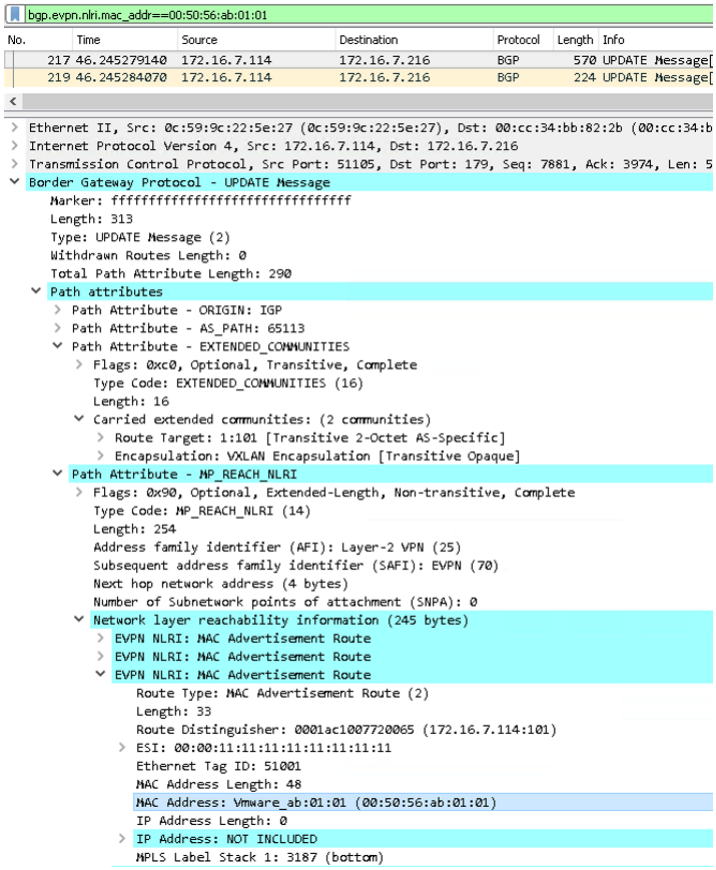


Figure 5.5 *Wireshark Decoded EVPN MAC Route-Type2 for Server1, Sent From Border-leaf2 to Border-leaf4 for DCI Purposes*

```
root@border-leaf2> show mac-vrf forwarding mac-table instance MACVRF101
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
         SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)
Ethernet switching table : 6 entries, 6 learned
Routing instance : MACVRF101
  VLAN             MAC              MAC     Logical             SVLBNH/     Active
  name             address           flags   interface           VENH Index   source
  VLAN1001          00:00:01:01:01:01   DRP     vtep.32771                     172.16.7.1
  VLAN1001          00:50:56:ab:01:01   DR      esi.640
00:00:88:88:88:88:88:88:01:01  >> Server 1
  VLAN1001          00:50:56:ab:01:02   DR      esi.641
00:00:88:88:88:88:88:88:02:02  >> Server 2
  VLAN1001          00:50:56:ab:01:03   DR      vtep.32774                     172.16.7.5
  VLAN1001          00:50:56:ab:01:04   DR      esi.647
00:00:22:22:22:22:22:22:22:22  >> Server4
  VLAN1001          00:50:56:ab:01:05   DR      esi.647
00:00:22:22:22:22:22:22:22:22  >> Server5
```

```
root@border-leaf2> show mac-vrf forwarding mac-ip-table instance MACVRF101 VLAN-name VLAN1001
MAC IP flags  (S - Static, D - Dynamic, L - Local , R - Remote, Lp - Local Proxy,
            Rp - Remote Proxy, K - Kernel, RT - Dest Route, (N)AD - (Not) Advt to remote,
            RE - Re-ARP/ND, RO - Router, OV - Override, Ur - Unresolved,
            RTS - Dest Route Skipped, RGw - Remote Gateway, FU - Fast Update)
 Routing instance : MACVRF101
 Bridging domain : VLAN1001
   IP                     MAC               Flags          Logical         Active
   address                 address                          Interface        source
   10.10.0.1               00:00:01:01:01:01   SR,K           vtep.32771       172.16.7.1
   10.10.0.101             00:50:56:ab:01:01   DR,K           esi.640
00:00:88:88:88:88:88:88:01:01  >> Server 1
   10.10.0.102             00:50:56:ab:01:02   DR,K           esi.641
00:00:88:88:88:88:88:88:02:02  >> Server 2
   10.10.0.103             00:50:56:ab:01:03   DR,K           vtep.32774       172.16.7.5
   10.10.0.104             00:50:56:ab:01:04   DR,K           esi.647
00:00:22:22:22:22:22:22:22:22   >> Server 4
root@border-leaf2>
```

On Border leaf 4, we check the type of tunnels created (WAN-VTEP for DCI, RVNE for fabric side of tunnels and I-ESI peer tunnel) and entries learned from the remote DCI gateway in the EVPN database with a new knob which helps to trace only the DCI-learned MAC addresses.

You can see, for example, at the border-leaf4 (DC site B), that server1/server2/server3 mac addresses are reachable via the site-A iESI 00:00:11:11:11:11:11:11:11:1, also used as the site identifier.

```
root@border-leaf4# run show mac-vrf forwarding vxlan-tunnel-end-point remote summary
Logical System Name      Id  SVTEP-IP        IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>              0   172.16.7.216    lo0.0    0
 RVTEP-IP     IFL-Idx    Interface    NH-Id   RVTEP-Mode  ELP-IP       Flags
 172.16.7.113   19309    vtep.32773   55058   Wan-VTEP
 172.16.7.114   19308    vtep.32772   55043   Wan-VTEP
 172.16.7.206   19307    vtep.32771   55022   RNVE
 172.16.7.215   19312    vtep.32774   55063   I-ESI-Peer
 RVTEP-IP       L2-RTT               IFL-Idx   Interface   NH-Id   RVTEP-Mode  ELP-IP       Flags
 172.16.7.114    MACVRF-techfest22      671547396 vtep-56.32772 55043  RNVE
 172.16.7.215    MACVRF-techfest22      671547398 vtep-56.32774 55063  I-ESI-Peer
 172.16.7.113    MACVRF101             671555589 vtep-57.32773 55058  Wan-VTEP  >> DCI tunnel
 172.16.7.114    MACVRF101             671555588 vtep-57.32772 55043  Wan-VTEP  >> DCI tunnel
 172.16.7.206    MACVRF101             671555587 vtep-57.32771 55022  RNVE
 172.16.7.215    MACVRF101             671555590 vtep-57.32774 55063  I-ESI-Peer
root@border-leaf4#
root@border-leaf4> show evpn database instance MACVRF101 origin dci-remote
Instance: MACVRF101
VLAN  DomainId  MAC address       Active source               Timestamp       IP address
    51001     00:00:01:01:01:01  00:00:11:11:11:11:11:11:11:11  May 31 06:35:20  10.10.0.1
    51001     00:50:56:ab:01:01  00:00:11:11:11:11:11:11:11:11  May 31 06:35:20  10.10.0.101  >>
Server 1
    51001     00:50:56:ab:01:02  00:00:11:11:11:11:11:11:11:11  May 31 06:35:20  10.10.0.102  >>
Server 2
    51001     00:50:56:ab:01:03  00:00:11:11:11:11:11:11:11:11  May 31 06:35:20  10.10.0.103  >>
Server 3
```

On Border leaf 4, let's check the routes received from Border leaf 2 for server 1:

```
root@border-leaf4> show route receive-protocol bgp 172.16.7.114 evpn-mac-address 00:50:56:ab:01:01
table MACVRF101.evpn.0 detail
MACVRF101.evpn.0: 68 destinations, 68 routes (68 active, 0 holddown, 0 hidden)
* 2:172.16.7.114:101::51001::00:50:56:ab:01:01/304 MAC/IP (1 entry, 1 announced)
    Import Accepted
    Route Distinguisher: 172.16.7.114:101
    Route Label: 61001 << the VXLAN VNI translation value from 51001 for DCI purposes
    ESI: 00:00:11:11:11:11:11:11:11:11  >> Interconnect ESI value rewritten by border leaf 2
    Nexthop: 172.16.7.114
    AS path: 65113 I >> ASN received is 65000 same as local-as so it's just set to internal flag I
    Communities: target:1:101 encapsulation:VXLAN(0x8)
* 2:172.16.7.114:101::51001::00:50:56:ab:01:01::10.10.0.101/304 MAC/IP (1 entry, 1 announced)
    Import Accepted
    Route Distinguisher: 172.16.7.114:101
    Route Label: 61001
    ESI: 00:00:11:11:11:11:11:11:11:11
    Nexthop: 172.16.7.114
    AS path: I >> ASN received is 65000 same as local-as so it's just set to internal flag I
    Communities: target:1:101 encapsulation:VXLAN(0x8)
```

On Border-leaf4, let's check the routes advertised to spine 4 for server1:

```
root@border-leaf4> show route advertising-protocol bgp 172.16.7.201 evpn-mac-address 00:50:56:ab:01:01
table MACVRF101.evpn.0 detail
MACVRF101.evpn.0: 68 destinations, 68 routes (68 active, 0 holddown, 0 hidden)
* 2:172.16.7.216:1::51001::00:50:56:ab:01:01/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
    Route Distinguisher: 172.16.7.216:1
    Route Label: >> VXLAN VNI value changed back from DCI VNI 61001 to Fabric VNI 51001
    ESI: 00:00:22:22:22:22:22:22:22:22  >> Interconnect ESI value rewritten by border leaf 4
    Nexthop: Self  >> protocol next-hop will be updated as well to the local loopback IP@
    Flags: Nexthop Change
    AS path: [65216] I >> ASN BGP rewrite to local fabric overlay value
    Communities: target:1:9999 encapsulation:VXLAN(0x8)
* 2:172.16.7.216:1::51001::00:50:56:ab:01:01::10.10.0.101/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
    Route Distinguisher: 172.16.7.216:1
    Route Label: 51001
    ESI: 00:00:22:22:22:22:22:22:22:22
    Nexthop: Self
    Flags: Nexthop Change
    AS path: [65215] I
    Communities: target:1:9999 encapsulation:VXLAN(0x8)
```

Once the rewriting of EVPN routes takes place at the border-leaf3/border-leaf4 from DCI to DC site in site B, leaf6 view for site A originated MAC@ is the following:

```
root@leaf6> show mac-vrf forwarding mac-table instance MACVRF101 vlan-id 1001
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
         SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)
Ethernet switching table : 7 entries, 7 learned
Routing instance : MACVRF101
Ethernet switching table : 7 entries, 7 learned
Routing instance : MACVRF101
```

```
  VLAN             MAC              MAC     Logical             SVLBNH/     Active
  name             address           flags   interface           VENH Index   source
  VLAN1001          00:00:01:01:01:01  DR      esi.1769            1792
00:00:22:22:22:22:22:22:22:22
  VLAN1001          00:50:56:ab:01:01  DR      esi.1769            1792
00:00:22:22:22:22:22:22:22:22
  VLAN1001          00:50:56:ab:01:02  DR      esi.1769            1792
00:00:22:22:22:22:22:22:22:22
  VLAN1001          00:50:56:ab:01:03  DR      esi.1769            1792
00:00:22:22:22:22:22:22:22:22
  VLAN1001          00:50:56:ab:01:04  DLR     ae0.0
  VLAN1001          00:50:56:ab:01:05  DLR     ae1.0
  VLAN1001          9c:8a:cb:05:64:00  DRP     vtep-9.32772                    172.16.7.207
{master:0}
root@leaf6>
```

You can see that server1/server2/server3 from leaf6's point of view are all reachable via border-leaf3/border-leaf4 iESI value 00:00:22:22:22:22:22:22:22:22 in site-B. So, even if we get significant number of MAC@ from site-A, they will all be reachable via the same pair of next hops.

In addition to the MACVRF mac@ learning state, we can verify if leaf6, thanks to the seamless stitching techniques, is not using any direct tunnel to the leaf1 which originated the MAC@ 00:50:56:ab:01:01 for server1 in DC site A:

```
root@leaf6> show mac-vrf forwarding VXLAN-tunnel-end-point esi
ESI                    RTT                 VLNBH INH    ESI-IFL   LOC-IFL   #RVTEPs
00:00:22:22:22:22:22:22:22:22 MACVRF101            1769  524291  esi.1769          2     Aliasing
RVTEP-IP          RVTEP-IFL     VENH    MASK-ID   FLAGS       MAC-COUNT
172.16.7.215       vtep-9.32770  1773    0       2           7
172.16.7.216       vtep-9.32771  1786    1       2           7
ESI                    RTT                 VLNBH INH    ESI-IFL   LOC-IFL   #RVTEPs
00:00:88:88:88:88:88:88:04:04 MACVRF101            1784  524286  esi.1784  ae0.0,   1     Aliasing
RVTEP-IP          RVTEP-IFL     VENH    MASK-ID   FLAGS       MAC-COUNT
172.16.7.207       vtep-9.32772  1760    0       2           1
ESI                    RTT                 VLNBH INH    ESI-IFL   LOC-IFL   #RVTEPs
00:00:88:88:88:88:88:88:05:05 MACVRF101            1785  524288  esi.1785  ae1.0,   1     Aliasing
RVTEP-IP          RVTEP-IFL     VENH    MASK-ID   FLAGS       MAC-COUNT
172.16.7.207       vtep-9.32772  1760    0       2           1
root@leaf6>
```

Leaf6 is not having any direct VXLAN tunnel to leaf1 or any other server leaf nodes on DC site A) . This behavior is expected in the case of seamless stitching enabled at the border-leaf nodes on each DC site.

Here leaf6 is still using EVPN-VXLAN but it uses its local border-leaf3 (172.16.7.215) and border-leaf4 (172.16.7.216) as next -hops for the server1 mac@00:50:56:ab:01:01 reachability.

The previous outputs help determine which RVTEPs IP addresses stand behind the esi.1769 logical interface and ESI 00:00:22:22:22:22:22:22:22:22. Leaf6 will continue to use direct tunnels to reach other leaf nodes located in its local DC.

For the cases where the destination mac@ is reachable via the local interface member, for example when server4 tries to communicate with server5, the local-bias will be used instead of the EVPN-VXLAN tunnel. This is because both leaf6 and leaf7 are multihomed to these two servers. The default behavior is to prefer local bias forwarding.

On the data plane side, server1(00:50:56:ab:01:01) packet destined to server4 (00:50:56:ab:01:04) is first received by leaf1 (172.16.7.1) and then sent to border-leaf1 (172.16.7.113).

From leaf1 to border-leaf1/2, you can see that the customer vlan-id is stripped before the layer 2 ethernet frame is pushed into the tunnel. For border-leaf1/2, it means the processing of the data packet is based on the ingress VNI value. In the example of VLAN1001, we see the packet has VNI 5100.

Figure 5.6 a pcap data frame captured at the border-leaf1 level before the stitching to DCI VX- LAN tunnel takes place:

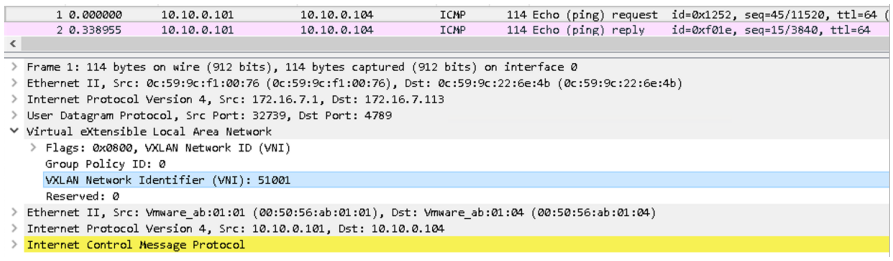


Figure 5.6 VXLAN Data Packet Destined to Server4 in DC Site B Sourced in DC A On Server1 - Before Stitching

Because the packet is destined to server4 in DC site B, the border-leaf1 decapsulates the packet received from leaf1 and encapsulates it into the new WAN VXLAN destined to site B border-leaf3 (172.16.7.215), performing the tunnel stitching function (we can see the VXLAN VNI changed to 61001), as shown in Figure 5.7.

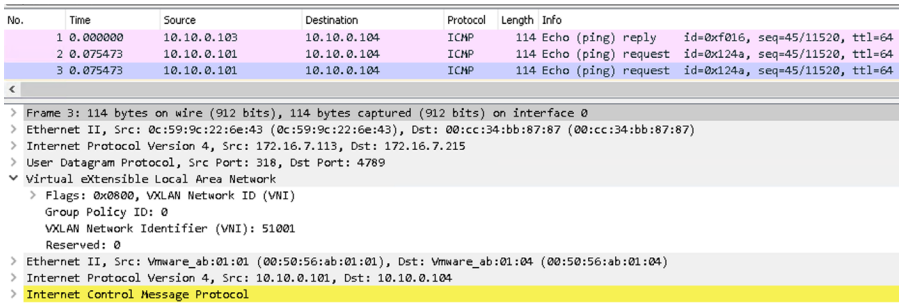


Figure 5.7 VXLAN Data Packet Destined To Server4 Sourced On Server1 - After Stitching To Interconnect VXLAN At The Border-leaf1

When more in-depth verification is required at the border-leaf3/4, which in our lab is the QFX5130-32cd switch, we can verify if the DCI reachable MAC@ is installed in PFE (Packet Forwarding Engine):

```
root@border-leaf3> start shell user root
[VRF:none] root@border-
leaf3:~# # >>we will enter the pfe cli at the Junos Evolved based switches such as the qfx5130-32cd
[VRF:none] root@border-leaf3:~# cli-pfe
root@border-leaf3:pfe>
root@border-leaf3:pfe>
root@border-leaf3:pfe> show l2 manager bridge-domains
index      name                            rtt index    flags
-----------------------------------------------------------
2          VLAN1001+1001                   51           0x0
index      name                            rtt index    flags
-----------------------------------------------------------
5          VLAN1002+1002                   51           0x0
index      name                            rtt index    flags
-----------------------------------------------------------
6          default+1                       50           0x0
root@border-leaf3:pfe>
root@border-leaf3:pfe>
root@border-leaf3:pfe> show evo-pfemand layer2 bd-table
-------------- ----------
Layer2 Token   Bd-Id
-------------- ----------
4096           2
4097           5
1              6
root@border-leaf3:pfe> show evo-pfemand layer2 mac-fdb-table
       Mac Address   Vlan       Port  isLag  isStatic   HW Synced    HW Prog  isRemote
   00:00:01:01:01:01   4096  0xb0000104  False    False         1         1     True
   00:50:56:ab:01:01   4096  0xb0000104  False    False         1         1     True  >> server1 MAC@
at the pfe level
   00:50:56:ab:01:02   4096  0xb0000104  False    False         1         1     True
   00:50:56:ab:01:03   4096  0xb0000104  False    False         1         1     True
   00:50:56:ab:01:04   4096  0xb0000103  False    False         1         1     True
   00:50:56:ab:01:05   4096  0xb0000103  False    False         1         1     True
   18:2a:d3:57:b1:80   4096  0xb0000102  False    False         1         1     True
   9c:8a:cb:05:64:00   4096  0xb0000103  False    False         1         1     True
   00:00:01:01:01:02   4097  0xb0000104  False    False         1         1     True
   18:2a:d3:57:b1:80   4097  0xb0000102  False    False         1         1     True
   9c:8a:cb:05:64:00   4097  0xb0000103  False    False         1         1     True
   c0:03:80:1c:7e:e0   4097  0xb0000104  False    False         1         1     True
   c0:03:80:1c:b5:e0   4097  0xb0000104  False    False         1         1     True
Global Mac Table Size : d
root@border-leaf3:pfe>
root@border-leaf3:pfe> show evo-pfemand virtual vtep
vtep-ifl(Count)   ifl-idx       ifl-name      ip           vtepFlavor    vPortToken
isTunnelInstalled   ucTunnelId    ucEgressIf
----------------- -------------- -------------- -------------- -------------- -------------- --------
----------- --------------- ---------------
source-ifl(3)    19304        vtep         172.16.7.215   Default      0x0          Yes
0x0          0x206a1 (Discard)
remote-ifl(1)    19305        vtep.32770    172.16.7.206   Default      0xb0000102    Yes
0x4c100102    0x206a2
remote-ifl(1)    19306        vtep.32771    172.16.7.207   Default      0xb0000103    Yes
```

```
0x4c100103    0x206a2
remote-ifl(1)    19307        vtep.32772    172.16.7.113   DCI-WAN       0xb0000104    Yes
0x4c100104    0x206a3
root@border-leaf3:pfe>
```

To verify which physical interface is associated with the ifl-index 19307 used for VXLAN tunneling, you can check this recursively at QFX5130 border-leaf3/4 using the following command:

```
root@border-leaf3:pfe> show evo-pfemand virtual vtep iflIndex 19307
vtep-ifl(Count)   ifl-idx       ifl-name      ip           vtepFlavor    vPortToken
isTunnelInstalled   ucTunnelId    ucEgressIf
----------------- -------------- -------------- -------------- -------------- -------------- --------
----------- -------------- --------------
remote-ifl(1)    19307        vtep.32772    172.16.7.113   DCI-WAN       0xb0000104    Yes
    0x4c100104    0x206a3
Tunnel SIP: 172.16.7.215
Tunnel DIP: 172.16.7.113
Source Vtep's IP: 172.16.7.215
VENH info
----------
Sw VTEP Ref Count: 1
Sw Venh Id: 57050
Hw Venh Id: 132771(0x206a3)
Hw Venh Id Ref count: 1
Underlay forwarding interface (Unicast):
interface: et-0/0/24.0, (port-id: 120)
Multihoming Local Bias Filter Info:
-----------------------------------
IfpVXLAN-EsiLB Filter is not Installed
Vtep Local Bias Info:
---------------------------
No rvtep is enabled for local bias
root@border-leaf3:pfe>
```

Realize that for the RVTEP tunnel destination 172.16.7.113, the interface et-0/0/24 is used as an outgoing physical interface. As we can see in the output, the DCI-WAN flag is also associated with the given tunnel IFL, indicating it's going to be used for DCI purposes.

This type of in-depth verification is not required on a day-to-day basis, we introduce it here as an additional reference for the verification tasks related to the data center interconnect, when using QFX5130-32cd as border-leaf node.

Type-5 EVPN-VXLAN to Type-5 EVPN-VXLAN - Implementation and Verification

In the previous Chapter 4 we focused on how to deliver Layer 2 DCI seamless connectivity with the EVPN-VXLAN tunnel stitching for the traffic within the same VLAN/bridge-domain. But there are situations where there's no requirement for L2 stretch and some of the tenant VLANs and IP prefixes are DC site specific.

In these situations, some pairs of leaf nodes will be enabled with rack specific IP subnet at the IP VRF Type-5 routing-instance level and advertise their reachability using pure Type-5 EVPN routes. This helps scaling out the fabric by advertising only the Type-5 LPM IP prefix instead of each MAC, MAC-IP advertisement.

Besides the intra-fabric scenario where you restrain the MAC/MAC-IP EVPN route propagation between the ToR nodes, you can have a requirement where the fabric in a given site has Type-2 MAC/MAC-IP as well as Type-5 EVPN advertisements. However, for the DCI purposes, only the Type-5 IP prefix advertisements are required for full reachability between the data center sites.

For such scenarios, in larger scale deployments and when more operational control is required, then similarly as described in Chapter 4 for Type-2 stitching, you can perform Type-5 tunnel stitching and originate, at the border-leaf level, a new DCI Type-5 tunnel to the remote site after terminating the local Type-5 tunnels.

Let's compare the pure Type-5 stitching to Type-2 EVPN VXLAN tunnel stitching and highlight the differentiators:

- There's no IFL tunnel interface creation in case of pure type-5 tunnels (interface-less) and the IP traffic is directed to tunnels-based overlay composite next hop resolution
- The load balancing for pure Type-5 tunnels for IP prefix reachability is based on overlay IP ECMP and not using the iESI which was used in case of layer 2 (MAC) stitching
- The VNI value change/rewrite is not performed and the same routing VNI value is used for DCI purposes as well as LAN fabric
- The router MAC (RMAC) is changed in Type-5 EVPN to the local border-leaf value

Similar to the Type-2 EVPN-VXLAN stitching implementation, a new Route-Target (iRT), new Route-Distinguisher(iRD), and IP next hop change is done at the border-leaf level nodes in each DC site.

When it comes to the allocation of the route-targets – it's recommended to consider local fabric Type-5 route-targets to be unique per site within the given T5 IPVRF and then use a common route-target for DCI purposes in the interconnect section. Each node will obviously have a different route-distinguisher for fabric and DCI purposes.

From the design point of view, when the distributed IP gateway at the leaf/ToR is deployed within the DC site, then often the Type-2 and Type-5 tunnel stitching will be conducted in parallel to better scale the solution and keep it consistent for Layer 2 as well as layer 3 DCI.

As the pure Type-5 interface-less EVPN routes are not using an ESI value, for site identification purposes, it is highly recommended to consider the new BGP attribute called

D-PATH – domain path. This attribute will identify the EVPN prefix origin and help avoid routing loops between the DC sites or any sub-optimal forwarding. Each Domain segment is comprised of <domain segment length, domain segment value>, where the domain segment value is a sequence of one or more domains. All domain IDs included in the D-PATH BGP attribute are compared with the local domain IDs after the route-target verification passed.

Managing the DCI solution using traditional communities can become too complex when dealing with many T5-IPVRFs, so the new D-PATH approach is considered more user friendly in terms of implementation.

Here's an example of how to enable the D-PATH at the border-leaf level in site A – same domain-id 101:1:EVPN will be enabled at border-leaf1/border-leaf2.

Config 1 D-PATH BGP attribute configuration for site A

```
set routing-options uniform-propagation-mode domain-id 101:1:EVPN
```

The same approach is taken for site B border-leaf3/border-leaf4 to enable the BGP D-PATH attribute:

Config 2 D-PATH BGP attribute configuration for site B

```
set routing-options uniform-propagation-mode domain-id 101:2:EVPN
```

To better understand the implementation and verification part of pure Type-5 tunnel stitching the following lab topology is introduced in Figure 5.8.

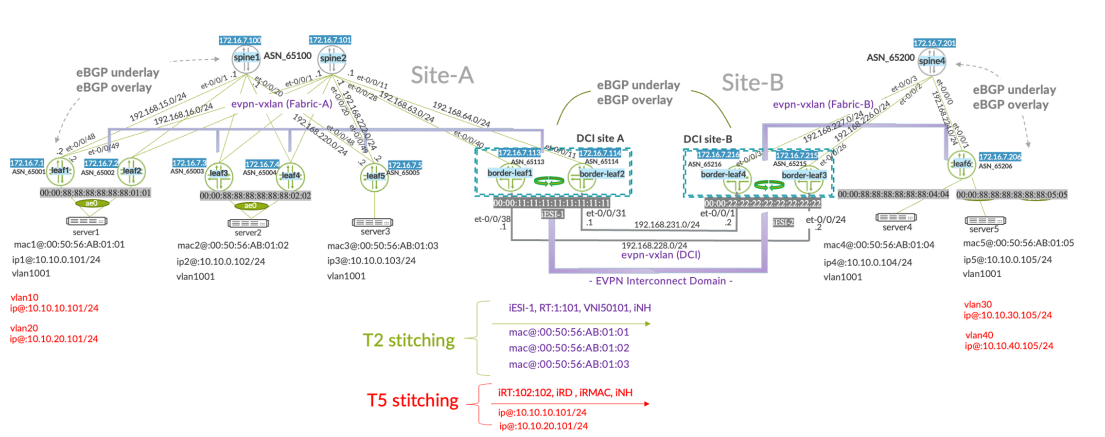


Figure 5.8 Seamless EVPN-VXLAN Stitching Lab for Pure Type-5 Tunnel Stitching

You can see in Figure 5.8 that VLAN10/VLAN20 is only present in DC site A and VLAN30/VLAN40 is specific to site B, while the VLAN1001 is still stretched between the two data centers sites.

The IP prefixes associated with the site specific VLANs will be advertised to the remote site, in this case, and not the MAC/MAC-IP information.

From the BGP underlay and overlay implementation point of view, in the case of pure Type-5 stitching, you continue using the same principle described in more details in the previous chapter. Both border-leaf nodes will be using the overlay eBGP ASN numbers and make sure optimal routes are installed and the default eBGP routing loop prevention mechanisms are in place. This means when, for some reasons, an EVPN prefix originated in site A is sent back as advertisement from the site B, then the local border-leaf1/border-leaf2 will drop that advertisement by default and only the remote site originated prefixes will be installed in the RIB.

The VRF routing-instance is used for pure Type-5 tunnel stitching with the additional interconnect block.

Config 3 Pure Type-5 EVPN-VXLAN tunnel stitching - border-leaf1 in site A

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.113/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn interconnect vrf-target target:102:102  >> common iRT
value in site A and site B
set routing-instances T5-VRF1 protocols evpn interconnect route-distinguisher 172.16.7.113:102  >>
unique iRD per node
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100  >> routing VNI stays the same
per Type-5 IPVRF
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1  >> to define
prefixes to be advertised
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.113:100  >> unique RD per node
set routing-instances T5-VRF1 vrf-target target:1100:1100  >> site local Type-5 route-target specific
to site A
set routing-instances T5-VRF1 vrf-table-label
set interfaces lo0 unit 1 family inet address 172.16.100.113/32
```

The policy-statement is used to explicitly define the prefixes local to the site and remote prefixes which will be advertised towards the local site leaf as well as to the remote locations.

Config 4 Type-5 routing policy-statement for border-leaf1/border-leaf2 stitching in site A

```
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.113/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term1 then accept
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.113/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.30.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.40.0/24
orlonger
```

```
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.10.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.20.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
```

The border-leaf2 from site A follows the same approach as border-leaf1 for the pure Type-5 VRF implementation. Only the local RD/iRD information is changed.

Config 5 Pure Type-5 EVPN-VXLAN tunnel stitching - border-leaf2 in site A

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.114/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn interconnect vrf-target target:102:102  >> iRT Type-5
stitching
set routing-instances T5-VRF1 protocols evpn interconnect route-distinguisher 172.16.7.114:102  >> iRD
unique per node
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100  >> same value in site A and
site B is used
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.114:100  >> site local RD
set routing-instances T5-VRF1 vrf-target target:1100:1100  >> site local RT
set routing-instances T5-VRF1 vrf-table-label
set interfaces lo0 unit 1 family inet address 172.16.100.114/32
```

At the border-leaf2 the same policy-statement configuration is used as highlighted above for border-leaf1 – we explicitly define all the IP EVPN prefixes that will be advertised within the site and between the sites.

Once the site A was provisioned with Type-5 stitching at the 'EVPN interconnect' level inside the Type-5 IPVPN, we continue enabling it in site B at border-leaf3/border-leaf4, using the same common DCI iRT target:102:102.

Config 6 Pure Type-5 EVPN-VXLAN tunnel stitching - border-leaf3 in site B

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.215/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn interconnect vrf-target target:102:102  >> iRT value
shared with site A
set routing-instances T5-VRF1 protocols evpn interconnect route-distinguisher 172.16.7.215:102  >> iRD
unique per node
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.215:100  >> RD unique per node
set routing-instances T5-VRF1 vrf-target target:1200:1200  >> site local RT specific to DC site B
set routing-instances T5-VRF1 vrf-table-label
set interfaces lo0 unit 1 family inet address 172.16.100.215/32
```

Config 7 Type-5 routing policy-statement for border-leaf3/border-leaf3 stitching in site B

```
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.215/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term1 then accept
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.215/32 exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.10.0/24 orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.20.0/24 orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.30.0/24 orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.40.0/24 orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
```

After all the IPVRF Type-5 EVPN implementation, the leaf1 from site A won't create direct Type-5 tunnels towards site B but rather go through the border-leaf1/border-leaf2 first and just install the two next-hops to his local site border-leaf nodes.

This is something you can verify using the following command at the border-leaf1 level.

Config 8 Verifying the pure Type-5 stitching IP prefix status in site A

```
root@border-leaf1# run show evpn ip-prefix-database l3-context T5-VRF1 extensive prefix 10.10.30.0/24
L3 context: T5-VRF1
IPv4->EVPN Exported Prefixes
Prefix: 10.10.30.0/24
  EVPN route status: Created  >> will be advertised to leaf1/leaf2/leaf3 in site A
  Change flags: 0x0
  Advertisement mode: Direct nexthop
  Encapsulation: VXLAN
  VNI: 1100
  Router MAC: 0c:59:9c:22:6d:a2  >> local router MAC (RMAC) corresponding to IRB MAC/chassis-mac@
EVPN->IPv4 Imported Prefixes
Prefix: 10.10.30.0/24, Ethernet tag: 0
  Change flags: 0x0
  Remote advertisements:
    Route Distinguisher: 172.16.7.215:102
      VNI: 1100
      Router MAC: 00:cc:34:bb:8c:2d  >> RMAC info received from border-leaf3
      BGP nexthop address: 172.16.7.215  >> next-hop IP@ of the border-leaf3
      IP route status: Created
root@border-leaf1#

root@border-leaf1# run show interfaces irb | match Hardware
  Current address: 0c:59:9c:22:6d:a2, Hardware address: 0c:59:9c:22:6d:a2  >> border-leaf1 uses it for
RMAC in Type-5

root@border-leaf1#
root@border-leaf3# run show interfaces irb | match Hardware
  Current address: 00:cc:34:bb:8c:2d, Hardware address: 00:cc:34:bb:8c:2d  >> border-leaf3 uses it for
RMAC in Type-5 route
root@border-leaf3#
```

The site B border-leaf3 view of remote prefix is as follows when it comes to RMAC and IP next-hops.

Config 9 Verifying the pure Type-5 stitching IP prefix status in site B

```
root@border-leaf3> show evpn ip-prefix-database l3-context T5-VRF1 prefix 10.10.10.0/24
L3 context: T5-VRF1
IPv4->EVPN Exported Prefixes
Prefix                              EVPN route status
10.10.10.0/24                        Created
EVPN->IPv4 Imported Prefixes
Prefix                              Etag
10.10.10.0/24                        0
  Route distinguisher   VNI/Label  Router MAC       Nexthop/Overlay GW/ESI   Route-Status  Reject-
Reason
  172.16.7.113:102      1100      0c:59:9c:22:6d:a2  172.16.7.113           Accepted     n/a
root@border-leaf3> show evpn ip-prefix-database l3-context T5-VRF1 prefix 10.10.10.0/24 extensive
L3 context: T5-VRF1
IPv4->EVPN Exported Prefixes
Prefix: 10.10.10.0/24  >> will be advertised to leaf6  in site B with new RMAC and new iNH
  EVPN route status: Created
  Change flags: 0x0
  Advertisement mode: Direct nexthop
  Encapsulation: VXLAN
  VNI: 1100
  Router MAC: 00:cc:34:bb:8c:2d  >> sets the RMAC corresponding to his local IRB hardware MAC@ value
EVPN->IPv4 Imported Prefixes
Prefix: 10.10.10.0/24, Ethernet tag: 0
  Change flags: 0x0
  Remote advertisements:
    Route Distinguisher: 172.16.7.113:102
      VNI: 1100
      Router MAC: 0c:59:9c:22:6d:a2  >> Type-5 prefix from site A is received with remote border-leaf1
RMAC
      BGP nexthop address: 172.16.7.113
      IP route status: Created
root@border-leaf3>
```

After the reception of the prefix 10.10.10.0/24 from site A at the border-leaf3 in site B, you can check if that prefix is advertised also to the route server spine4 and then to server leaf6.

Config 10 Pure Type-5 verification of routes advertised to spine4 (local site route-server/reflector)

```
root@border-leaf3> show route advertising-protocol bgp 172.16.7.201 table T5-VRF1.
evpn.0 | match 10.10.
  5:172.16.7.215:100::0::10.10.10.0::24/248
  5:172.16.7.215:100::0::10.10.20.0::24/248
  5:172.16.7.215:100::0::10.10.10.1::32/248
  5:172.16.7.215:100::0::10.10.10.101::32/248
  5:172.16.7.215:100::0::10.10.20.1::32/248
  5:172.16.7.215:100::0::10.10.20.101::32/248
  5:172.16.7.215:102::0::10.10.100.215::32/248
root@border-leaf3>
root@border-leaf3> show route advertising-protocol bgp 172.16.7.201 table T5-VRF1.evpn.0 match-prefix
5:172.16.7.215:100::0::10.10.10.0::24/248 extensive
T5-VRF1.evpn.0: 30 destinations, 30 routes (30 active, 0 holddown, 0 hidden)
* 5:172.16.7.215:100::0::10.10.10.0::24/248 (1 entry, 1 announced)
```

```
 BGP group overlay type External
    Route Distinguisher: 172.16.7.215:100
    Route Label: 1100
    Overlay gateway address: 0.0.0.0
    Nexthop: Self
    Flags: Nexthop Change
    AS path: [65215] 65113 65100 65001 I
    Communities: target:1200:1200 encapsulation:vxlan(0x8) router-mac:00:cc:34:bb:8c:2d
root@border-leaf3>
```

All the IP Prefix advertisements between the sites happen because the server leaf1 in DC site A and leaf6 in DC site B are also provisioned with Type-5 IP routing instances with the route-targets unique per site. Each of the server leafs advertise to the local spine route servers, the local IRB interfaces corresponding IP prefix as Type-5 EVPN route, which is then received at the border-leaf and re-originated with some of the EVPN BGP attribute changes.

Here's an example of the Type-5 instance configuration from leaf1 in site A where the prefix 10.10.10.0/24 originated.

Config 11 Type-5 IP VRF at the server leaf1 in site A

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.1/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1
set routing-instances T5-VRF1 interface irb.10
set routing-instances T5-VRF1 interface irb.20
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.1:100
set routing-instances T5-VRF1 vrf-target target:1100:1100
set routing-instances T5-VRF1 vrf-table-label
set interfaces lo0 unit 1 family inet address 172.16.100.1/32
```

The target:1100:1100 from leaf1 is corresponding to border-leaf1/border-leaf2 site local RT, which is different from the iRT used only for DCI purposes. To make sure we only advertise at the server leaf1 the local IRB interface corresponding prefixes, as well as the static route and loopback0.1 per VRF, the following routing policy statement is introduced at the server leaf1 in site A.

Config 12 Type-5 dedicated routing policy at server leaf1in site A

```
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.1/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term1 then accept
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.1/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.10.0/24
orlonger
```

```
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.20.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
```

For the site B server leaf6 the configuration template is very similar, however as you can see below the site local route-target (RT) is target:1200:1200 and corresponds to the one used at border-leaf3/border-leaf4 in site B. This RT is not the one used for the DCI purposes target:102:102, which is only used at all the border-leaf nodes communicating to each other through Type-5 EVPN advertisement.

Config 13 Type-5 IPVRF at the server leaf6 in site B

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.206/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1
set routing-instances T5-VRF1 interface irb.30  >> Integrated Routing-Bridging interface corresponding
to VLAN30
set routing-instances T5-VRF1 interface irb.40  >> Integrated Routing-Bridging interface corresponding
to VLAN40
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.206:100
set routing-instances T5-VRF1 vrf-target target:1200:1200
set routing-instances T5-VRF1 vrf-table-label
set interfaces lo0 unit 1 family inet address 172.16.100.206/32
```

The leaf6 also defines explicitly which prefixes it will precisely be advertising to this local spine4 route-server and then to the border-leaf3/border-leaf4.

Config 14 Type-5 policy-statement to define which prefixes are advertised at the server leaf6 level in DC site B

```
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.206/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term1 then accept
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.206/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.40.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.30.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
```

The additional prefix advertisement is optional for loopback0.1 and static route. It's introduced here just to show how to advertise prefixes which are not specific to any IRB interface connected.

The keyword `orlonger` inside the policy statement is optional but used here in order to also advertise the host route Type-5 corresponding to the server5. This means the border-leaf will receive not only an LPM entry but also the most specific server5 prefix

10.10.30.105/32. This is something optional but useful when the same subnet is enabled on multiple top of racks, because the border-leaf will know exactly which of the server leaf nodes to forward traffic coming from the core-IP. In case the subnet 10.10.30.0/24 was only enabled at leaf6, then obviously the host route 10.10.30.105/32 advertisement would not be needed for the full IP reachability.

Once this is in place, we are able to see how the server leaf6 in site B processes the prefix 10.10.10.0/24 originated in site A at the server leaf1.

Config 15 Type-5 EVPN prefix database verification at leaf6 in site B

```
root@leaf6# run show evpn ip-prefix-database l3-context T5-VRF1 prefix 10.10.10.0/24
L3 context: T5-VRF1
EVPN->IPv4 Imported Prefixes
Prefix                                Etag
10.10.10.0/24                          0
  Route distinguisher   VNI/Label  Router MAC       Nexthop/Overlay GW/ESI   Route-Status  Reject-
Reason
  172.16.7.215:100      1100      00:cc:34:bb:8c:2d  172.16.7.215            Accepted     n/a
  172.16.7.216:100      1100      00:cc:34:bb:87:2d  172.16.7.216            Accepted     n/a
{master:0}[edit]
root@leaf6#
root@leaf6# run show evpn ip-prefix-database l3-context T5-VRF1 prefix 10.10.10.0/24 extensive
L3 context: T5-VRF1
EVPN->IPv4 Imported Prefixes
Prefix: 10.10.10.0/24, Ethernet tag: 0
  Change flags: 0x0
  Remote advertisements:
    Route Distinguisher: 172.16.7.215:100
      VNI: 1100
      Router MAC: 00:cc:34:bb:8c:2d
      BGP nexthop address: 172.16.7.215
      IP route status: Created
    Route Distinguisher: 172.16.7.216:100
      VNI: 1100
      Router MAC: 00:cc:34:bb:87:2d
      BGP nexthop address: 172.16.7.216
      IP route status: Created
{master:0}[edit]
root@leaf6#
```

We can also verify the tunnel installation itself from the border-leaf3 and border-leaf4 point of view by entering the PFE (Packet Forwarding Engine) CLI and calling the `evo-pfemand` commands.

Config 16 EVPN-VXLAN tunnel PFE installation verification at qfx5130

```
[VRF:none] root@border-leaf3:~# cli-pfe
root@border-leaf3:pfe> show evo-pfemand virtual vtep
vtep-ifl(Count)   ifl-idx       ifl-name      ip           vtepFlavor    vPortToken
isTunnelInstalled   ucTunnelId    ucEgressIf
----------------- -------------- -------------- -------------- -------------- -------------- --------
source-ifl(1)     19304         vtep          172.16.7.215   Default       0x0          Yes
```

```
0x0          0x206a1 (Discard)
remote-ifl(1)    19305        vtep.32770    172.16.7.206   Default       0xb0000202    Yes
0x4c100202    0x206a2
remote-ifl(1)    19306        vtep.32771    172.16.7.113   Default       0xb0000203    Yes
0x4c100203    0x206a3
remote-ifl(1)    19307        vtep.32772    172.16.7.216   DCI-PEER-GW    0xb0000206    Yes
0x4c100206    0x206a2
remote-ifl(1)    0           tunnelType5   172.16.7.113   Default       0xb0000205    Yes
0x4c100205    0x206a3
remote-ifl(1)    0           tunnelType5   172.16.7.206   Default       0xb0000204    Yes
0x4c100204    0x206a2
root@border-leaf3:pfe>
root@border-leaf4:pfe> show evo-pfemand virtual vtep
vtep-ifl(Count)   ifl-idx       ifl-name      ip           vtepFlavor    vPortToken
isTunnelInstalled   ucTunnelId    ucEgressIf
----------------- -------------- -------------- -------------- -------------- -------------- --------
source-ifl(1)    19304        vtep         172.16.7.216   Default       0x0          Yes
0x0          0x206a1 (Discard)
remote-ifl(1)    19306        vtep.32770    172.16.7.215   DCI-PEER-GW    0xb0000204    Yes
0x4c100204    0x206a2
remote-ifl(1)    19308        vtep.32771    172.16.7.206   Default       0xb0000205    Yes
0x4c100205    0x206a2
remote-ifl(1)    19309        vtep.32772    172.16.7.114   DCI-WAN       0xb0000206    Yes
0x4c100206    0x206a3
remote-ifl(1)    0           tunnelType5   172.16.7.114   Default       0xb0000203    Yes
0x4c100203    0x206a3
remote-ifl(1)    0           tunnelType5   172.16.7.206   Default       0xb0000202    Yes
0x4c100202    0x206a2
root@border-leaf4:pfe>
```

Based on the EVPN Type-5 route control plane exchanged with the remote border-leaf1 (172.16.7.113) and local site B leaf6 (172.16.7.206), we can clearly see the type of tunnels (tunnelType5) installed in the forwarding table of the Trident4 chip. The same output shows the other type of Type2 tunnels which starts with the name vtep.xxx. The key difference we can see is related to the fact that the tunnelType5 is not really attached to any IFL (logical interface) index while the type-2 EVPN VXLAN tunnels create in the PFE also a logical interface instance.

After all that detailed verification now we need to go to basics and at least run a basic ping from site A to site B between server1 IP@10.10.10.101 and server5 IP@10.10.30.105.

Config 17 leaf6 ping based forwarding verification from DC site B to site A

```
root@server1# run ping 10.10.30.105 source 10.10.10.101
PING 10.10.30.105 (10.10.30.105): 56 data bytes
64 bytes from 10.10.30.105: icmp_seq=0 ttl=60 time=1.873 ms
64 bytes from 10.10.30.105: icmp_seq=1 ttl=60 time=11.133 ms
64 bytes from 10.10.30.105: icmp_seq=2 ttl=60 time=28.586 ms
64 bytes from 10.10.30.105: icmp_seq=3 ttl=60 time=11.155 ms
^C
--- 10.10.30.105 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.873/13.187/28.586/9.663 ms

{master:0}[edit]
root@server1# >
```

The format of the data packet received at border-leaf1 via pure Type-5 EVPN-VXLAN tunnel from the server leaf1 is shown below. We can see that the MAC@ used inside the tunnel are the chassis IRB MAC of the leaf1 and border-leaf1 instead of the original server1 and server5 – this is based on the RMAC community advertisement inside EVPN Type-5 route. This way DC site A to DC site B for server1 to server5 communication only need to advertise the Pure type-5 prefixes instead of the end host MAC addreses, which would be the case of the L2 stretched DCI with Type-2/Type-2 EVPN-VXLAN stitching – covered in previous chapter.

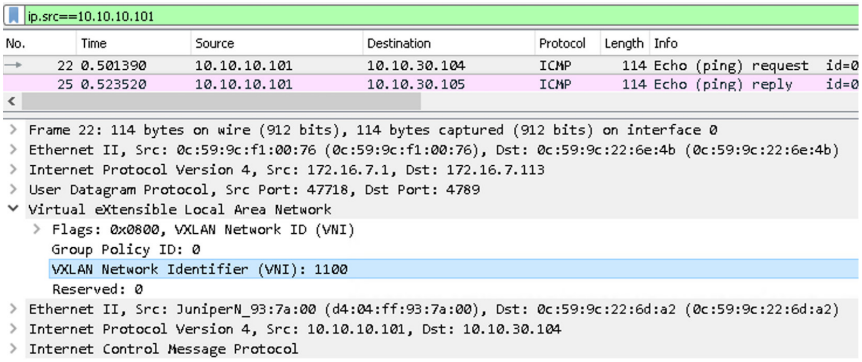


Figure 5.9 *Type-5 EVPN-VXLAN tunnel encapsulated data packet sent from leaf1 to border-leaf1 – before pure Type-5 tunnel stitching on border-leaf1*

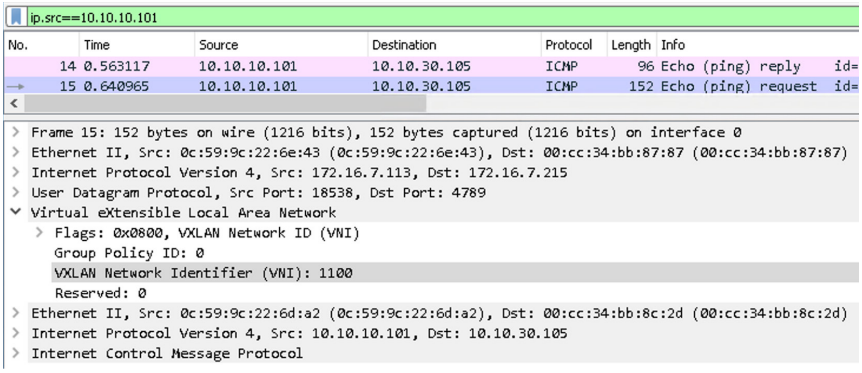


Figure 5.10 *Type-5 EVPN-VXLAN tunnel encapsulated data packet sent from border-leaf1 to border-leaf3 – after pure Type-5 tunnel stitching on border-leaf1*

After the Type-5 stitching on border-leaf1 we can also observe the changes on the data packet format – a new RMAC (Router MAC) information is used (the one from border-leaf nodes). The figure above shows the data plane result of the tunnel stitching at the border-leaf1 for pure Type-5 tunnels – fabric T5 tunnel received from server leaf1 is stitched to the DCI Type-5 tunnel destined to the remote location border-leaf3.

Chapter 6

Seamless EVPN-VXLAN to EVPN-MPLS Stitching - Implementation and Verification

The cool thing about the new RFC9014 seamless stitching techniques is that it can be used in different types of transport requirements. In fact, besides the EVPN-VXLAN to EVPN-VXLAN stitching scenario covered in previous sections, the second very popular scenario for DCI is when the same EVI is used to seamlessly deliver the Ethernet bridging between multiple data center sites, using VXLAN in the LAN DC and MPLS in the WAN for DCI purposes.

This is something we can explain in detail using the following lab scenario where the intermediate EVPN domain is using the MPLS encapsulation while the local site domains are using EVPN-VXLAN.

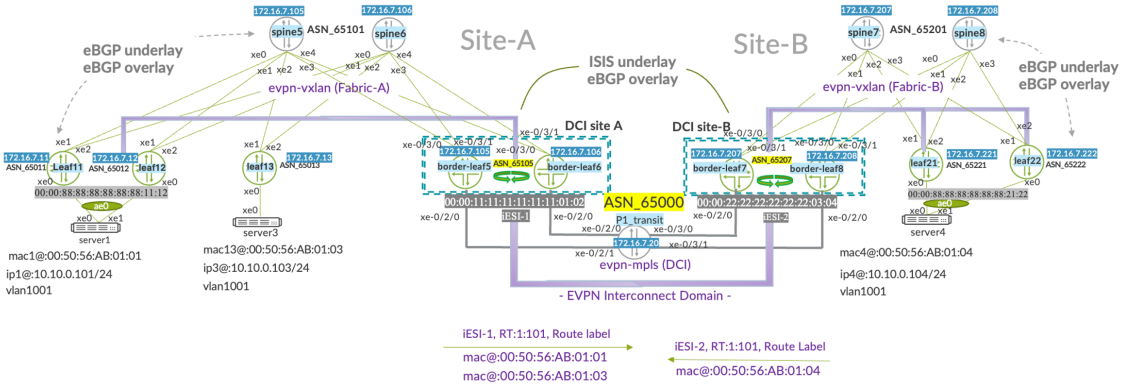


Figure 6.1 *DCI EVPN-VXLAN to EVPN-MPLS Seamless Tunnel Stitching - Lab Topology*

In the scenario shown in Figure 6.1, the border-leaf nodes, aka gateways, are part of the fabric and will seamlessly change the transport format from VXLAN to MPLS. This will also be done automatically using built-in BGP EVPN attributes. We'll use this topology for further configuration discussion and verifications.

Before we go into the details, it's also worth reminding the reader of situations this option can be considered. Here are the most common reasons some organizations may prefer using the MPLS versus VXLAN in the DCI scenario:

- Existing core network is already running the MPLS, and the P router connectivity is directly available in the DC room.
- Smaller DC fabrics satellite sites are deployed in regions and adding a separate PE to connect to MPLS is not cost effective.
- Existing PE devices are integrated as border-leaf nodes into the fabric, and they already have the long reach MPLS connectivity to remote sites.
- VPLS was used so far as the DCI technique so the migration to EVPN-MPLS DCI can offer fast convergence capabilities.

BGP Peers Connecting to the Existing Backbone IP MPLS

From the IP routing design point of view, the solution will be using ISIS dynamic routing protocol for the underlay core IP. It will be enabled only at the border-leaf level and for Level 2.

The core IP already has an existing route reflector P1_transit which will be also used for overlay iBGP peering purposes with new family EVPN signalization. A common overlay iBGP ASN number 65000 will be enabled between the border-leaf nodes in site A and site B.

P1_transit is playing the role of the P router. Even if border-leaf nodes have a single physical connection to the MPLS core, they share the same interconnect iESI value, so the EVPN aliasing will be used for packet load balancing.

Here are the example configurations used for the underlay DCI routing at the border-leaf5/6.

Config 18 underlay ISIS configuration at border-leaf5

```
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
set protocols isis level 1 disable
set interfaces lo0 unit 0 family inet address 172.16.7.115/32 primary
set interfaces lo0 unit 0 family iso address 49.0001.1720.1600.7115.00
set interfaces xe-0/2/0 unit 0 family iso
```

Config 19 underlay ISIS configuration at border-leaf6

```
set protocols isis interface xe-0/2/0.0
set protocols isis interface lo0.0
```

```
set protocols isis level 1 disable
set interfaces lo0 unit 0 family inet address 172.16.7.116/32 primary
set interfaces lo0 unit 0 family iso address 49.0001.1720.1600.7116.00
set interfaces xe-0/2/0 unit 0 family iso
```

For the MPLS connectivity we also need to explicitly enable the LDP and MPLS protocols on the xe-0/2/0 interface connected to the P1_transit router and to get the transport labels allocated.

Config 20 MPLS protocol and interface configurations at border-leaf5

```
set protocols ldp interface xe-0/2/0.0
set protocols ldp interface lo0.0
set protocols mpls interface xe-0/2/0.0
set interfaces xe-0/2/0 mtu 9216
set interfaces xe-0/2/0 unit 0 family inet address 192.168.14.2/24
set interfaces xe-0/2/0 unit 0 family mpls
```

Config 21 MPLS protocol and interface configurations at border-leaf6

```
set protocols ldp interface xe-0/2/0.0
set protocols ldp interface lo0.0
set protocols mpls interface xe-0/2/0.0
set interfaces xe-0/2/0 mtu 9216
set interfaces xe-0/2/0 unit 0 family inet address 192.168.15.2/24
set interfaces xe-0/2/0 unit 0 family mpls
```

We can also verify the basic state of the ISIS and MPLS using the following commands. This is recommended before starting any EVI and overlay BGP configurations.

```
root@border-leaf6_re> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
xe-0/2/0.0            P1_transit_re  2  Up                   26  2c:6b:f5:4c:cb:52
root@border-leaf6_re> show ldp database
Input label database, 172.16.7.116:0--172.16.7.20:0
Labels received: 5
  Label     Prefix
      3      172.16.7.20/32
 299936      172.16.7.115/32
 299920      172.16.7.116/32
 299888      172.16.7.217/32
 299952      172.16.7.218/32
Output label database, 172.16.7.116:0--172.16.7.20:0
Labels advertised: 5
  Label     Prefix
 299840      172.16.7.20/32
 300832      172.16.7.115/32
      3      172.16.7.116/32
 299952      172.16.7.217/32
 301088      172.16.7.218/32
root@border-leaf6_re>
root@border-leaf6_re> show ldp interface
```

```
Interface          Address                        Label space ID   Nbr   Next
                                                                   count  hello
lo0.0              172.16.7.116                   172.16.7.116:0    0      0
xe-0/2/0.0         192.168.15.2                   172.16.7.116:0    1      4
root@border-leaf6_re>
```

The MPLS label 299888 associated with border-leaf7 loopback IP @ 172.16.7.217 will be used as an outer/transport label when sending packets from DC site A to site B. You will see in the next section how the data packet is built when the EVPN EVI label and transport label are both used.

At the iBGP level, the local-as number 65000 will be used on all border-leaf nodes and they will peer with the core IP route-reflector. Here, the route reflector loopback IP address is 172.16.7.20.

Config 22 Border-leaf5 iBGP configuration for the connectivity with route-reflector P1_transit

```
set protocols bgp group WAN type internal
set protocols bgp group WAN local-address 172.16.7.115  >> local loopback IP@ enabled at lo0.0
advertised to ISIS as well
set protocols bgp group WAN family EVPN signaling  >> EVPN signalization to be used when peering to the
route-reflector
set protocols bgp group WAN local-as 65000  >> all border-leaf nodes share the same iBGP ASN in the core
network
set protocols bgp group WAN neighbor 172.16.7.20  >> loopback of the core IP route-reflector, reachable
via ISIS
set protocols bgp group WAN vpn-apply-export
```

Config 23 Border-leaf6 iBGP configuration for the connectivity with route-reflector P1_transit

```
set protocols bgp group WAN type internal
set protocols bgp group WAN local-address 172.16.7.116  >> local loopback IP@ enabled at lo0.0
advertised to ISIS as well
set protocols bgp group WAN family EVPN signaling
set protocols bgp group WAN local-as 65000  >> all border-leaf nodes share the same iBGP ASN
set protocols bgp group WAN neighbor 172.16.7.20  >> loopback of the core IP route-reflector
set protocols bgp group WAN vpn-apply-export  >> this is needed in case any export policy-statement was
needed in future
```

The route-reflector in the lab example is in-path but obviously if there's an existing off-path route-reflector in the core IP network, it can also be used and only the additional family EVPN signalization will have to be added at the RR level.

Config 24 Route-reflector P1_transit iBGP peering configuration

```
set protocols bgp group WAN type internal
set protocols bgp group WAN local-address 172.16.7.20
set protocols bgp group WAN family EVPN signaling  >> EVPN signalization for control-plane purposes
set protocols bgp group WAN cluster 172.16.7.20  >> setting a cluster id is needed at the route-
reflector node
set protocols bgp group WAN local-as 65000  >> overlay iBGP ASN number used also at all border-leaf
nodes
```

```
set protocols bgp group WAN neighbor 172.16.7.115  >> border-leaf5 in site A
set protocols bgp group WAN neighbor 172.16.7.116  >> border-leaf6 in site A
set protocols bgp group WAN neighbor 172.16.7.217  >> border-leaf7 in site B
set protocols bgp group WAN neighbor 172.16.7.218  >> border-leaf8 in site B
```

The route-reflector has the local cluster value enabled and peers to all border-leaf nodes, if the cluster-id is not set the route-reflection of EVPN routes between the data center sites won't work in the iBGP scenario.

The border-leaf in each site must also connect to the local fabric spines to learn the EVPN routes originated at the server leaf nodes. The following configuration is specific to integrate the border-leaf to the fabric LAN EVPN-VXLAN side.

Config 25 border-leaf6 fabric underlay/overlay eBGP peering configurations

```
set protocols bgp group underlay type external
set protocols bgp group underlay export my_underlay_export
set protocols bgp group underlay local-as 65115  >> border-leaf5/6 share the same underlay BGP ASN
set protocols bgp group underlay multipath multiple-as
set protocols bgp group underlay neighbor 192.168.9.1 peer-as 65101  >> peering underlay to spine5 in
site A
set protocols bgp group underlay neighbor 192.168.10.1 peer-as 65101  >> peering underlay to spine6 in
site A
set protocols bgp group overlay type external
set protocols bgp group overlay multihop
set protocols bgp group overlay local-address 172.16.7.116
set protocols bgp group overlay family EVPN signaling
set protocols bgp group overlay local-as 65115  >> border-leaf5/6 share the same overlay BGP ASN for
LAN EVPN-VXLAN
set protocols bgp group overlay multipath multiple-as
set protocols bgp group overlay bfd-liveness-detection minimum-interval 100
set protocols bgp group overlay bfd-liveness-detection multiplier 3
set protocols bgp group overlay neighbor 172.16.7.105 peer-as 65101  >> peering overlay to spine5 in
site A
set protocols bgp group overlay neighbor 172.16.7.106 peer-as 65101  >> peering overlay to spine6 in
site A
set protocols bgp group overlay vpn-apply-export
```

Config 26 border-leaf5 fabric underlay/overlay eBGP peering configurations

```
set protocols bgp group underlay type external
set protocols bgp group underlay export my_underlay_export
set protocols bgp group underlay local-as 65115  >> border-leaf5/6 share the same underlay BGP ASN
set protocols bgp group underlay multipath multiple-as
set protocols bgp group underlay neighbor 192.168.7.1 peer-as 65101
set protocols bgp group underlay neighbor 192.168.8.1 peer-as 65101
set protocols bgp group overlay type external
set protocols bgp group overlay multihop
set protocols bgp group overlay local-address 172.16.7.115
set protocols bgp group overlay family EVPN signaling
set protocols bgp group overlay local-as 65115  >> border-leaf5/6 share the same overlay BGP ASN for
LAN EVPN-VXLAN
set protocols bgp group overlay multipath multiple-as
set protocols bgp group overlay bfd-liveness-detection minimum-interval 100
```

```
set protocols bgp group overlay bfd-liveness-detection multiplier 3
set protocols bgp group overlay neighbor 172.16.7.105 peer-as 65101
set protocols bgp group overlay neighbor 172.16.7.106 peer-as 65101
set protocols bgp group overlay vpn-apply-export
```

From the verification point of view, we can use the well-known commands for BGP to check the peers for the fabric as well as for the DCI purposes are in the 'Established' state.

```
root@border-leaf6_re> show bgp summary
Threading mode: BGP I/O
Default eBGP mode: advertise - accept, receive - accept
Groups: 3 Peers: 5 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.evpn.0
                     118         80          0          0          0          0
inet.0
                       8          8          0          0          0          0
Peer                 AS     InPkt    OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/Received/
Accepted/Damped...
172.16.7.20         65000      2533      2093      0      4   15:21:17 Establ  >> overlay peering to
the route-reflector
  bgp.evpn.0: 61/80/61/0
  MACVRF101.evpn.0: 26/43/26/0
  __default_evpn__.evpn.0: 1/1/1/0
172.16.7.105        65101      2021      2421      0      4   15:21:13 Establ  >> peering to the fabric
spine1 in site A
  bgp.evpn.0: 17/19/19/0
  MACVRF101.evpn.0: 16/17/17/0
  __default_evpn__.evpn.0: 0/0/0/0
172.16.7.106        65101      2021      2422      0      4   15:21:13 Establ  >> peering to the fabric
spine2 in site A
  bgp.evpn.0: 2/19/19/0
  MACVRF101.evpn.0: 1/17/17/0
  __default_evpn__.evpn.0: 0/0/0/0
192.168.9.1         65101      2004      2027      0      4   15:21:17 Establ  >> underlay peering to
the fabric spine1 in site A
  inet.0: 4/4/4/0
192.168.10.1        65101      2004      2027      0      4   15:21:17 Establ  >> underlay peering to
the fabric spine2 in site A
  inet.0: 4/4/4/0
root@border-leaf6_re>
```

In these verification tasks, we see that both border-leaf nodes are sharing the same BGP ASN number for the EVPN-VXLAN peering towards the spines, while still maintaining the eBGP as the type of peering towards spines5/6. This helps to prevent any sub-optimal routes to be installed in the RIB – which could happen of both border-leaf nodes were using different overlay ASN.

EVPN Instance (EVI) Provisioning When Stitching to MPLS

The EVI (aka MAC-VRF) configuration is essential when enabling the new EVPN L2 service for the customer. For the VXLAN to MPLS stitching scenario, the concept is similar to a VXLAN-to-VXLAN scenario when it comes to the control of the stretched VLANs. The operator gets the right toolset to define which workloads are stretched and

which are staying local. This is important as in many cases, there's a customer requirement to keep the data only local – for example when the requirement is not to stretch it between the two countries. In this case the admin will precisely define only the VLANs for interconnect purposes that have high DRS (disaster recovery solution) requirements and are not restricted to site-to-site data storage replication.

Before we go into more details on the configurations and verification, we will quickly refresh the simplified topology used for our EVPN-VXLAN to EVPN-MPLS lab in Figure 6.2.

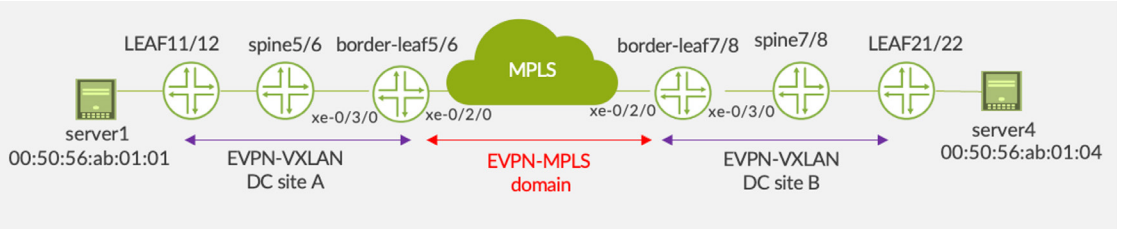


Figure 6.2 EVPN-VXLAN to EVPN-MPLS Stitching - Simplified Lab Topology View

Here's an example of the EVI (EVPN instance) configuration used at the border-leaf5 and border-leaf6 from data center site A. Comparing this to the previous use case you can see that within the interconnect block of configurations the encapsulation format is now MPLS and the instance type is virtual-switch for the MX border-leaf.

Config 27 border-leaf5 EVI config when stitching VXLAN to MPLS in DC site A

```
set routing-instances MACVRF101 instance-type virtual-switch  >> will enable the VLAN-aware service-
type on MX
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN  >> encapsulation used in LAN fabric
to server-leaf
set routing-instances MACVRF101 protocols evpn extended-vni-list all
set routing-instances MACVRF101 protocols evpn interconnect vrf-target target:1:101  >> route-target
for DCI EVPN-MPLS
set routing-instances MACVRF101 protocols evpn interconnect route-distinguisher 172.16.7.115:101  >>
iRD unique per node
set routing-instances MACVRF101 protocols evpn interconnect esi 00:00:11:11:11:11:11:11:11:11  >>
site-id iESI in DC A
set routing-instances MACVRF101 protocols evpn interconnect esi all-active
set routing-instances MACVRF101 protocols evpn interconnect interconnected-VLAN-list 1001
set routing-instances MACVRF101 protocols evpn interconnect encapsulation mpls
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 bridge-domains bd1001 domain-type bridge
set routing-instances MACVRF101 bridge-domains bd1001 vlan-id 1001
set routing-instances MACVRF101 bridge-domains bd1001 routing-interface irb.1001
set routing-instances MACVRF101 bridge-domains bd1001 vxlan vni 51001  >> vxlan vni value we stitch to
MPLS
set routing-instances MACVRF101 bridge-domains bd1001 VXLAN ingress-node-replication
set routing-instances MACVRF101 route-distinguisher 172.16.7.115:1  >> unique LAN EVPN-VXLAN RD per
node
set routing-instances MACVRF101 vrf-target target:1:8888  >> route-target for LAN EVPN-VXLAN in fabric
site A
```

A similar configuration will be enabled at the boder-leaf6 from site A where the iESI will be the same to allow for site identification but also load-balancing to happen whenever the site B hosts are trying to reach the site A hosts. The EVPN-MPLS interconnect route-target for DCI purposes, target:1:101, is the same on all border-leaf nodes enabled for DCI, as well as the local LAN EVPN-VXLAN target:1:8888 is shared between the border-leaf nodes and the server-leaf nodes.

Config 28 border-leaf6 EVI config when stitching VXLAN to MPLS in DC site A

```
set routing-instances MACVRF101 instance-type virtual-switch >> will enable the VLAN-aware service-type
on MX
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN >> encapsulation used in LAN fabric to
server-leaf
set routing-instances MACVRF101 protocols evpn extended-vni-list all
set routing-instances MACVRF101 protocols evpn interconnect vrf-target target:1:101 >> route-target for
DCI EVPN-MPLS
set routing-instances MACVRF101 protocols evpn interconnect route-distinguisher 172.16.7.116:101 >> iRD
unique per node
set routing-instances MACVRF101 protocols evpn interconnect esi 00:00:11:11:11:11:11:11:11:11 >> site-id
iESI in DC A
set routing-instances MACVRF101 protocols evpn interconnect esi all-active >> only all-active is
currently supported
set routing-instances MACVRF101 protocols evpn interconnect interconnected-VLAN-list 1001 >> explicit
list of stretched VLANs
set routing-instances MACVRF101 protocols evpn interconnect encapsulation mpls >> change of
encapsulation to MPLS
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 bridge-domains bd1001 domain-type bridge
set routing-instances MACVRF101 bridge-domains bd1001 vlan-id 1001
set routing-instances MACVRF101 bridge-domains bd1001 vxlan vni 51001 >> vxlan vni value we stitch to
MPLS
set routing-instances MACVRF101 bridge-domains bd1001 VXLAN ingress-node-replication
set routing-instances MACVRF101 route-distinguisher 172.16.7.116:1 >> unique LAN EVPN-VXLAN RD per node
set routing-instances MACVRF101 vrf-target target:1:8888 >> route-target for LAN EVPN-VXLAN in fabric
site A
```

When compared to the previous example of stitching, here we specify in the interconnect section the list of VLANs we want to stretch using the interconnect-VLAN-list. This is because in case of stitching from VXLAN to MPLS we don't use the notion of VNI anymore.

The data center site B configurations are consistent and share the same interconnect route-target, target:1:101, however for the LAN EVPN-VXLAN purposes we allocate a different route-target than the one from the site A. Here's an example configuration of the DC site B EVI for border-leaf nodes.

Config 29 border-leaf7 EVI config when stitching VXLAN to MPLS in DC site B

```
set routing-instances MACVRF101 instance-type virtual-switch >> will enable the VLAN-aware service-type
on MX
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN >> encapsulation used in LAN fabric to
server-leaf
set routing-instances MACVRF101 protocols evpn extended-vni-list all
```

```
set routing-instances MACVRF101 protocols evpn interconnect vrf-target target:1:101  >> route-target
for DCI EVPN-MPLS
set routing-instances MACVRF101 protocols evpn interconnect route-distinguisher 172.16.7.217:101  >>
iRD EVPN-MPLS
set routing-instances MACVRF101 protocols evpn interconnect esi 00:00:22:22:22:22:22:22:22:22  >>
site-id iESI in DC B
set routing-instances MACVRF101 protocols evpn interconnect esi all-active
set routing-instances MACVRF101 protocols evpn interconnect interconnected-VLAN-list 1001  >> explicit
list of stretched VLANs
set routing-instances MACVRF101 protocols evpn interconnect encapsulation mpls  >> change of
encapsulation to MPLS
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 bridge-domains bd1001 domain-type bridge
set routing-instances MACVRF101 bridge-domains bd1001 vlan-id 1001
set routing-instances MACVRF101 bridge-domains bd1001 vxlan vni 51001  >> vxlan vni value we stitch to
MPLS
set routing-instances MACVRF101 bridge-domains bd1001 VXLAN ingress-node-replication
set routing-instances MACVRF101 route-distinguisher 172.16.7.217:1  >> LAN EVPN-VXLAN route-
distinguisher
set routing-instances MACVRF101 vrf-target target:1:9999  >> route-target for LAN EVPN-VXLAN in fabric
site B
```

DC site B border-leaf8 is also sharing a new iESI (interconnect ESI) 00:00:22:22:22:22:22:22:22:22 with the local site border-leaf7, however it will join the site A border-leaf5/6 by using the same interconnect route target (iRT) target:1:101. As indicated, the local EVPN-VXLAN for LAN purposes will be using for the given EVI instance a new route-target target:1:9999. This value will be also used at the server leaf nodes – LEAF21/LEAF22.

The route distinguisher for LAN and for WAN is also allocated in the given EVI configuration and is unique per each node.

Config 30 border-leaf8 EVI config when stitching VXLAN to MPLS in DC site B

```
set routing-instances MACVRF101 instance-type virtual-switch  >> will enable the VLAN-aware service-
type on MX
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN  >> encapsulation used in LAN
fabric to server-leaf
set routing-instances MACVRF101 protocols evpn extended-vni-list all
set routing-instances MACVRF101 protocols evpn interconnect vrf-target target:1:101  >> route-target
for DCI EVPN-MPLS
set routing-instances MACVRF101 protocols evpn interconnect route-distinguisher 172.16.7.218:101  >>
iRD EVPN-MPLS
set routing-instances MACVRF101 protocols evpn interconnect esi 00:00:22:22:22:22:22:22:22:22  >>
site-id iESI in DC B
set routing-instances MACVRF101 protocols evpn interconnect esi all-active
set routing-instances MACVRF101 protocols evpn interconnect interconnected-VLAN-list 1001  >> explicit
list of stretched VLANs
set routing-instances MACVRF101 protocols evpn interconnect encapsulation mpls  >> change of
encapsulation to MPLS
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 bridge-domains bd1001 domain-type bridge
set routing-instances MACVRF101 bridge-domains bd1001 vlan-id 1001  >> vlan-id used for local pe-ce
ports
set routing-instances MACVRF101 bridge-domains bd1001 vxlan vni 51001  >> vxlan vni value we stitch to
```

```
MPLS
set routing-instances MACVRF101 bridge-domains bd1001 VXLAN ingress-node-replication
set routing-instances MACVRF101 route-distinguisher 172.16.7.218:1  >> LAN EVPN-VXLAN route-
distinguisher
set routing-instances MACVRF101 vrf-target target:1:9999  >> route-target for LAN EVPN-VXLAN in fabric
site B
```

DC site A and DC site B configurations are very similar, so automating them using Ansible or Python shouldn't be a big problem. Mainly the interconnect iESI and local LAN route-targets are changing, besides route-distinguishers which are unique per node in site A and B – on each node one RD for LAN fabric part and one for WAN DCI.

Border-leaf config for DCI purposes is the focus of this *Day One* book, however it's also important to bring up what the server-leaf EVPN-VXLAN configuration in each data center site looks like. This will help you understand later how the RT and ESI values are changing for the MAC addresses that are advertised through EVPN between the two data center sites.

Here, for example, is the server-leaf LEAF11 and LEAF12 configuration of the EVI (MAC-VRF) when connecting server1.

Config 31 LEAF11 server-leaf config in DC site A

```
set routing-instances MACVRF101 instance-type mac-vrf  >> instance type MACVRF is used on qfx series of
switches
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001  >> we define which VNIs are
enabled explicitly
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 service-type vlan-aware  >> explicit definition of the service-type
set routing-instances MACVRF101 interface ae0.0  >> we enable the interface connected to server1 in the
mac-VRF
set routing-instances MACVRF101 route-distinguisher 172.16.7.11:1  >> unique LAN level Route-
Distinguisher (RD)
set routing-instances MACVRF101 vrf-target target:1:8888  >> same value used on all server-leaf nodes
sharing that mac-VRF
set routing-instances MACVRF101 vlans vlan1001 vlan-id 1001
set routing-instances MACVRF101 vlans vlan1001 l3-interface irb.1001
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 51001
set interfaces ae0 description "EP-style to server1"
set interfaces ae0 mtu 9100
set interfaces ae0 esi 00:00:88:88:88:88:88:88:11:12  >> local LAN EVPN-VXLAN ESI used for server1
multihoming
set interfaces ae0 esi all-active  >> QFX server leaf only support all-active ESI but we enable it
explicitly as well
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:01:88:88:01:01  >> same value as on
LEAF12
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members 1001
set interfaces xe-0/0/0 description member-link_ae0
set interfaces xe-0/0/0 ether-options 802.3ad ae0  >> physical interface connected to server1 and
```

```
member of AE0 LAG
set chassis aggregated-devices ethernet device-count 2  >> defines the number of AE interfaces to
servers, it can be higher
set forwarding-options evpn-vxlan shared-tunnels
# The following configs are also required when the same qfx5120 server-leaf node runs the T5 IP VRF
instances
set routing-options forwarding-table chained-composite-next-hop ingress EVPN
set forwarding-options vxlan-routing next-hop 32768
set forwarding-options vxlan-routing interface-num 8192
set forwarding-options vxlan-routing overlay-ecmp
```

You can see that at the QFX server-leaf node the instance-type is mac-VRF while on the MX we're still using the traditional virtual-switch instance type. This is because at the time of writing this book we used the Junos release which qualified only the traditional virtual-switch instance type and because of the official support of virtual-switch for any MPLS purposes when running release 21.4r2 of Junos at the border-leaf MX devices.

For the completeness of our solution, we also introduce the MAC-VRF config used at the server-leaf LEAF12 – both in data center site A, as well as the LEAF21 from site B.

Config 32 LEAF12 server-leaf config in DC site A

```
set routing-instances MACVRF101 instance-type mac-vrf  >> this instance type is used on server-leaf
nodes such as QFX
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 service-type vlan-aware  >> explicitly set on QFX but border-leaf MX
use virtual-switch
set routing-instances MACVRF101 interface ae0.0
set routing-instances MACVRF101 route-distinguisher 172.16.7.12:1
set routing-instances MACVRF101 vrf-target target:1:8888  >> this value was also used at the border-
leaf5/6 for LAN section
set routing-instances MACVRF101 vlans vlan1001 vlan-id 1001
set routing-instances MACVRF101 vlans vlan1001 l3-interface irb.1001
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 51001
set interfaces ae0 description "EP-style to server1"
set interfaces ae0 mtu 9100
set interfaces ae0 esi 00:00:88:88:88:88:88:88:11:12  >> local LAN EVPN-VXLAN ESI used for server1
multihoming
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:01:88:88:01:01  >> same value as on
LEAF11
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members 1001
set interfaces xe-0/0/0 description member-link_ae0
set interfaces xe-0/0/0 ether-options 802.3ad ae0  >> physical interface connected to server1 and
member of AE0 LAG
set chassis aggregated-devices ethernet device-count 2  >> defines the number of AE interfaces to
servers, it can be higher
set forwarding-options evpn-vxlan shared-tunnels  >> this global config is specific to qfx5120 to
enable optimal MAC-VRF
```

```
# The following configs are also required when the same qfx5120 server-leaf node runs the T5 IP VRF
instances
set routing-options forwarding-table chained-composite-next-hop ingress EVPN
set forwarding-options vxlan-routing next-hop 32768  >> for qfx5120 specific to enable more overlay
next-hops
set forwarding-options vxlan-routing interface-num 8192  >> for qfx5120 leaf when dealing with IPv4 &
IPv6 IRBs
set forwarding-options vxlan-routing overlay-ecmp  >> This is only required when in parallel the Type-5
instance is used
```

The following configuration of the server-leaf in site B – leaf21 can also be highlighted, mainly to mention again that it should not use the same route-target as the route-targets in site A.

Config 33 server-leaf LEAF21 configuration of mac-VRF in site B

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-instances MACVRF101 protocols evpn encapsulation VXLAN
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 service-type vlan-aware  >> this service will work with MX instance type
virtual-switch
set routing-instances MACVRF101 interface ae0.0
set routing-instances MACVRF101 route-distinguisher 172.16.7.206:1
set routing-instances MACVRF101 vrf-target target:1:9999  >> different value of EVPN-VXLAN LAN RT in
site B
set routing-instances MACVRF101 vlans vlan1001 vlan-id 1001
set routing-instances MACVRF101 vlans vlan1001 l3-interface irb.1001
set routing-instances MACVRF101 vlans vlan1001 vxlan vni 51001
set forwarding-options evpn-vxlan shared-tunnels  >> this is only required at the qfx5120
```

Server4 interface connect configuration of the ESI at interface AE0, used locally within the DC site B EVPN-VXLAN fabric and explicitly enabled.

Config 34 ESI-LAG config for server4 connection in DC site B

```
set interfaces ae0 description ep-style
set interfaces ae0 mtu 9100
set interfaces ae0 esi 00:00:88:88:88:88:88:88:21:22  >> this ESI local to site B will be translated to
iESI at border-leaf7/8
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:01:88:88:01:01  >> value shared between
LEAF21/LEAF22
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae0 unit 0 family ethernet-switching vlan members 1001
set interfaces xe-0/0/0 description “ep-style to server4”
set interfaces xe-0/0/0 ether-options 802.3ad
set chassis aggregated-devices ethernet device-count 2  >> this configuration may be already in place,
can be higher
```

Once the EVI configs shown in Figurer 6.3 were enabled at border-leaf nodes in each DC site as well as at the server leaf nodes, the following prefix advertisement is received at the border-leaf5 from leaf11 in site A within the EVPN-VXLAN fabric domain. The same MAC@ will then be re-originated for seamless DCI stitching purposes but this time with MPLS encapsulation information. The label value instead of the VNI value will be used and the community attribute will be informing about the encapsulation type as part of the BGP EVPN advertisement.

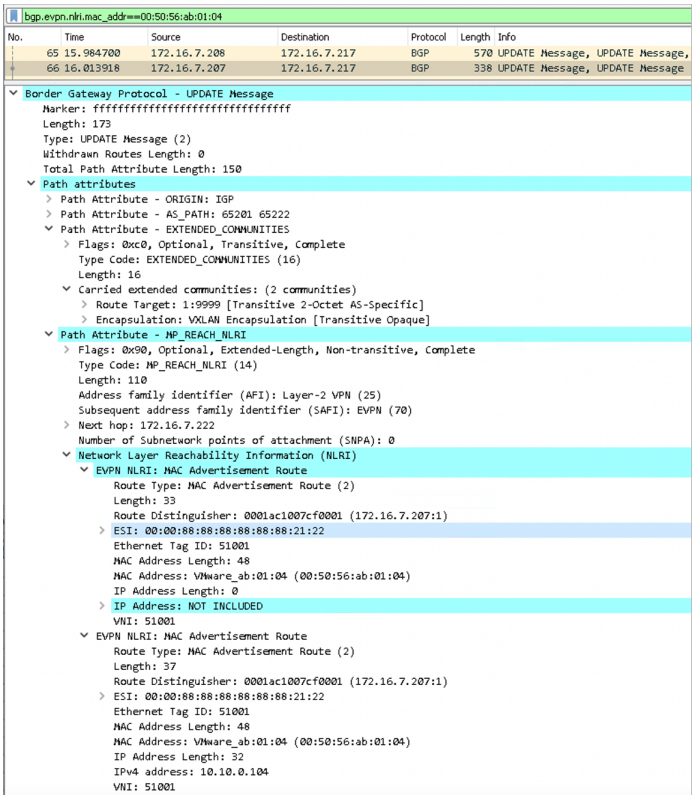


Figure 6.3 Server4 MAC EVPN Route Received on the Border-leaf7 From leaf22 in DC Site B

The border-leaf7 verification of the local fabric route received from LEAF22 in the data center site B is also confirming it's getting it from the spine7/8 route servers and next-hop IP corresponding to LEAF21/LEAF22 – here the next hop for LEAF22 is 172.16.7.222 and the neighbor spine7 IP@ is 172.16.7.207:

```
root@border-leaf7_re# run show route receive-protocol bgp 172.16.7.207 table MACVRF101.
evpn.0 active-path next-hop 172.16.7.222
MACVRF101.evpn.0: 45 destinations, 59 routes (45 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  1:172.16.7.207:1::8888888888882122::0/192 AD/EVI
```

```
*                       172.16.7.222                            65201 65222 I
  1:172.16.7.222:0::888888888888122::FFFF:FFFF/192 AD/ESI
*                       172.16.7.222                            65201 65222 I
  2:172.16.7.207:1::51001::00:00:01:01:01:01/304 MAC/IP
*                       172.16.7.222                            65201 65222 I
  2:172.16.7.207:1::51001::00:50:56:ab:01:04/304 MAC/IP
*                       172.16.7.222                            65201 65222 I
  2:172.16.7.207:1::51001::00:00:01:01:01:01::10.10.0.1/304 MAC/IP
*                       172.16.7.222                            65201 65222 I
  2:172.16.7.207:1::51001::00:50:56:ab:01:04::10.10.0.104/304 MAC/IP
*                       172.16.7.222                            65201 65222 I
  3:172.16.7.207:1::51001::172.16.7.222/248 IM
*                       172.16.7.222                            65201 65222 I
[edit]
root@border-leaf7_re#
```

To verify the detailed information for one of the EVPN prefix local to the site B before it gets re-originated to the MPLS domain we run the following verification command:

```
root@border-leaf7_re# run show route table MACVRF101.evpn.0 match-prefix 2:172.16.7.207:1::51001::00:
50:56:ab:01:04/304 detail active-path
MACVRF101.evpn.0: 45 destinations, 59 routes (45 active, 0 holddown, 0 hidden)
2:172.16.7.207:1::51001::00:50:56:ab:01:04/304 MAC/IP (2 entries, 1 announced)
      *BGP   Preference: 170/-101
             Route Distinguisher: 172.16.7.207:1
             Next hop type: Indirect, Next hop index: 0
             Address: 0x76b6160
             Next-hop reference count: 30
             Source: 172.16.7.207
             Protocol next hop: 172.16.7.222
             Indirect next hop: 0x2 no-forward INH Session ID: 0
             State: <Secondary Active Ext>
             Peer AS: 65201
             Age: 17:36     Metric2: 0
             Validation State: unverified
             Task: BGP_65201_65217.172.16.7.207
             Announcement bits (1): 0-MACVRF101-EVPN
             AS path: 65201 65222 I
             Communities: target:1:9999 encapsulation:VXLAN(0x8)
             Import Accepted
             Route Label: 51001
             ESI: 00:00:88:88:88:88:88:88:21:22
             Localpref: 100
             Router ID: 172.16.7.207
             Primary Routing Table: bgp.evpn.0
             Thread: junos-main
[edit]
root@border-leaf7_re#
```

Indeed, you can observe at border-leaf7 in DC site B that the server4 MAC@ when originated at LEAF22 is still using the VXLAN community and original ESI value of 00:00:88:88:88:88:88:88:21:22.

Once the server4 MAC EVPN route is received, it will be re-originated from the same EVI to the DCI domain but with new BGP EVPN attribute values. The new

interconnect ESI will be set at the border-leaf7 level and because we changed the encapsulation format from VXLAN to MPLS, at the control plane level in the given MAC@ of server4 (00:50:56:ab:01:04) will have now the Route-Label associated – here the route label 299808 is used. This information will then be used by the border-leaf5/6 in site A to build the MPLS data packet with label 299808 used as VPN label. The VNI related information is not part of the advertisement from the boder-leaf7/8 in this DCI use case anymore because the VXLAN to MPLS stitching is taking place. The vlan-tag information is corresponding also to the VLAN we configured at the interconnect EVPN level – here we used 1001 value which was the same as the local fabric VNI value, however if there were values above 4092 for local fabric VNIs then we would need to allocate in the configuration some value from the range of 4K VLANs when stitching to the MPLS domain.

```
root@border-leaf7_re# run show route advertising-protocol bgp 172.16.7.20 table MACVRF101.evpn.0
evpn-mac-address 00:50:56:ab:01:04 active-path
MACVRF101.evpn.0: 45 destinations, 59 routes (45 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  2:172.16.7.217:101::1001::00:50:56:ab:01:04/304 MAC/IP
*                         Self                            100        I
  2:172.16.7.217:101::1001::00:50:56:ab:01:04::10.10.0.104/304 MAC/IP
*                         Self                            100        I
[edit]
root@border-leaf7_re#
root@border-leaf7_re# run show route advertising-protocol bgp 172.16.7.20 table MACVRF101.evpn.0
evpn-mac-address 00:50:56:ab:01:04 active-path detail
MACVRF101.evpn.0: 45 destinations, 59 routes (45 active, 0 holddown, 0 hidden)
* 2:172.16.7.217:101::1001::00:50:56:ab:01:04/304 MAC/IP (1 entry, 1 announced)
 BGP group WAN type Internal
    Route Distinguisher: 172.16.7.217:101
    Route Label: 299808
    ESI: 00:00:22:22:22:22:22:22:22:22
    Nexthop: Self
    Flags: Nexthop Change
    Localpref: 100
    AS path: [65000] I
    Communities: target:1:101
* 2:172.16.7.217:101::1001::00:50:56:ab:01:04::10.10.0.104/304 MAC/IP (1 entry, 1 announced)
 BGP group WAN type Internal
    Route Distinguisher: 172.16.7.217:101
    Route Label: 299808
    ESI: 00:00:22:22:22:22:22:22:22:22
    Nexthop: Self
    Flags: Nexthop Change
    Localpref: 100
    AS path: [65000] I
    Communities: target:1:101
[edit]
root@border-leaf7_re#
```

To follow the control plane part of the verification, the border-leaf6 in site A once it received the server4 MAC@ info from the remote site B, will process it and readvertise to the local fabric A with the new EVPN BGP attributes – specific to his local site A

configurations of the border-leaf5/6. It will advertise it to the spine5/6 route servers using the local interconnect iESI value of 00:00:11:11:11:11:11:11:11:11 and the route label information now corresponds to the vxlan vni 51001 instead of the route label value 299808 used when originally received from the MPLS DCI domain route-reflector.

```
root@border-leaf6_re> show route advertising-protocol bgp 172.16.7.105 table MACVRF101.evpn.0 active-
path evpn-mac-address 00:50:56:ab:01:04
MACVRF101.evpn.0: 48 destinations, 83 routes (48 active, 0 holddown, 17 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  2:172.16.7.116:1::51001::00:50:56:ab:01:04/304 MAC/IP
*                         Self                                    I
  2:172.16.7.116:1::51001::00:50:56:ab:01:04::10.10.0.104/304 MAC/IP
*                         Self                                    I
root@border-leaf6_re>
root@border-leaf6_re> show route advertising-protocol bgp 172.16.7.105 table MACVRF101.
evpn.0 active-path evpn-mac-address 00:50:56:ab:01:04 detail
MACVRF101.evpn.0: 48 destinations, 83 routes (48 active, 0 holddown, 17 hidden)
* 2:172.16.7.116:1::51001::00:50:56:ab:01:04/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
     Route Distinguisher: 172.16.7.116:1
     Route Label: 51001
     ESI: 00:00:11:11:11:11:11:11:11:11
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [65115] I
     Communities: target:1:8888 encapsulation:VXLAN(0x8)
* 2:172.16.7.116:1::51001::00:50:56:ab:01:04::10.10.0.104/304 MAC/IP (1 entry, 1 announced)
 BGP group overlay type External
     Route Distinguisher: 172.16.7.116:1
     Route Label: 51001
     ESI: 00:00:11:11:11:11:11:11:11:11
     Nexthop: Self
     Flags: Nexthop Change
     AS path: [65115] I
     Communities: target:1:8888 encapsulation:VXLAN(0x8)
root@border-leaf6_re>
```

We can also observe that the DCI community target:1:101 used by all border-leaf nodes for DCI purposes is replaced with the site local route-target target:1:8888. In fact, each site should use per EVI/MAC-VRF a site-specific local route-target – here the site A uses target:1:8888 and site B is using target:1:9999.

At the data plane level, when server1 pings server4, the packet reaches border-leaf6 interfaces xe-0/3/0 and xe-0/3/ before stitching to MPLS, the data packet looks like Figure 6.4.

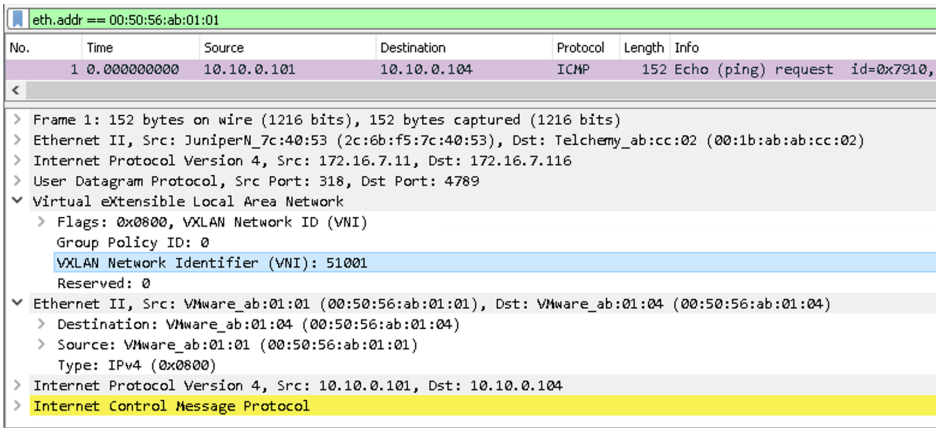


Figure 6.4 *Server1 Sourced Data Frame Destined To Server4, When Received at the Border-leaf6 Before Stitching VXLAN To MPLS*

You can see that the VXLAN encapsulation is used and the local site VNI value of 51001 is used also at the forwarding level. It can be also noticed that by default the original vlan-id is stripped and only the VNI value in the VXLAN header is used when the packet is sent from server LEAF11/LEAF12 to border-leaf5/6. See Figure 6.5.

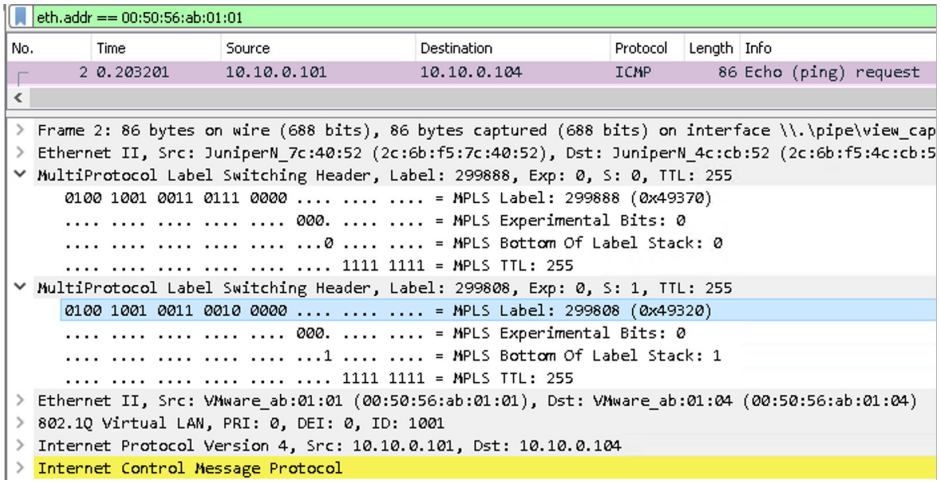


Figure 6.5 *Server1 Sourced Data Frame Destined to Server4, When Received at the Border-leaf6 After Stitching VXLAN To MPLS*

After the VXLAN to MPLS stitching, the border-leaf5/6 uses the inner (EVI) label 299808 to reach the server4 mac@ via the MPLS label mapped to the remote site B interconnect iESI, it also adds the transport MPLS label 299888.

In DC site A, to analyze the data plane for specific MAC@ of server4 originated in DC site B, use the following Junos commands:

```
root@border-leaf6_re> show route table MACVRF101.evpn.0 evpn-mac-address 00:50:56:ab:01:04 active-
path next-hop 172.16.7.217
MACVRF101.evpn.0: 39 destinations, 57 routes (39 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
2:172.16.7.217:101::1001::00:50:56:ab:01:04/304 MAC/IP
                *[BGP/170] 2d 06:03:06, localpref 100, from 172.16.7.20
                   AS path: I, validation-state: unverified
                 >  to 192.168.15.1 via xe-0/2/0.0, Push 299888
2:172.16.7.217:101::1001::00:50:56:ab:01:04::10.10.0.104/304 MAC/IP
                *[BGP/170] 2d 06:03:06, localpref 100, from 172.16.7.20
                   AS path: I, validation-state: unverified
                 >  to 192.168.15.1 via xe-0/2/0.0, Push 299888
root@border-leaf6_re> show route label 299888
mpls.0: 15 destinations, 16 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
299888          *[EVPN/7] 2d 06:03:11, remote-pe 172.16.7.217, routing-instance MACVRF101, route-type
Egress-MAC
                >  to 192.168.15.1 via xe-0/2/0.0, Push 299808, Push 299888(top)
root@border-leaf6_re>
```

You can see the MPLS label allocation will happen for inner 299808 (related to destination EVI/iESI) and outer label 299888 – top label for transport purposes.

Because the destination server4 MAC@ from site B is reachable via two border-leaf nodes, border-leaf7/8, the EVPN aliasing feature (load balancing) per destination ESI at the per EVI level will be used:

```
root@border-leaf6_re> show route table mpls.0 protocol EVPN | grep "Egress-MAC"
301056          *[EVPN/7] 00:00:49, routing-instance MACVRF101, route-type Egress-MAC, ESI
00:00:22:22:22:22:22:22:22:22
301072          *[EVPN/7] 02:43:21, remote-pe 172.16.7.217, routing-instance MACVRF101, route-type
Egress-MAC
301120          *[EVPN/7] 00:00:49, remote-pe 172.16.7.218, routing-instance MACVRF101, route-type
Egress-MAC
root@border-leaf6_re> show route label 301056
mpls.0: 16 destinations, 17 routes (16 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
301056          *[EVPN/7] 00:00:51, routing-instance MACVRF101, route-type Egress-MAC, ESI
00:00:22:22:22:22:22:22:22:22
                   to 192.168.15.1 via xe-0/2/0.0, Push 299808, Push 299888(top)
                >  to 192.168.15.1 via xe-0/2/0.0, Push 299808, Push 299952(top)
root@border-leaf6_re>
```

When it comes to bridge the MAC table verification at the MX border-leaf level you can also check which are the ESIs associated with the server mac addresses:

```
root@border-leaf6_re> show bridge mac-table instance MACVRF101
MAC flags      (S -static MAC, D -dynamic MAC, L -locally learned, C -Control MAC
   O -OVSDB MAC, SE -Statistics enabled, NM -Non configured MAC, R -Remote PE MAC, P -Pinned MAC, FU
- Fast Update)
Routing instance : MACVRF101
 Bridging domain : bd1001, VLAN : 1001
   MAC                 MAC      Logical          NH     MAC         active
```

```
   address             flags    interface        Index  property    source
   00:00:01:01:01:01   DRP      vtep.32769                          172.16.7.11
   00:50:56:ab:01:01   DR       esi.624                             00:00:88:88:88:88:88:11:12
   00:50:56:ab:01:03   DR       vtep.32770                          172.16.7.13
   00:50:56:ab:01:04   DC       .local..1048589  1048589            00:00:22:22:22:22:22:22:22:22
root@border-leaf6_re>
```

To further trace the history of MAC@ within the EVPN database you can specify the origin of the server4 mac address you want to review:

```
root@border-leaf6_re> show evpn database origin dci-remote
Instance: MACVRF101
VLAN  DomainId  MAC address        Active source                  Timestamp        IP address
1001            00:00:01:01:01:01  00:00:22:22:22:22:22:22:22:22  Jun 27 05:56:31
1001            00:50:56:ab:01:04  00:00:22:22:22:22:22:22:22:22  Jun 27 05:56:31  10.10.0.104
root@border-leaf6_re> show evpn database origin dci-remote mac-address 00:50:56:ab:01:04 extensive
Instance: MACVRF101
VLAN ID: 1001, MAC address: 00:50:56:ab:01:04
  Nexthop ID: 1048589
  State: 0x0
  Source: 00:00:22:22:22:22:22:22:22:22, Rank: 1, Status: Active
    Remote origin: 172.16.7.217
    Remote state: <Mac-Only-Adv Interconnect-DC>
    Remote origin: 172.16.7.218
    Remote state: <Mac-Only-Adv Interconnect-DC>
    Mobility sequence number: 0 (minimum origin address 172.16.7.217)
    Timestamp: Jun 27 05:56:31.636360 (0x62b9a8ff)
    State: <Remote-To-Local-Adv-Done>
    MAC advertisement route status: Not created (no local state present)
    Interconn advertisement route status: DC route created
    IP address: 10.10.0.104
    Flags: <Sent-to-l2ald Interconnect-DC>
      Remote origin: 172.16.7.217
      Remote state: <Interconnect-DC>
      Remote origin: 172.16.7.218
      Remote state: <Interconnect-DC>
      Interconn advertisement route status: DC route created
    History db: <No entries>
root@border-leaf6_re>
root@border-leaf6_re>
```

The DC route-created status is saying that even if we received the MAC@ from Interconnect (remote state), the local DC route was also created and will be advertised to the local site spines and server leaf devices.

In the EVPN-VXLAN-to-EVPN-MPLS stitching ethernet bridging scenario, similarly to what was discussed in the EVPN-VXLAN-to-EVPN-VXLAN stitching, the flooding mesh-groups will be installed at all the border-leaf nodes for the given EVI and bridge-domain. This is something you can also verify at the MX level using the following command:

```
root@border-leaf6_re> show bridge flood instance MACVRF101 detail
Name: MACVRF101
CEs: 0
VEs: 6
Bridging domain: bd1001
```

```
Flood Routes:
  Prefix    Type          Owner                NhType          NhIndex
  0x30004/51 FLOOD_GRP_COMP_NH __ves__          ulst            1048580
  0x30006/51 FLOOD_GRP_COMP_NH __wan_flood__    ulst            1048579
  0x30003/51 FLOOD_GRP_COMP_NH __re_flood__     ulst            1048584
root@border-leaf6_re>
```

For the flood groups in the case of the EVPN-VXLAN-to-EVPN-MPLS stitching will have part of the member logical interfaces on the VXLAN domain side and part on the MPLS domain side:

```
root@border-leaf6_re> show bridge flood instance MACVRF101 extensive
Name: MACVRF101
CEs: 0
VEs: 6
Bridging domain: bd1001
  Flood route prefix: 0x30004/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __ves__
  Flood group name: __ves__
  Flood group index: 0
  Nexthop type: ulst
  Nexthop index: 1048580
    Flooding to:
    Name             Type          NhType          Index
    __wan_flood__    Group          comp            609
        Composition: flood-to-all
    Component flood-nh(s) (for flooding to EVPN core):
    Index         Peer             NH-Type
    608           172.16.7.217     comp (IM/SH)
    637           172.16.7.218     comp (IM/SH)
  Flood route prefix: 0x30006/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __wan_flood__
  Flood group name: __wan_flood__
  Flood group index: 15
  Nexthop type: ulst
  Nexthop index: 1048579
    Flooding to:
    Name             Type          NhType          Index
    __ves__          Group          comp            625
        Composition: flood-to-all
        Flooding to:
        Name             Type          NhType          Index          RVTEP-IP
        vtep.32768       CORE_FACING    venh            621           172.16.7.12
        vtep.32769       CORE_FACING    venh            622           172.16.7.11
        vtep.32770       CORE_FACING    venh            623           172.16.7.13
        vtep.32771       CORE_FACING    venh            640           172.16.7.115
  Flood route prefix: 0x30003/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __re_flood__
  Flood group name: __re_flood__
  Flood group index: 65534
  Nexthop type: ulst
  Nexthop index: 1048584
    Flooding to:
    Name             Type          NhType          Index
```

```
    __ves__          Group          comp            625
        Composition: flood-to-all
        Flooding to:
        Name             Type          NhType          Index          RVTEP-IP
        vtep.32768       CORE_FACING    venh            621           172.16.7.12
        vtep.32769       CORE_FACING    venh            622           172.16.7.11
        vtep.32770       CORE_FACING    venh            623           172.16.7.13
        vtep.32771       CORE_FACING    venh            640           172.16.7.115
    Flooding to:
    Name             Type          NhType          Index
    __wan_flood__    Group          comp            609
        Composition: flood-to-all
    Component flood-nh(s) (for flooding to EVPN core):
    Index         Peer             NH-Type
    608           172.16.7.217     comp (IM/SH)
    637           172.16.7.218     comp (IM/SH)
root@border-leaf6_re>
```

Finally, after all the border-leaf level verification, you may need to quickly verify the information regarding the site B originated MAC@ at the server-leaf level in site B.

```
root@LEAF11_re> show mac-vrf forwarding mac-table instance MACVRF101
MAC flags (S - static MAC, D - dynamic MAC, L - locally learned, P - Persistent static
           SE - statistics enabled, NM - non configured MAC, R - remote PE MAC, O - ovsdb MAC)
Ethernet switching table : 3 entries, 3 learned
Routing instance : MACVRF101
   VLAN                MAC                MAC      Logical                SVLBNH/      Active
   name                address            flags    interface              VENH Index   source
   VLAN1001            00:50:56:ab:01:01  DLR      ae0.0
   VLAN1001            00:50:56:ab:01:03  DR       vtep.32770                     172.16.7.13
  VLAN1001          00:50:56:ab:01:04  DR      esi.1764
00:00:11:11:11:11:11:11:11:11
{master:0}
root@LEAF11_re>
```

You can see that from the site A server-leaf LEAF11 perspective the server4 located in DC site B is reachable via the local fabric border-leaf5/6 iESI 00:00:11:11:11:11:11:11:11:11 and not through his original ESI 00:00:88:88:88:88:88:88:21:22 enabled at LEAF21/LEAF22 in site B.

It means that even if we were learning many MAC@s from site B, the local server leaf will be installing just the local site border-leaf iESI and local next-hops, instead of all the remote site ESIs and next-hops. This aspect helps optimizing the TCAM resources used at the low-end server leaf nodes and simply connect more servers in each data center site without directly impacting the switches scale between two data center sites.

Additionally, you can confirm the local site A server leaf LEAF11 tunnels:

```
root@LEAF11_re> show mac-vrf forwarding VXLAN-tunnel-end-point remote instance MACVRF101
Logical System Name       Id  SVTEP-IP         IFL   L3-Idx    SVTEP-Mode    ELP-SVTEP-IP
<default>                 0   172.16.7.11      lo0.0    0
 RVTEP-IP         L2-RTT                  IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 172.16.7.12      MACVRF101               557       vtep.32771   1762    RNVE
    VNID          MC-Group-IP
```

```
    51001          0.0.0.0
 RVTEP-IP          L2-RTT                    IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 172.16.7.13       MACVRF101                 556       vtep.32770   1722    RNVE
    VNID           MC-Group-IP
    51001          0.0.0.0
 RVTEP-IP          L2-RTT                    IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 172.16.7.115      MACVRF101                 555       vtep.32769   1716    RNVE
    VNID           MC-Group-IP
    51001          0.0.0.0
 RVTEP-IP          L2-RTT                    IFL-Idx   Interface    NH-Id   RVTEP-Mode  ELP-
IP        Flags
 172.16.7.116      MACVRF101                 568       vtep.32772   1763    RNVE
    VNID           MC-Group-IP
    51001          0.0.0.0
{master:0}
root@LEAF11_re>
```

As expected, the LEAF11 in site A just has local site server-leaf and border-leaf tunnels reducing the load of the local node. The tunnels to RVTEP 172.16.7.115 and 172.16.7.116 are the tunnels used to reach the border-leaf5/6 and then stitching at the borders to the MPLS domain in seamless way.

Chapter 7

EVPN-VXLAN T5-to-IPVPN-MPLS Internetworking Implementation and Verification

In the previous scenario we covered the use case of DCI where VXLAN is stitching to MPLS for bridging/switching L2 DCI purposes and where the same VLAN/bridge-domain is stretched between the DC sites.

In some cases, some IP prefixes may become specific to a given DC site and be used for the reachability of services running in the DC by the external users located in the POP locations.

In this case, when there is an existing MPLS core, the border-leaf nodes advertise the site-specific IP prefixes as IPVPN NLRI and the encapsulation format changes from the original EVPN-VXLAN (Type-5 route signalized within the fabric) to MPLS IPVPN at the DCI level. This is the DCI option-6 we highlighted in the summary of the DCI options early on in Chapter 2.

We will break down this option in the following example where two DC sites are communicating using just IP prefixes for a tenant/service. Both sites keep the tenant IP virtualization from the server leaf nodes to the border-leaf. The border leaf changes the encapsulation format from VXLAN to MPLS. In this case, the border-leaf also becomes a PE node from the IPVPN MPLS network perspective.

The routing-instance used for this scenario at the border-leaf is the same one used for Type-5 IP prefix advertisement purposes, however, because the peering between the sites also uses the inet-vpn unicast type, the IP prefixes we receive from remote location PE will have the precedence for the MPLS encapsulation and will be installed in the forwarding table with the LSP label push/pop information.

The topology used is shown in Figure 7.1, where two sites are connected to each other via the core MPLS network.

Prefixes from one site to the other are reflected by the core route-reflector which is also used as a P device. The intermediate MX tap node in the middle is used here for lab examination of the encapsulation and is usually not used in real deployments.

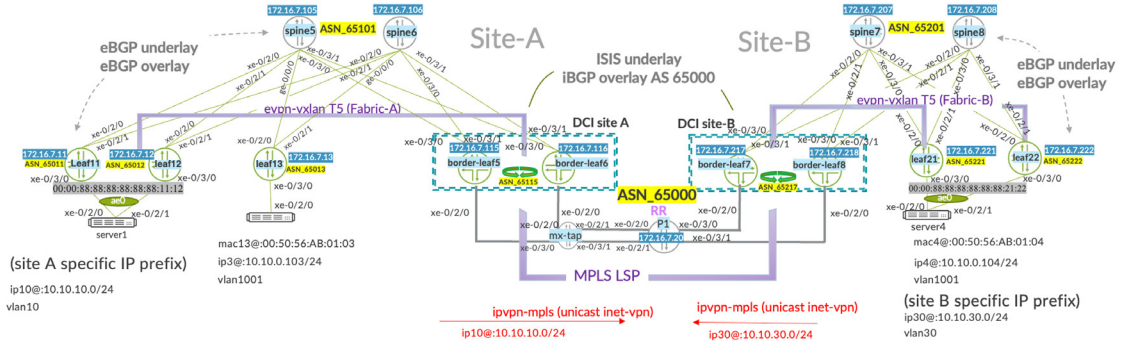


Figure 7.1 *EVPN-VXLAN Pure Type-5 to IPVPN-MPLS Internetworking DCI*

The fabric running in Site-A/Site-B is EVPN-VXLAN Edge Routed Bridging design (ERB) and the server leaf11/leaf12 in site A originates the EVPN Type-5 prefix 10.10.10.0/24 which is specific to this pair of leaf nodes. This prefix is received by all other leaf nodes of the fabric via EVPN Type-5 and therefore VXLAN encapsulation is used by leaf13 for reachability of server1 within fabric-A. When the prefix of server1 is received at border-leaf5/border-leaf6, it is translated to an IPVPN-MPLS advertisement and is sent to the core IP route-reflector (RR) as inet-vpn unicast NLRI, which then reflects it to the DC site B border-leaf7/8 PE nodes.

With this implementation, leaf11/leaf12 in site A and leaf21/leaf22 in site B are not forming any direct Type-5 EVPN VXLAN tunnel and the border-leaf is becoming the DC gateway for the bidirectional IP reachability. leaf21/22 will get the prefix 10.10.10.0/24 needed for server1 reachability as the Type-5 EVPN-VXLAN information because the border-leaf7/8 performs the translation from IPVPN-MPLS -to-EVPN-VXLAN pure Type-5.

Consequently, leaf21/22 instead uses the local border-leaf nodes for all server-leaf nodes of the remote location, efficiently reducing the number of next-hops used for the end-to-end IP reachability.

The BGP underlay and overlay inside the fabric is following the same approach explained in the previous chapter: eBGP underlay from leaf to spine and eBGP overlay from leaf to spine with an explicit no-next-hop change at the spines.

Each server leaf gets a unique BGP ASN number and the same number is used for underlay and overlay. Both lean spines are using the same BGP ASN number for the overlay.

Implementation and Verification

Inside the core-IP network, we already have a route-reflector deployed so border-leaf nodes in both sites are joining that overlay BGP ASN 65000. The undelay between the DC sites is running ISIS and the only advertisement is for the loopback reachability of the border-leaf nodes.

Here are leaf11/leaf12 configurations for pure Type-5 EVPN-VXLAN instance that delivers connectivity to server1:

Config 35 leaf11 EVPN-VXLAN Type-5 to IPVPN-MPLS internetworking – Type-5 instance config

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.11/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1
set routing-instances T5-VRF1 interface irb.10
set routing-instances T5-VRF1 interface irb.20
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.11:100
set routing-instances T5-VRF1 vrf-target target:5:8888
set routing-instances T5-VRF1 vrf-table-label
set interfaces irb unit 10 family inet address 10.10.10.1/24  >> anycast IP gateway at leaf11 specific
to leaf11/leaf12
set interfaces irb unit 10 mac 00:00:01:01:01:01
set interfaces irb unit 20 family inet address 10.10.20.1/24
set interfaces irb unit 20 mac 00:00:01:01:01:01
## The policy statement configuration aka route-maps is used to define specifically which IP prefixes
will be advertised as Type-5 EVPN within the IP fabric
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.11/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.11/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.10.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.20.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
```

Config 36 leaf12 EVPN-VXLAN Type-5 to IPVPN-MPLS internetworking – Type-5 instance config

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.12/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1 >> we define
what to advertise as IP
set routing-instances T5-VRF1 interface irb.10  >> IRB interface used for server1 IP anycast gateway
```

```
set routing-instances T5-VRF1 interface irb.20
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.12:100  >> unique RD value per each node in
the fabric
set routing-instances T5-VRF1 vrf-target target:5:8888  >> same route target will be used at the
border-leaf6/7
set routing-instances T5-VRF1 vrf-table-label
set interfaces irb unit 10 family inet address 10.10.10.1/24  >> anycast IP gateway at leaf12 specific
to leaf11/leaf12
set interfaces irb unit 10 mac 00:00:01:01:01:01
set interfaces irb unit 20 family inet address 10.10.20.1/24
set interfaces irb unit 20 mac 00:00:01:01:01:01
## The policy statement configuration aka route-maps is used to define specifically which IP prefixes
will be advertised as Type-5 EVPN within the IP fabric
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.12/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.12/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.10.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.20.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
```

As server1 is multihomed, MAC-VRF is provisioned on both leaf nodes. This is something not relevant for the Type-5 EVPN-VXLAN-to-IPVPN-MPLS internetworking, but we reference it here for the completeness of the solution. The leaf nodes in our lab example are also vMX nodes so the mac-VRF configuration is using the bridge-domain instead of VLAN convention. The QFX5K MAC-VRF configuration examples can be checked in Chapter 6 where we analyze in detail the VXLAN-to-VXLAN stitching use-cases.

Config 37 leaf11 - Mac-VRF Config for ESI-LAG Purposes to Connect Server1

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-instances MACVRF101 protocols evpn encapsulation vxlan
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 5010
set routing-instances MACVRF101 protocols evpn extended-vni-list 5020
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 bridge-domains bd10 vlan-id 10  >> MX uses the 'bridge-domain' & QFX
the 'VLAN' approach
set routing-instances MACVRF101 bridge-domains bd10 routing-interface irb.10  >> we associate IRB.10
with VLAN10 aka bd10
set routing-instances MACVRF101 bridge-domains bd10 vxlan VNI 5010
set routing-instances MACVRF101 bridge-domains bd1001 vlan-id 1001
set routing-instances MACVRF101 bridge-domains bd1001 routing-interface irb.1001
set routing-instances MACVRF101 bridge-domains bd1001 vxlan VNI 51001
set routing-instances MACVRF101 bridge-domains bd20 vlan-id 20
set routing-instances MACVRF101 bridge-domains bd20 routing-interface irb.20
set routing-instances MACVRF101 bridge-domains bd20 vxlan VNI 5020
set routing-instances MACVRF101 service-type vlan-aware
set routing-instances MACVRF101 interface ae0.0
set routing-instances MACVRF101 route-distinguisher 172.16.7.11:1
```

```
set routing-instances MACVRF101 vrf-target target:1:8888  >> shared route-target between leaf11/leaf12
set interfaces ae0 mtu 9100
set interfaces ae0 esi 00:00:88:88:88:88:88:88:11:12  >> ESI value for server1 L2 multihoming
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:01:88:88:01:01
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 unit 0 family bridge interface-mode trunk
set interfaces ae0 unit 0 family bridge VLAN-id-list 10  >> vlan-id for the IRB.10 with prefix
10.10.10.0/24
set interfaces ae0 unit 0 family bridge VLAN-id-list 20
set interfaces ae0 unit 0 family bridge VLAN-id-list 1001
set interfaces xe-0/3/0 gigether-options 802.3ad ae0  >> we associate the given physical interface with
the AE0 aggregate
set chassis aggregated-devices ethernet device-count 2  >> we just decided to use 2 AE interfaces
typically around 24 or 48 AE aggregated interfaces can be enabled locally to connect servers
set chassis network-services enhanced-ip  >> needed only when MX is used as a leaf node
```

Config 38 leaf12 - mac-VRF config for ESI-LAG purposes to connect server1

```
set routing-instances MACVRF101 instance-type mac-vrf
set routing-instances MACVRF101 protocols evpn encapsulation vxlan
set routing-instances MACVRF101 protocols evpn default-gateway no-gateway-community
set routing-instances MACVRF101 protocols evpn extended-vni-list 5010
set routing-instances MACVRF101 protocols evpn extended-vni-list 5020
set routing-instances MACVRF101 protocols evpn extended-vni-list 51001
set routing-instances MACVRF101 vtep-source-interface lo0.0
set routing-instances MACVRF101 bridge-domains bd10 vlan-id 10
set routing-instances MACVRF101 bridge-domains bd10 routing-interface irb.10
set routing-instances MACVRF101 bridge-domains bd10 vxlan VNI 5010
set routing-instances MACVRF101 bridge-domains bd1001 vlan-id 1001
set routing-instances MACVRF101 bridge-domains bd1001 routing-interface irb.1001
set routing-instances MACVRF101 bridge-domains bd1001 vxlan VNI 51001
set routing-instances MACVRF101 bridge-domains bd20 vlan-id 20
set routing-instances MACVRF101 bridge-domains bd20 routing-interface irb.20
set routing-instances MACVRF101 bridge-domains bd20 vxlan VNI 5020
set routing-instances MACVRF101 service-type vlan-aware
set routing-instances MACVRF101 interface ae0.0
set routing-instances MACVRF101 route-distinguisher 172.16.7.12:1
set routing-instances MACVRF101 vrf-target target:1:8888
## the following ESI-LAG configuration was used for the multihoming purposes of server1 – leaf11/leaf12
share same ESI value
set interfaces ae0 mtu 9100
set interfaces ae0 esi 00:00:88:88:88:88:88:88:11:12
set interfaces ae0 esi all-active
set interfaces ae0 aggregated-ether-options lacp active
set interfaces ae0 aggregated-ether-options lacp system-id 00:01:88:88:01:01
set interfaces ae0 aggregated-ether-options lacp admin-key 1
set interfaces ae0 unit 0 family bridge interface-mode trunk
set interfaces ae0 unit 0 family bridge VLAN-id-list 10  >> VLAN used to connect server1
set interfaces ae0 unit 0 family bridge VLAN-id-list 20
set interfaces ae0 unit 0 family bridge VLAN-id-list 1001
set interfaces xe-0/3/0 gigether-options 802.3ad ae0  >> we associate the ae0 aggregate with the
specific physical port
set chassis aggregated-devices ethernet device-count 2  >> this is to enable globally the aggregated
interfaces
set chassis network-services enhanced-ip  >> only required for MX used as server or border-leaf node
```

Once leaf11/leaf12 in site A are provisioned, we can focus on demonstrating border-leaf5/6 conversion of EVPN-VXLAN Type-5 prefixes to IPVPN-MPLS. As a matter of fact, when border-leaf is not used for L2 extension in parallel, the MAC-VRF configuration is not even needed at the borders.

Border-leaf5 is simply provisioned with the same routing-instance as leaf11/leaf12 using a common route-target and, additionally, family inet-vpn unicast is enabled at the core-IP iBGP peering towards the route-reflector. When the DCI solution is a combination of IPVPN internetworking and RFC9014 for VXLAN-to-MPLS, EVPN signalization also needs to be enabled at the peering level.

The following example shows the IPVRF configuration commonly used for IPVPN and Type-5 EVPN-VXLAN purposes. The protocol EVPN part is used at the fabric level, while the rest of the IP prefix advertisements transformation is done automatically at the BGP level just because we enabled the inet-vpn unicast as a family at the WAN iBGP peering level.

The level of control we suggest in this implementation is only regarding the prefixes we inject at the border-leaf from the core-IP. Here, this is done by using the policy-statement called my-t5-export-VRF1 where we simply define the prefixes received from the remote location that can be advertised towards the local DC.

Config 39 EVPN-VXLAN Type-5 to IPVPN internetworking – border-leaf5 in site A

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.115/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100  >> routing VNI just for fabric
LAN purposes
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1  >> we define
which IP to advertise
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.115:100
set routing-instances T5-VRF1 vrf-target target:5:8888  >> route-target same as on leaf11/leaf12 for
Type-5 instance
set routing-instances T5-VRF1 vrf-table-label
set chassis network-services enhanced-ip  >> needed for MX used as border-leaf – for example mx304
## define which prefixes will be advertised into the fabric A as Type-5 EVPN, received from the remote
location site-B
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.115/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.115/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.30.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.40.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
```

As border-leaf5 is not enabled for any L2 stretch purposes, we don't have to provision any MAC-VRF instance.

A similar configuration is deployed on border-leaf6, where only the IPVRF is needed and the overlay iBGP EVPN signalization towards the core-IP RR.

Config 40 EVPN-VXLAN Type-5 to IPVPN-MPLS internetworking configuration - border-leaf6 in site A

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.116/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.116:100
set routing-instances T5-VRF1 vrf-target target:5:8888  >> same route-target as the local leaf11/12
set routing-instances T5-VRF1 vrf-table-label  >> label allocated per IPVPN
set chassis network-services enhanced-ip  >> required at the MX level when used as a border-leaf
## we define the remote site prefixes that will be injected to the local fabric in site A as Type-5
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.115/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.115/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.30.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.40.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
```

To enable internetworking between VXLAN and MPLS, the iBGP overlay peering is used towards the core IP WAN route-reflector 172.16.7.20.

Config 41 border-leaf5 iBGP WAN DCI peering for IPVPN internetworking use-case

```
set protocols bgp group underlay type external
set protocols bgp group underlay export my_underlay_export
set protocols bgp group underlay local-as 65115
set protocols bgp group underlay multipath multiple-as
set protocols bgp group underlay neighbor 192.168.7.1 peer-as 65101  >> underlay peering to the lean
spines in site A
set protocols bgp group underlay neighbor 192.168.8.1 peer-as 65101
set protocols bgp group overlay type external
set protocols bgp group overlay multihop no-nexthop-change
set protocols bgp group overlay local-address 172.16.7.115
set protocols bgp group overlay family EVPN signaling
set protocols bgp group overlay local-as 65115
set protocols bgp group overlay multipath multiple-as
set protocols bgp group overlay bfd-liveness-detection minimum-interval 100
set protocols bgp group overlay bfd-liveness-detection multiplier 3
set protocols bgp group overlay neighbor 172.16.7.105 peer-as 65101   >> eBGP peering towards the local
spines in site A
set protocols bgp group overlay neighbor 172.16.7.106 peer-as 65101
```

```
set protocols bgp group overlay vpn-apply-export
set protocols bgp group WAN type internal  >> for DCI in this use-case we use iBGP and route-reflector
from the core IP
set protocols bgp group WAN local-address 172.16.7.115
set protocols bgp group WAN family inet-vpn unicast  >> required for Type-5 EVPN to IPVPN
internetworking
set protocols bgp group WAN family EVPN signaling  >> required when in parallel the border-leaf
performs the VXLAN to #MPLS stitching for bridging L2 purposes (rfc9014 DCI use-case)
set protocols bgp group WAN local-as 65000
set protocols bgp group WAN neighbor 172.16.7.20
set protocols bgp group WAN vpn-apply-export
```

The border-leaf in site A and B are connected as PE nodes to the core IP MPLS so they use ISIS underlay to reach the route-reflector and the remote site loopback IP addresses.

Here are the configurations used for ISIS and MPLS.

Config 42 border-leaf5 underlay ISIS and MPLS LDP configuration

```
set protocols isis interface xe-0/2/0.0  >> underlay interface used to connect to the core-IP MPLS
set protocols isis interface lo0.0  >> loopback used for IPVPN-MPLS purposes
set protocols isis level 1 disable  >> we decided to use only level 2 ISIS for simplicity
set protocols ldp interface xe-0/2/0.0  >> to advertise the MPLS labels we use the LDP at the border-
leaf in both DC sites
set protocols ldp interface lo0.0
set protocols mpls interface xe-0/2/0.0  >> interface connected to the core MPLS is enabled at the
MPLS protocol level
set interfaces xe-0/2/0 mtu 9216
set interfaces xe-0/2/0 unit 0 family inet address 192.168.14.2/24
set interfaces xe-0/2/0 unit 0 family iso  >> to run the ISIS underlay we enable that family
explicitly at the underlay interface
set interfaces xe-0/2/0 unit 0 family mpls  >> we also specify the MPLS family for the underlay port
connected to the core
set interfaces xe-0/3/0 unit 0 family inet address 192.168.7.2/24
set interfaces xe-0/3/1 unit 0 family inet address 192.168.8.2/24
set interfaces lo0 unit 0 family inet address 172.16.7.115/32 primary
set interfaces lo0 unit 0 family iso address 49.0001.1720.1600.7115.00  >> is required for the ISIS to
be used for DCI underlay
set interfaces lo0 unit 1 family inet address 172.16.100.115/32
```

To explain how site B is configured, we focus on border-leaf7/8 and server leaf21/21. The same approach is used in site B: tenant identification with route-target information shared between the IPVPN domain (DCI) and fabric EVPN-VXLAN Type-5.

Config 43 Type-5 EVPN-VXLAN to IPVPN MPLS internetworking - leaf7 in site B

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.217/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes vni 1100  >> RVNI used within the
fabric only in this use-case
```

```
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1 >> define which
prefixes to inject
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.217:100
set routing-instances T5-VRF1 vrf-target target:5:8888  >> the same route-target in site B is used as
in the site A
set routing-instances T5-VRF1 vrf-table-label
##define which IP prefixes from site A will be injected as Type-5 EVPN prefixes on site B
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.217/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.217/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.10.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.20.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
set chassis network-services enhanced-ip
```

The similar configuration is also enabled on border-leaf8 in DC site B:

```
set routing-instances T5-VRF1 instance-type vrf
set routing-instances T5-VRF1 routing-options static route 10.10.100.218/32 discard
set routing-instances T5-VRF1 routing-options multipath
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes advertise direct-nexthop
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes encapsulation vxlan
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes VNI 1100
set routing-instances T5-VRF1 protocols evpn ip-prefix-routes export my-t5-export-VRF1  >> to define
the prefixes to be
## injected into the site B fabric when originated in site A
set routing-instances T5-VRF1 interface lo0.1
set routing-instances T5-VRF1 route-distinguisher 172.16.7.218:100
set routing-instances T5-VRF1 vrf-target target:5:8888  >> the key aspect is to make sure route-
targets are same on both sites
set routing-instances T5-VRF1 vrf-table-label
set policy-options policy-statement my-t5-export-VRF1 term term1 from route-filter 172.16.100.218/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.100.218/32
exact
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.20.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 from route-filter 10.10.30.0/24
orlonger
set policy-options policy-statement my-t5-export-VRF1 term term2 then accept
set chassis network-services enhanced-ip  >> this is needed for the MX as a border-leaf platform
```

Once all the configurations are in place, we want to verify the data plane part before and after the stitching from VXLAN to MPLS at border-leaf5/6. We send an ICMP echo request from server1 located in DC A to server4 located in DC B. And we observe the following data plane packet capture (Figure 7.2) with two different encapsulation formats.

First with VXLAN when packet arrives at border-leaf5. It is using VNI 1100.

Then we observe it in MX_tap node, inside the core MPLS network, after the stitching occurs.

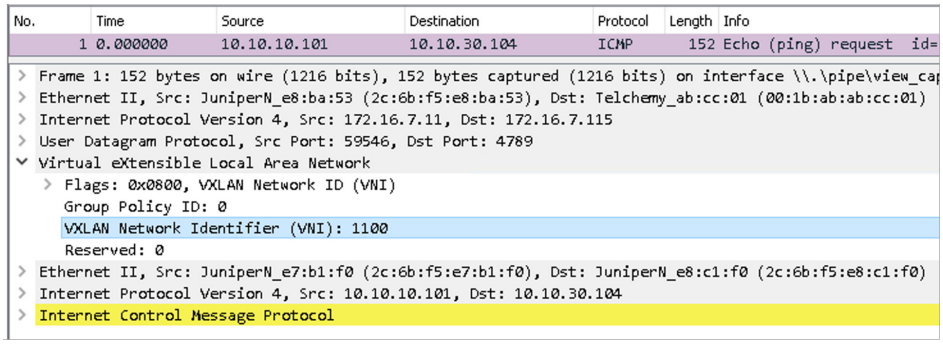


Figure 7.2 *The ICMP Packet Before IP Stitching From Type-5 EVPN-VXLAN Domain To IPVPN-MPLS (Ipvpn Internetworking)*

After stitching from VXLAN to MPLS, using IPVPN EVPN internetworking, the data packet is using transport MPLS label 299952 and label 18 for the given IPVPN. See Figure 7.3.

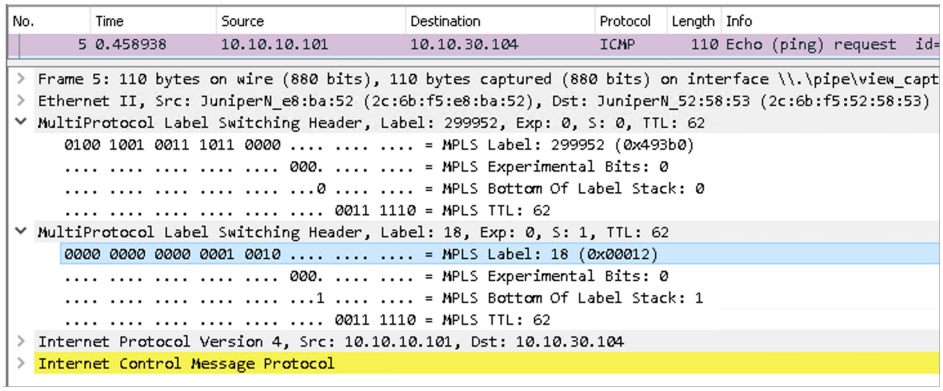


Figure 7.3 *The ICMP Packet After IP Stitching from Type-5 EVPN-VXLAN Domain to IPVPN-MPLS (Ipvpn Internetworking)*

We are also able to confirm, on border-leaf5, which MPLS label is used towards the core IP to reach server4 in DC site B:

```
root@border-leaf5# run show route forwarding-table destination 10.10.30.104/32 vpn T5-VRF1 detail
Routing table: T5-VRF1.inet
Internet:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
10.10.30.104/32    user     0                    ulst  1048577     8
                                                 indr  1048587     5
                            192.168.14.1         Push 18, Push 299936(top)      653     2 xe-0/2/0.0
                                                 indr  1048582     4
                            192.168.14.1         Push 18, Push 299952(top)      652     2 xe-0/2/0.0

[edit]
root@border-leaf5#
```

We can also see the two different transport MPLS labels – 299936 as well as 299952 because the destination prefix 10.10.30.104/32 (server4 in site B) is reachable through two borders, border-leaf7/8. The IPVPN inner label18 allocated to the given IPVPN is common and will be used for the final recursive resolution at the border-leaf nodes7/8 in DC site B.

Getting to the bottom of the MPLS and the IP destination is important but what about checking the source IP 10.10.10.101/32 from the border-leaf5 perspective?

You can trace this back using the same approach as before by additionally calling the pfe for recursive resolution of the next-hop-id:

```
root@border-leaf5> show route forwarding-table destination 10.10.10.101/32 vpn T5-VRF1
Routing table: T5-VRF1.inet
Internet:
Destination        Type RtRef Next hop           Type Index    NhRef Netif
10.10.10.101/32    user     0                    ulst  1048592     6
                                                 indr  1048585     3
                                                 comp      591     2
                                                 indr  1048590     3
                                                 comp      594     2
root@border-leaf5> request pfe execute command “show nhdb id 591 det” target fpc0
SENT: Ukern command: show nhdb id 591 det
   ID     Type     Interface    Next Hop Addr    Protocol      Encap    MTU            Flags  PFE
internal Flags
-----  --------  -------------  ---------------  ----------  ------------  ----  ------------------
------------------
  591   Compst  -           -                 IPv4          -    0  0x0000000000000000
BFD Session Id: 0
Composite NH:
  Function: Tunnel Function
  Hardware Index: 0x0
  Composite flag: 0x0
  Composite pfe flag: 0xe
  Lower-level NH Ids:
  Derived NH Ids:
  Tunnel Data:
     Type    : VXLAN
     Tunnel ID: 806354950
     Encap VRF: 0
     Decap VRF: 8
     MTU     : 0
     Flags   : 0x0
     AnchorId : 0
     Encap Len: 53
     Mode    : Encap-Decap
     Encap   : 0x01 0x73 0x07 0x10 0xac 0x00 0x00 0x00
               0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
               0x00 0x0b 0x07 0x10 0xac 0x00 0x00 0x00
               0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
               0x00 0x4c 0x04 0x00 0x00 0x4c 0x04 0x00
               0x00 0x2c 0x6b 0xf5 0xe7 0xb1 0xf0 0x2c
               0x6b 0xf5 0xe8 0xc1 0xf0
     Data Len : 0
     Encap VXLAN id: 1100
     Decap VXLAN id: 1100
     Src mac: 2c.6b.f5.e8.c1.f0
```

```
      Dst mac: 2c.6b.f5.e7.b1.f0
TunnelModel:
Dynamic Tunnel Model:
         Name = VXLAN <src: 172.16.7.115, dst: 172.16.7.11>
         ID   = 330
         MTU = 0
         VRF = T5-VRF1.8(8)
         Source Entropy = 1
         Carry Hash = 1
         Reassemble = 0
         Packets = 0 Bytes = 0
Source IP     : 172.16.7.115
Destination IP: 172.16.7.11
```

We can observe that from border-leaf5, to reach out server1, the VXLAN encapsulation is used towards leaf11 (172.16.7.11) and is mapped as dynamic tunnel for the Type-5 IPVPN called T5-VRF1.

Chapter 8

Seamless EVPN-VXLAN Tunnel Stitching Conclusion

After all the implementation examples from Chapters 6 and 7 covering VXLAN-to-VXLAN and VXLAN-to-MPLS data center interconnect options, we think most readers will be comfortable discussing more specific implementations and be able to replicate the example in a more complex scenario – for example when more than just two data center sites are used.

The various data center interconnect options covered in this book shows how flexible the EVPN control plane has become over the last couple of years. It proves that it can be used in greenfield scenarios, stitching VXLAN-to-VXLAN, as well as brownfield where sometimes VXLAN-to-MPLS stitching is a better choice to quickly interconnect remote data centers, as well as larger scale POP locations or remote campuses sites, allowing them to access data from different data center locations.

From the outcomes point of view, we think the scaling optimization (reduced number of tunnels, next hops, logical interfaces) is one part of the story, while the second is purely related to operations where we fully control which workloads gets extended to which remote location.

As a matter of fact, some data may need to be replicated just in some locations while other may need to go to a different location due to legislation rules of the given organization.

This book is covering six different DCI options with a bigger focus on option-3 (EVPN-VXLAN-to-EVPN-VXLAN for L2 bridging/switching purposes) and option-5 (EVPN-VXLAN-to-EVPN-VXLAN for L3 IP prefix advertisement purposes). While the MPLS DCI is still very popular, we think, in many greenfield environments, that the VXLAN-to-VXLAN option-3/5 are sufficient and fulfill most of the requirements, opening it with additional emerging use cases where Group Based Policy (GBP) profiles are also extended between DC sites for micro-segmentation purpose.

We believe DCI options may also evolve to SRv6 use cases whenever the new core-IP infrastructure moves to SRv6-based signalization. This is something we believe may happen for greenfield 5G core infrastructure and will be mainly needed for the IP prefixes site and domain segmentation.

We also covered the new domain path BGP attribute (D-PATH) which automatically protects the DCI solution from any routing loops and avoids implementation complexities at the border-leaf level.

The security of the DCI was also highlighted and can become more important for some organizations. For example, when, instead of traditional service chaining, we decided to use firewalls as border-leaf devices.

The DCI options covered in this book, deployed at a larger number of sites, will have full success when automated by the Apstra intent-based networking tool.

For the control of the operation and management, additional protocols like BFD inside VXLAN may be used while, in the case of MPLS stitching, the traditional OAM MPLS toolset will stay very important for the monitoring of the state of the DCI.