

Interoperability of EOS MLAG and Junos EVPN-VXLAN Anycast Multihoming with Egress Link Protection

Author: Shalini Mukherjee

Published: 8/27/2026

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Juniper Networks, the Juniper Networks logo, Juniper, and Junos are registered trademarks of Juniper Networks, Inc. in the United States, and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Copyright © 2026 Juniper Networks, Inc. All rights reserved.

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS (Operating System) has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <https://support.juniper.net/support/eula/>. By downloading, installing, or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

<i>Table of Contents</i>	3
<i>Executive Summary</i>	4
<i>Introduction</i>	4
<i>Lab Topology and Baseline Behavior</i>	5
<i>Anycast Multihoming for Symmetric IRB</i>	6
<i>Anycast + Symmetric Type-2 with Egress Link Protection</i>	8
<i>Global Repair</i>	10
<i>Blocking Advertisement of Ethernet A-D per ES Route with an All-zero ESI</i>	12
<i>Anycast Multihoming for BUM</i>	13
<i>Auto ESI Based on Type-1 LACP</i>	15
<i>Configurations</i>	17
<i>Event Script for EVPN Core Isolation</i>	22
<i>Verification and Troubleshooting</i>	26
<i>Benefits of Anycast Multihoming Compared to Regular ESI</i>	32
<i>Failover Test Case With and Without Anycast</i>	33

Executive Summary

Modern data center fabrics are increasingly built from a mix of networking platforms and redundancy models. A common interoperability challenge arises when an EVPN-VXLAN fabric must connect devices using **EVPN all-active multihoming** with platforms that use a traditional **MLAG-based anycast VTEP** architecture. Although both approaches provide multichassis resiliency, they represent redundant VTEPs differently, which can complicate forwarding behavior, convergence, and operational consistency across a multivendor fabric.

This paper addresses that challenge by demonstrating how **Junos OS Evolved anycast multihoming** can be used to present a pair of Juniper leaf switches as a single logical VXLAN tunnel endpoint to the rest of the EVPN-VXLAN fabric. In the validated design, both Junos leaves share the same anycast VTEP address while retaining unique loopback addresses for control-plane reachability and local forwarding functions. The pair also uses a common Ethernet Segment Identifier (ESI), LACP system ID, and anycast gateway attributes for the dual-homed downstream segment.

The primary benefit of this approach is that **remote devices do not need to understand or track the individual Junos multihoming members as separate service VTEPs**. Instead, the fabric forwards traffic toward a shared anycast VTEP, while the Junos pair determines how traffic is delivered across the active multihoming members. The configuration specifically enables anycast-multihoming mlag-interop-mode, allowing the Junos pair to participate in an architecture designed to interoperate with MLAG-based peers.

When Customers Should Consider This Design

Customers should consider an anycast VTEP design when they are:

- Building or migrating to a **multivendor EVPN-VXLAN fabric** where some leaf pairs use MLAG-based anycast VTEPs and Juniper leaves use EVPN multihoming.
- Introducing Juniper switches into an existing environment and want to **preserve a common logical VTEP model** rather than redesigning the surrounding fabric.
- Looking for active-active downstream connectivity while maintaining consistent VXLAN reachability and minimizing vendor-specific dependencies on remote switches.
- Performing phased migrations where traditional MLAG and EVPN multihoming technologies must coexist.

A conventional **EVPN all-active multihoming design without an anycast VTEP** may be preferable when the entire fabric natively supports EVPN multihoming and there is no requirement to interoperate with an MLAG-style VTEP architecture. In that environment, standard EVPN procedures can provide the required redundancy without introducing an additional shared VTEP abstraction.

Similarly, a traditional **MLAG or MC-LAG design** may remain appropriate for smaller Layer 2 environments where VXLAN, EVPN control-plane signaling, fabric-scale mobility, or multivendor EVPN interoperability are not required.

Introduction

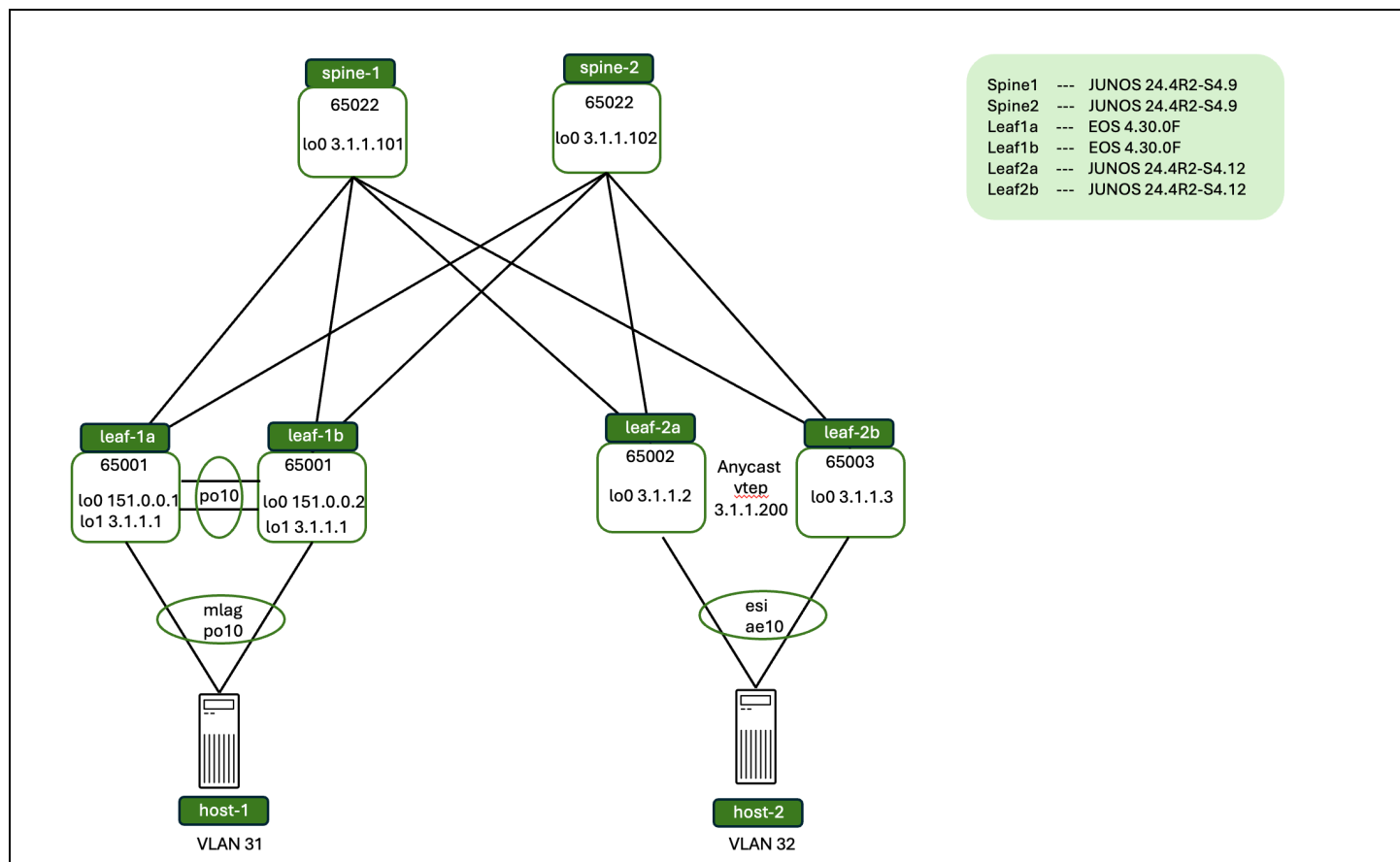
The purpose of this paper is to provide customers and network architects with a **practical reference for deploying Junos anycast VTEP in a multivendor EVPN-VXLAN environment**. It explains the architecture, required configuration, control-plane behavior, forwarding operation, and interoperability considerations using Junos OS Evolved and Arista EOS as the reference platforms.

The validation focuses not only on configuration correctness, but also on operational behavior—including shared VTEP reachability, EVPN route exchange, symmetric IRB forwarding, multihoming state, and traffic continuity. The goal is to help customers understand where this design fits, what problem it solves, and how it can be introduced into an existing data center fabric with predictable forwarding and resiliency.

Lab Topology and Baseline Behavior

Before explaining anycast multihoming, it is important to anchor the discussion to the lab. The feature is easier to explain when everyone understands which device is acting as the ingress VTEP, which devices are the multihoming peers, and which side is the interop peer.

Lab topology used in this document



Important lab values

Item	Value in the lab	Why it matters
Host-1	31.1.1.10/24 in VLAN 31	Source host behind the EOS MLAG pair.
Host-2	32.1.1.10/24 in VLAN 32	Destination host behind the Junos ESI pair.
EOS Leaf-1a/1b	MLAG pair, AS 65001	Remote interop side. Acts as ingress VTEP for tests from Host-1 to Host-2.
Junos Leaf-2a	ESI peer, AS 65002, unique VTEP 3.1.1.2	One local multihoming peer for Host-2.

Junos Leaf-2b	ESI peer, AS 65003, unique VTEP 3.1.1.3	Second local multihoming peer for Host-2.
Junos spines	AS 65022	Transit EVPN overlay nodes. Must preserve EVPN next-hop.
L2 VNI VLAN 31	10031	EOS local bridge domain for Host-1.
L2 VNI VLAN 32	10032	Junos local bridge domain for Host-2.
L3 VNI	3333	Tenant VRF VNI for symmetric IRB.
IP-VRF RT	target:3:3	Common tenant routing domain.

Anycast Multihoming for Symmetric IRB

What problem this chapter solves

In normal EVPN multihoming, remote nodes may see two separate egress leaf VTEPs for a dual-homed CE. In the lab, those VTEPs are 3.1.1.2 for Leaf-2a and 3.1.1.3 for Leaf-2b. Anycast multihoming changes the model so that the remote side can treat Leaf-2a and Leaf-2b as one logical egress VTEP.

How it maps to the lab

Before anycast MH	After anycast MH
EOS may learn Junos reachability through unique VTEPs 3.1.1.2 and 3.1.1.3.	EOS learns Junos MH reachability through one anycast VTEP, for example 3.1.1.200.
Remote traffic may be tied to a specific egress leaf.	Remote traffic is sent to the anycast VTEP and underlay ECMP chooses Leaf-2a or Leaf-2b.
Classic aliasing/ESI route handling is more visible to the remote node.	The pair is presented as a single logical endpoint in interop mode.

Symmetric IRB path for Host-1 to Host-2

```

Host-1 3.1.1.10
|
| VLAN 31 / L2 VNI 10031
v
EOS Leaf-1a/1b MLAG pair
|
| Route lookup in VRF1_IP_VRF

```

```

| VXLAN encapsulation using L3 VNI 3333
v
EVPN-VXLAN fabric
|
| Tunnel endpoint = Junos anycast VTEP 3.1.1.200
v
Junos Leaf-2a or Leaf-2b
|
| VXLAN decapsulation
| IP-VRF lookup
| local L2 forwarding in VLAN 32 / VNI 10032
v
Host-2 32.1.1.10

```

What changes in EVPN route advertisement

With anycast multihoming in interop mode, the multihoming PEs advertise EVPN routes so that the remote fabric sees the shared anycast VTEP as the forwarding endpoint. For example, instead of the remote PE forwarding to individual VTEPs PE1 and PE2, it forwards to the anycast VTEP shared by the multihoming pair. This means the remote PE has a simplified forwarding view:

Destination MAC/IP or prefix -> Anycast VTEP

The unique VTEPs of the individual multihoming PEs may still be present in the control plane for repair, reroute, ELP, or global repair behavior, but the normal forwarding endpoint presented to the remote fabric is the anycast VTEP.

Why Type-2 symmetric route is important

For symmetric IRB, a Type-2 MAC/IP route can be used not only for L2 reachability but also for routed host reachability. The route carries information that lets the remote node route through the L3 VNI and still know how to reach the destination host at the egress side.

Information in symmetric Type-2	Meaning
Host MAC/IP	Identifies the destination host, such as Host-2 32.1.1.10.
L2 VNI	Identifies the bridge domain, such as VNI 10032 for VLAN 32.
L3 VNI	Identifies the tenant routing domain, such as VNI 3333.
Protocol next-hop	In interop mode, this should be the anycast VTEP.
Router MAC	Advertised as the anycast router MAC for symmetric IRB forwarding.

Why the MH peer should not install certain peer routes

The Junos MH peer can receive a Type-2 symmetric route from the other MH peer. Because the route uses the anycast VTEP as next-hop, that next-hop may be the same anycast IP configured locally. If the leaf blindly installs that host route in the IP-VRF, it could create incorrect recursion or ambiguous forwarding. Therefore, the implementation must recognize that this route belongs to the local anycast multihoming group and handle it differently.

For the MH peer: if the route next-hop is my own anycast VTEP, do not treat it like a normal remote route. Use the ELP/reroute information and local ESI state instead.

Summary

- Anycast multihoming is not just an anycast gateway IP/MAC feature. It changes how EVPN routes advertise VTEP reachability for a multihomed ESI pair.
- In the lab, the EOS side should see the Junos pair through a shared anycast VTEP, not as two unrelated VTEPs.
- The L3 VNI 3333 still matters. Anycast does not replace symmetric IRB; it improves how the egress multihoming pair is represented.
- The unique VTEPs are still needed internally for reroute/ELP behavior even though the remote side may use the anycast VTEP.

Anycast + Symmetric Type-2 with Egress Link Protection

What problem ELP solves

Anycast multihoming allows the remote ingress node to send traffic to one anycast VTEP. The underlay may deliver the packet to either Leaf-2a or Leaf-2b. That is good for load sharing and node failure behavior, but it creates one important problem: what happens if traffic lands on a leaf whose local access link to Host-2 is down?

ELP solves the link-failure case. If Leaf-2a is alive but its local link to the CE is down, Leaf-2a can redirect traffic to Leaf-2b over an ELP tunnel instead of dropping it.

Important feature relationship note: ELP and anycast multihoming are separate feature capabilities. Anycast multihoming controls how the multihomed pair is represented to the remote EVPN fabric, for example by using a shared anycast VTEP such as 3.1.1.200. ELP controls local egress-link repair between the multihoming peers. They are independent, but they complement each other. In this guide we discuss them together because the best access-link failure behavior in an anycast VTEP design comes from combining anycast forwarding with ELP local repair.

ELP and anycast are independent but complementary

ELP does not require anycast multihoming, and anycast multihoming does not automatically mean ELP is doing local repair. Think of them as two separate blocks in the design:

- ELP function protects the CE-facing or access-link failure case by redirecting traffic from one MH peer to the other peer over a reroute/ELP path.
- Anycast multihoming function presents the MH pair as one logical VTEP to the remote fabric and lets the underlay ECMP choose the physical egress leaf.

When both are used together, traffic can be sent to the shared anycast VTEP, and if the selected leaf has lost its local access path, that selected leaf can use ELP to send the packet to the surviving peer. If anycast is present but ELP/local repair is not active, packets may still land on a leaf with a failed access link and can be dropped until global repair or control-plane convergence changes the forwarding path.

Steady-state forwarding

```
Host-1 -> EOS Leaf -> L3 VNI 3333 -> Anycast VTEP 3.1.1.200
|
+-- Underlay ECMP lands on Leaf-2a
|
+-- Leaf-2a decapsulates VXLAN
+-- Leaf-2a does IP-VRF lookup
+-- Leaf-2a resolves Host-2 via ARP/ND
+-- Leaf-2a forwards L2 packet to Host-2
```

In steady state, the selected Junos leaf forwards directly to the CE over its local access bundle. In the lab, that access bundle is ae10 with VLAN 32.

The VPLAG/ELP next-hop idea

When the destination host is behind an ESI, the ARP/ND next-hop is not simply a physical interface. It is represented by a composite forwarding object that has a primary local path and a backup peer path. This is called an ELP-NH and VPLAG object.

VPLAG is an internal forwarding abstraction for an EVPN multihomed ESI path in the HPE Networking Junos operating system. It represents the CE-facing multihomed LAG and can include both the local access path and the ELP backup path to the peer.

The VPLAG/ELP next-hop is an internal forwarding object. In Junos operational CLI, it is verified indirectly by checking the ESI state, the EVPN database, the ARP/route resolution, and the ELP tunnel endpoint programming. The command `show ethernet-switching vxlan-tunnel-end-point remote summary` confirms that an ELP tunnel exists between the multihoming peers.

```
ARP/ND entry for Host-2
|
v
VPLAG / ELP next-hop
|
+-- Primary path: local access interface toward CE/Host-2
|
+-- Backup path: ELP tunnel to the other MH peer
```

```
root@Leaf-2b> show ethernet-switching vxlan-tunnel-end-point remote summary
Logical System Name      Id  SVTEP-IP      IFL  L3-Idx      SVTEP-Mode      ELP-SVTEP-IP
<default>                0   3.1.1.200     lo0.0  0           3.1.1.3
RVTEP-IP      IFL-Idx      Interface      NH-Id  RVTEP-Mode  ELP-IP      Flags
3.1.1.2        832          vtep.32770     1775   RNVE        ELP
RVTEP-IP      L2-RTT              IFL-Idx  Interface      NH-Id  RVTEP-Mode  ELP-IP
Flags
3.1.1.2        VRF1_MAC_VRF_AWARE      671088642 vtep-5.32770 1775   RNVE        ELP
```

Component	Role
RPD	Receives and advertises EVPN control-plane routes.
L2ALD	Builds L2/EVPN forwarding state and creates composite ELP next-hop information.
PFE	Programs the forwarding object in hardware and performs fast local repair.

The important operational detail is that the backup path is already programmed. When the link fails, the PFE can repair traffic by using the backup ELP path without waiting for slow remote route convergence.

Link failure example

Before failure:

Host-1 -> EOS -> Anycast VTEP -> Leaf-2a -> Host-2

Leaf-2a access link to Host-2 fails:

Host-1 -> EOS -> Anycast VTEP -> Leaf-2a -> ELP tunnel -> Leaf-2b -> Host-2

This is called local repair. The remote EOS leaf may still send traffic to the anycast VTEP, and the underlay may still deliver some packets to Leaf-2a. Leaf-2a is still alive, so it repairs traffic locally by forwarding to Leaf-2b over the ELP path.

Node failure example

Leaf-2a node fails completely:

Underlay route to anycast VTEP through Leaf-2a is removed.

Spines retain only the path to 3.1.1.200 through Leaf-2b.

EOS traffic goes directly to Leaf-2b.

ELP is not used when the node is gone, because the failed node cannot repair anything. Node failure is handled by underlay ECMP convergence for the anycast VTEP route.

Summary

- ELP protects the case where the egress leaf is alive, but its local CE-facing link is down.
- The selected leaf repairs traffic by sending it to its multihoming peer.
- The repair is fast because the backup path is pre-programmed in hardware.
- This is different from node failure, where the underlay removes the failed path to the anycast VTEP.

Global Repair

Why global repair is needed

Local repair is fast, but it can create a sub-optimal traffic path. If traffic lands on Leaf-2a and Leaf-2a forwards it to Leaf-2b using ELP, traffic still works, but it hairpins through Leaf-2a. Global repair is the follow-up step that moves the remote ingress node to send traffic directly to the working leaf.

Local repair path after Leaf-2a access link failure:

EOS -> anycast VTEP -> Leaf-2a -> ELP tunnel -> Leaf-2b -> Host-2

Global repair optimized path:

EOS -> Leaf-2b unique VTEP -> Host-2

Local repair versus global repair

Repair type	Who repairs?	Speed	Result
Local repair / ELP	The egress leaf that received the traffic	Fast, local hardware repair	Traffic is redirected to the peer but may take a sub-optimal path.

Global repair	The remote ingress node	Control-plane driven after AD route update	Remote node sends directly to the surviving/working leaf.
---------------	-------------------------	--	---

How the remote node decides where to send traffic

The remote node needs to know both the anycast VTEP and the unique VTEPs of the MH peers. In normal state, when both peers are healthy, the remote node uses the anycast VTEP. If one peer withdraws its Ethernet A-D per ES route, the remote node can switch to the unique VTEP of the remaining peer.

AD per-ESI state seen by ingress node	Forwarding behavior
Two Ethernet A-D per ES routes present: Leaf-2a and Leaf-2b healthy	Use anycast VTEP 3.1.1.200.
Only Leaf-2b Ethernet A-D per ES route present	Use Leaf-2b unique VTEP 3.1.1.3.
Only Leaf-2a Ethernet A-D per ES route present	Use Leaf-2a unique VTEP 3.1.1.2.

```
Leaf-1a#show vxlan vtep

Remote VTEPS for Vxlan1:

VTEP           Tunnel Type(s)
-----
3.1.1.2         unicast
3.1.1.3         unicast
3.1.1.200       unicast

Total number of remote VTEPS:  3
```

Step-by-step failure flow

1. Steady state: EOS sends Host-1 to Host-2 traffic to the Junos anycast VTEP.
2. Underlay ECMP may deliver the packet to Leaf-2a.
3. Leaf-2a access link to Host-2 fails.
4. Leaf-2a immediately uses ELP local repair and redirects traffic to Leaf-2b.
5. Leaf-2a withdraws its Ethernet A-D per ES route.
6. Remote ingress node receives the withdraw and understands Leaf-2a is no longer a valid egress for that ESI.
7. Remote ingress node changes forwarding from anycast VTEP to Leaf-2b unique VTEP.
8. Traffic now goes directly to Leaf-2b and the hairpin is removed.
9. When the Leaf-2a access link recovers, Leaf-2a advertises AD per-ESI again and the remote node can return to anycast VTEP mode.

Summary

- Local repair is the immediate protection mechanism.

- Global repair is the optimization mechanism that avoids the hairpin path.
- AD per-ESI withdraw is the signal that tells remote nodes a peer is no longer valid for that Ethernet Segment.
- When the failed link recovers, the system can switch back to anycast VTEP mode.

Blocking Advertisement of Ethernet A-D per ES Route with an All-zero ESI

Problem

The ELP feature needs information to build peer-to-peer protection tunnels between multihoming leafs. In earlier behavior, an Ethernet A-D per ES route with an all-zero ESI could be advertised and received not only by the actual multihoming peer, but also by remote nodes. That can cause remote nodes to build ELP tunnels they will never use.

Useful ELP tunnels:

Leaf-2a <-> Leaf-2b

Unnecessary ELP tunnels:

EOS Leaf / remote PE -> Leaf-2a

EOS Leaf / remote PE -> Leaf-2b

Why this matters

ELP tunnel resources are limited. If every remote node builds unnecessary ELP tunnels to every MH peer, the fabric wastes resources. The remote node only needs normal VTEP forwarding and global repair behavior. It does not need an ELP tunnel to the Junos MH peer, because it is not part of that local Ethernet Segment.

New behavior with anycast multihoming

When anycast multihoming is configured, the Ethernet A-D per ES route with an all-zero ESI is not advertised in the same way. Instead, the ELP reroute address is carried in the AD per non-zero ESI route. A node creates an ELP tunnel only if it receives that AD per non-zero ESI and has the same ESI locally configured.

Node type	Has local ESI?	Should build ELP tunnel?
Leaf-2a receiving Leaf-2b AD per-ESI	Yes, same ESI on ae10	Yes. It needs peer protection.
Leaf-2b receiving Leaf-2a AD per-ESI	Yes, same ESI on ae10	Yes. It needs peer protection.
EOS remote ingress leaf receiving Junos AD route	No, it does not own that Junos ESI	No. It should not waste ELP tunnel resources.

Only actual multihoming peers should create ELP tunnels. Remote nodes should not create ELP tunnels for Ethernet Segments that are not locally configured.

Summary

- Ethernet A-D per ES route with an all-zero ESI behavior is optimized to avoid unnecessary ELP tunnel creation.
- The ELP reroute address is still advertised, but it is carried with the non-zero ESI information.
- A receiver should create ELP only if the same ESI exists locally.
- This is primarily a scale and resource optimization.

Anycast Multihoming for BUM

What is BUM?

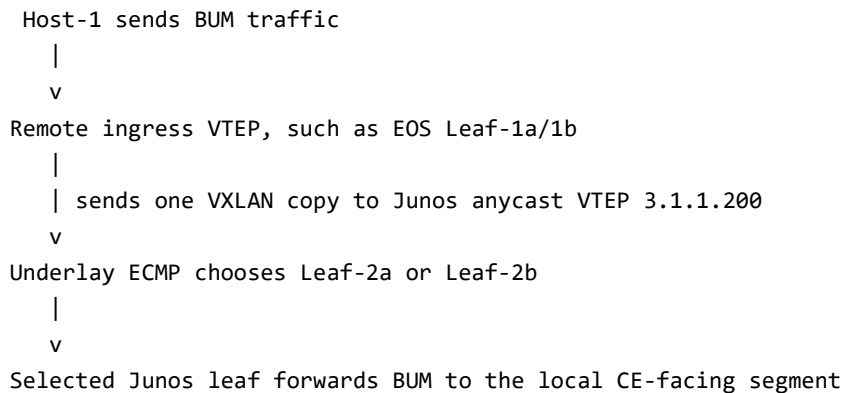
BUM means broadcast, unknown unicast, and multicast. In EVPN-VXLAN, BUM traffic is flooded using Type-3 IMET routes. Examples include ARP requests, IPv6 neighbor discovery, broadcast DHCP, unknown unicast traffic, and some multicast traffic.

What changes with anycast multihoming

With anycast multihoming, Leaf-2a and Leaf-2b advertise Type-3 IMET routes using the anycast VTEP. Even if the route distinguishers are different, the remote node treats the destination as one logical flood endpoint because the VTEP is the same.

Without anycast MH	With anycast MH
Remote node may see Leaf-2a and Leaf-2b as separate flood destinations.	Remote node sees one anycast VTEP flood destination.
Remote node may send more than one BUM copy depending on flood-list behavior.	Remote node sends only one BUM copy to the anycast VTEP.
Classic DF election prevents duplicate CE delivery.	Both nodes act as DF because only one BUM copy lands on one leaf through underlay ECMP.

Steady-state BUM flow



Why DF election is not required for the anycast BUM model

In classic EVPN multihoming, DF election prevents two multihomed PEs from forwarding duplicate BUM traffic to the same CE. In the anycast BUM model, the remote node sends only one BUM copy to the shared anycast VTEP, and the underlay delivers that single copy to only one physical leaf. Because the ingress side does not send one copy to each multihoming peer, classic DF election is not required for this anycast BUM forwarding behavior. Duplicate prevention is handled by the single-copy anycast flood behavior plus local-bias filtering for ELP-received BUM traffic.

Operationally, both peers are allowed to forward BUM when they are the leaf that receives the single anycast VXLAN copy. Type-4 ES-route based DF election is therefore not required for the steady-state anycast BUM behavior. The important operational check is not "which node is DF," but whether the remote node sends one copy to the anycast VTEP and whether local-bias behavior prevents duplicate forwarding over ELP.

ELP tunnel in the flood next-hop

For BUM protection, the ELP peer tunnel is added to the flood next-hop by default. This avoids building the flood group only after failure, which would be costly. However, if the ELP tunnel is already in the flood next-hop, the peer must avoid forwarding duplicate BUM traffic in steady state.

BUM flood next-hop on Leaf-2a:

- local access path to CE/Host-2
- ELP tunnel to Leaf-2b

BUM flood next-hop on Leaf-2b:

- local access path to CE/Host-2
- ELP tunnel to Leaf-2a

Local bias filter

The local bias filter prevents duplicate BUM delivery. In steady state, if Leaf-2b receives BUM traffic from Leaf-2a over the ELP tunnel, Leaf-2b drops it. This is correct because Leaf-2a should already have forwarded the BUM traffic to the CE locally.

Steady state:

- Leaf-2a receives BUM from anycast VTEP.
- Leaf-2a forwards to CE.
- Leaf-2a may also send to Leaf-2b over ELP flood path.
- Leaf-2b drops that ELP-received BUM due to local bias filter.

Link failure behavior

When Leaf-2a access link fails, Leaf-2a cannot forward BUM locally. It still may receive BUM traffic because the anycast VTEP is reachable through it. Leaf-2a sends the BUM to Leaf-2b through the ELP tunnel. Leaf-2b must now stop dropping that ELP-received BUM and forward it to the CE.

Leaf-2a access link fails:

- Leaf-2a withdraws AD per-ESI.
- Leaf-2b learns that peer access link is down.
- Leaf-2b removes local bias drop for Leaf-2a ELP traffic.
- Leaf-2b forwards BUM received from Leaf-2a over ELP to CE.

Why BUM convergence is slower than known-unicast ELP

Known-unicast traffic can often be repaired directly by a pre-programmed PFE object. BUM traffic needs a local bias filter change. That change is triggered by control-plane state, especially AD per-ESI withdraw from the peer. This means BUM convergence can be slower than known-unicast ELP convergence.

Scaling issue with many routing instances

If an ESI carries many routing instances or VLANs, an AD per-ESI withdraw may be processed per routing instance. RPD may send many IFL-specific updates to L2ALD, and L2ALD may send many local bias filter delete messages to PFE. PFE may wait for the last IFL-specific delete before deleting the port-level filter. This creates a gap between the first and last update and can cause BUM traffic loss.

Large scale example:

ESI carries 150 routing instances.
Peer access link fails.
RPD processes AD per-ESI withdraw in each RI.
L2ALD sends many IFL-specific local bias delete messages.
PFE waits for the last IFL update.
BUM traffic may be dropped during the delay.

Mass AD per-ESI withdraw optimization

To improve convergence, RPD sends a mass AD per-ESI withdraw to L2ALD. Instead of waiting for each routing instance, L2ALD sends a generic configuration message to PFE so that PFE deletes the local bias filter at the port/ESI level.

This optimization applies when port-based ESI granularity is configured. The CLI contains the words df-election-granularity, but in this anycast BUM discussion the goal is not to elect a single DF. The goal is to make local-bias and mass-withdraw handling operate at the ESI/port level instead of waiting on many per-IFL updates:

```
set interfaces <ae> esi df-election-granularity per-esi
```

Summary

- Type-3 IMET is advertised with the anycast VTEP so the remote node sends only one BUM copy.
- Because only one BUM copy is sent, both MH peers can act as DF by default.
- ELP is pre-added to the flood next-hop to avoid rebuilding flood groups after failure.
- Local bias filters prevent duplicate BUM forwarding in steady state.
- Upon peer access-link failure, the local bias behavior must change so BUM received over ELP can be forwarded.
- BUM convergence is more control-plane dependent than known-unicast ELP convergence.

Auto ESI Based on Type-1 LACP

Why auto-ESI is useful

In a large fabric, manually managing ESI values on every multihomed bundle is difficult. Each ESI must be unique per Ethernet Segment and must match on both multihoming peers. Auto-ESI reduces operational overhead by deriving the ESI automatically, for example from LACP information.

Manual ESI	Auto-ESI
Operator configures an explicit ESI on both peers.	Device derives the ESI using a supported method such as LACP.
Simple in a small lab.	Better at scale where many bundles exist.
Risk of duplicate or mismatched ESI values.	Reduces manual ESI management overhead.

Why force-up creates a special case

Force-up mode allows the AE link to come up logically before full LACP negotiation completes. That means the interface may be up and may learn a server MAC before the dynamically derived ESI is known. During this temporary period, the ESI is all zeroes.

Temporary force-up state:

- Link is logically up.
- LACP-derived ESI is not discovered yet.
- ESI = all zeroes.
- Leaf can learn the server MAC.
- MAC is treated like a single-homed MAC at first.

Problem in anycast mode before ESI is discovered

Anycast VTEP allows remote traffic to land on either MH peer. But before the ESI is discovered, only one leaf may have learned the server MAC. If traffic lands on the other peer, the peer needs a way to redirect traffic to the actual owner leaf.

- Leaf-2a force-up link comes up first.
- Leaf-2a learns Host-2 MAC.
- ESI is still all zeroes.
- Leaf-2a advertises Type-2 route.
- Remote EOS sends traffic to anycast VTEP.
- Underlay ECMP may land packet on Leaf-2b.
- Leaf-2b must redirect traffic to Leaf-2a using ELP.

Why Type-2 carries ELP address even with all-zero ESI

The Type-2 route must carry the ELP/reroute address in the BGP tunnel encapsulation attribute even when the ESI is all zeroes. This gives the peer enough information to redirect traffic to the leaf that owns the MAC.

Receiver	How it interprets the route
Peer MH leaf	Sees that the anycast IP matches its own anycast IP, so it programs the MAC toward the ELP/reroute address.
Remote PE/EOS leaf	Does not have the anycast IP locally, so it programs the MAC toward the anycast VTEP.

What happens after LACP discovers the ESI

Once LACP PDUs are received and the non-zero ESI is discovered, the leaves advertise Type-1 AD per-ES routes. At that point, the MAC can move from temporary ELP programming to the proper ESI-aware ELPNH.

Before ESI discovery on peer:

MAC -> ELP address

After non-zero ESI is discovered:

MAC -> ELPNH { esi-ifd-id, elp-rvtep1, elp-rvtep2, ... }

Why this matters during reboot

This special handling is important during reboot scenarios. One leaf may come back and receive Type-2 routes before its local ESI state is fully rebuilt. The device needs to temporarily program traffic toward the ELP address and later move it to the proper ELPNH once the ESI becomes known. Without this transition, traffic could be blackholed or require unnecessary relearning.

Summary

- Auto-ESI removes the need to manually manage ESI values across many bundles.
- Force-up can bring an interface up before the ESI is discovered.
- During that period, the MAC is advertised with all-zero ESI but still carries ELP information.
- Remote nodes use anycast VTEP; MH peers use the ELP address to redirect traffic if it lands on the wrong peer.
- Once the non-zero ESI is discovered, Type-1 AD routes are advertised and forwarding moves to the normal ESI-aware ELPNH.

Configurations

Configuration assumptions

Leaf-2a and Leaf-2b are Junos ESI all-active peers. EOS Leaf-1a and Leaf-1b are the remote Arista MLAG peers. The anycast configuration must be identical on all Junos multihoming peers where required, especially the anycast VTEP address and router MAC.

2-IP model used for this lab

The 2-IP model uses one unique/reroute address and one shared anycast address. In the lab, Leaf-2a uses unique VTEP/reroute address 3.1.1.2, Leaf-2b uses unique VTEP/reroute address 3.1.1.3, and both advertise the shared anycast VTEP 3.1.1.200.

Device	Unique/reroute VTEP	Anycast VTEP	Router MAC
Leaf-2a	3.1.1.2	3.1.1.200	00:02:00:00:00:11
Leaf-2b	3.1.1.3	3.1.1.200	00:02:00:00:00:11

Leaf-2a anycast multihoming configuration

```
## Loopback addresses
set interfaces lo0 unit 0 family inet address 3.1.1.2/32 primary
set interfaces lo0 unit 0 family inet address 3.1.1.200/32

## ELP/reroute address. This must be the unique local VTEP, not the anycast VTEP.
set forwarding-options evpn-vxlan reroute-address inet 3.1.1.2
set forwarding-options evpn-vxlan shared-tunnels

## Global anycast multihoming
set protocols evpn anycast-multihoming
set protocols evpn anycast-multihoming mlag-interop-mode
set protocols evpn anycast-multihoming inet address 3.1.1.200
set protocols evpn anycast-multihoming router-mac 00:02:00:00:00:11

## ESI interface behavior
set interfaces ae10 esi anycast
```

```
set interfaces ae10 esi df-election-granularity per-esi
```

Leaf-2b anycast multihoming configuration

```
## Loopback addresses
set interfaces lo0 unit 0 family inet address 3.1.1.3/32 primary
set interfaces lo0 unit 0 family inet address 3.1.1.200/32

## ELP/reroute address. This must be the unique local VTEP, not the anycast VTEP.
set forwarding-options evpn-vxlan reroute-address inet 3.1.1.3
set forwarding-options evpn-vxlan shared-tunnels

## Global anycast multihoming
set protocols evpn anycast-multihoming
set protocols evpn anycast-multihoming mlag-interop-mode
set protocols evpn anycast-multihoming inet address 3.1.1.200
set protocols evpn anycast-multihoming router-mac 00:02:00:00:00:11

## ESI interface behavior
set interfaces ae10 esi anycast
set interfaces ae10 esi df-election-granularity per-esi
```

Anycast-supported ae10.0 trunk attachment for VLAN 32

Apply on both Leaf-2a and Leaf-2b:

```
set interfaces ae10 mtu 9192
set interfaces ae10 network-isolation-profile g1
set interfaces ae10 esi 00:00:00:00:00:00:00:00:32:32
set interfaces ae10 esi all-active
set interfaces ae10 esi df-election-granularity per-esi
set interfaces ae10 esi anycast
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae10 aggregated-ether-options lacp system-id 00:00:00:32:32:32
set interfaces ae10 unit 0 family ethernet-switching interface-mode trunk
set interfaces ae10 unit 0 family ethernet-switching vlan members VLAN32
set routing-instances VRF1_MAC_VRF_AWARE interface ae10.0
set routing-instances VRF1_MAC_VRF_AWARE vlans VLAN32 interface ae10.0
set routing-instances VRF1_MAC_VRF_AWARE vlans VLAN32 vlan-id 32
set routing-instances VRF1_MAC_VRF_AWARE vlans VLAN32 vxlan vni 10032
set routing-instances VRF1_MAC_VRF_AWARE vlans VLAN32 l3-interface irb.32
```

```
## Physical member hold-time used in the lab
## Leaf-2a physical AE member hold-time
set interfaces et-0/0/2 hold-time up 30000 down 0
## Leaf-2b physical AE member hold-time
set interfaces et-0/0/8 hold-time up 30000 down 0
```

Why the reroute-address is unique per leaf

The reroute address is used for ELP redirection between multihoming peers. It must identify a specific physical leaf. Therefore, Leaf-2a uses 3.1.1.2 and Leaf-2b uses 3.1.1.3. The anycast VTEP 3.1.1.200 is shared and should not be used as the reroute address.

Address type	Shared or unique?	Used by
Anycast VTEP 3.1.1.200	Shared by Leaf-2a and Leaf-2b	Remote EOS side sends traffic to this. Underlay ECMP chooses a leaf.
Leaf-2a reroute 3.1.1.2	Unique to Leaf-2a	Leaf-2b can redirect traffic to Leaf-2a using ELP.
Leaf-2b reroute 3.1.1.3	Unique to Leaf-2b	Leaf-2a can redirect traffic to Leaf-2b using ELP.

MH peer import policy for interop mode

In interop mode, EVPN routes can arrive at an MH peer with the protocol next-hop equal to the local anycast IP. Without a policy, the leaf may not import the peer route correctly because it appears to have its own next-hop. The import policy rewrites the next-hop to the peer unique/reroute address and adds NO_ADVERTISE so the route is not re-advertised unnecessarily.

Leaf-2a import policy

```
set policy-options community NO_ADVERTISE members no-advertise

set policy-options policy-statement mh_peer_import term 1 from family evpn
set policy-options policy-statement mh_peer_import term 1 from next-hop 3.1.1.200
set policy-options policy-statement mh_peer_import term 1 then community add NO_ADVERTISE
set policy-options policy-statement mh_peer_import term 1 then next-hop 3.1.1.3
set policy-options policy-statement mh_peer_import term 1 then accept
set policy-options policy-statement mh_peer_import term 2 then accept

set protocols bgp group OVERLAY import mh_peer_import
```

Leaf-2b import policy

```
set policy-options community NO_ADVERTISE members no-advertise

set policy-options policy-statement mh_peer_import term 1 from family evpn
set policy-options policy-statement mh_peer_import term 1 from next-hop 3.1.1.200
set policy-options policy-statement mh_peer_import term 1 then community add NO_ADVERTISE
set policy-options policy-statement mh_peer_import term 1 then next-hop 3.1.1.2
set policy-options policy-statement mh_peer_import term 1 then accept
set policy-options policy-statement mh_peer_import term 2 then accept

set protocols bgp group OVERLAY import mh_peer_import
```

EVPN route priority policy

The route priority policy gives important EVPN route types higher BGP output queue priority. The intent is to improve convergence behavior by prioritizing certain EVPN route advertisements, such as Type-1 AD routes.

```

set policy-options policy-statement evpn-rt-priority-policy term 1 from family evpn
set policy-options policy-statement evpn-rt-priority-policy term 1 from nlri-route-type 1
set policy-options policy-statement evpn-rt-priority-policy term 1 then bgp-output-queue-priority priority 16

set policy-options policy-statement evpn-rt-priority-policy term 2 from family evpn
set policy-options policy-statement evpn-rt-priority-policy term 2 from nlri-route-type 2
set policy-options policy-statement evpn-rt-priority-policy term 2 then bgp-output-queue-priority priority 1

set policy-options policy-statement evpn-rt-priority-policy term 3 from family evpn
set policy-options policy-statement evpn-rt-priority-policy term 3 from nlri-route-type 3
set policy-options policy-statement evpn-rt-priority-policy term 3 then bgp-output-queue-priority priority 2

set policy-options policy-statement evpn-rt-priority-policy term 4 from family evpn
set policy-options policy-statement evpn-rt-priority-policy term 4 from nlri-route-type 4
set policy-options policy-statement evpn-rt-priority-policy term 4 then bgp-output-queue-priority priority 16

set protocols bgp group OVERLAY export evpn-rt-priority-policy

```

Underlay advertisement of the anycast VTEP

The anycast VTEP must be reachable in the underlay. In the lab, the underlay export policy already exports 3.1.1.0/24 or longer, so 3.1.1.200/32 should be advertised if it is configured on lo0.0. The spines should see two paths to 3.1.1.200: one through Leaf-2a and one through Leaf-2b.

Expected check on spine:

```

root@Spine-1-qfx5120> show route 3.1.1.200
inet.0: 36 destinations, 40 routes (35 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
3.1.1.200/32      *[BGP/170] 6d 05:49:33, localpref 100
                  AS path: 65003 I, validation-state: unverified
                   to 171.0.1.5 via et-0/0/50.0
                   > to 171.0.1.7 via et-0/0/51.0
                  [BGP/170] 6d 05:49:33, localpref 100
                   AS path: 65002 I, validation-state: unverified
                   > to 171.0.1.5 via et-0/0/50.0

```

```

root@Spine-1-qfx5120> show lldp neighbors

```

Local Interface	Parent Interface	Chassis Id	Port info	System Name
et-0/0/51	-	18:2a:d3:57:a2:80	et-0/0/14	Leaf-
2b.englab.juniper.net				
et-0/0/50	-	9c:8a:cb:05:55:00	et-0/0/1	Leaf-
2a.juniper.net				

Fast-convergence and core-isolation settings

The lab uses BFD, LACP fast, physical member hold-time, l2-learning stability knobs, and native network-isolation. The intended recovery order is core links and BGP recover first, ae10 is allowed back after 50 seconds, and anycast advertisement is restored after the event-script timer of 100 seconds.

BFD used on Leaf-2a, Leaf-2b, Spine-1, and Spine-2

```
set protocols bgp group UNDERLAY bfd-liveness-detection minimum-interval 100
set protocols bgp group UNDERLAY bfd-liveness-detection multiplier 3
set protocols bgp group UNDERLAY bfd-liveness-detection session-mode automatic
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 bfd-liveness-detection session-mode automatic
```

Native network-isolation action on both Leaf-2a and Leaf-2b

```
set interfaces ae10 network-isolation-profile g1
set protocols network-isolation group g1 detection hold-time up 50000
set protocols network-isolation group g1 detection service-tracking core-isolation
set protocols network-isolation group g1 service-tracking-action link-down
```

L2 learning and access-side stability knobs

```
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces et-0/0/2 hold-time up 30000 down 0    ## Leaf-2a
set interfaces et-0/0/8 hold-time up 30000 down 0    ## Leaf-2b
set protocols l2-learning global-mac-table-aging-time 1200
set protocols l2-learning platform-parameters no-mac-flush-on-aa-ae-down
```

Optional auto-ESI configuration placeholder

The lab uses a manually configured ESI on ae10. For the later auto-ESI Type-1 LACP test, the exact CLI syntax should be verified on your Junos build. Conceptually, you would replace the manual ESI value with the auto-derive LACP method and keep all-active, anycast, and per-ESI DF granularity.

Conceptual only. Verify exact syntax on the target release.

```
delete interfaces ae10 esi 00:00:00:00:00:00:00:00:32:32
set interfaces ae10 esi auto-derive lacp
set interfaces ae10 esi all-active
set interfaces ae10 esi anycast
set interfaces ae10 esi df-election-granularity per-esi
```

Event Script for EVPN Core Isolation

Purpose

This event script is used only for the core-isolation test case. It is not required for normal access-link, ae interface, single-uplink, or full-node failover tests.

The goal is simple:

- Normal state: advertise the anycast VTEP to the underlay.
- Core-isolated state: stop advertising the anycast VTEP so the isolated leaf does not attract traffic.
- Recovery state: after a timer, advertise the anycast VTEP again.

Why this is needed

In an anycast VTEP design, more than one leaf can advertise the same anycast VTEP address. If one leaf becomes isolated from the core but keeps advertising the anycast VTEP, remote devices may continue sending traffic to that leaf and traffic can blackhole.

This script changes the active apply-group when core isolation is detected or cleared.

High-level behavior

Timer behavior used in this lab:

- Core isolation detected: the event script runs with timer 0 and changes apply-groups from adv_anycast to reject_anycast.
- Core isolation detected: native network-isolation brings ae10 down using service-tracking-action link-down.
- Core isolation cleared: network-isolation waits 50000 ms, so ae10 comes back after 50 seconds.
- Core isolation cleared: the event script waits 100 seconds, then changes apply-groups from reject_anycast back to adv_anycast.

This order intentionally brings ae10 back before advertising 3.1.1.200/32 again.

State	Active apply-group	Behavior
Normal	adv_anycast	Anycast VTEP is advertised to the underlay.
Core isolated	reject_anycast	Anycast VTEP is rejected/suppressed from underlay export.
Core isolation cleared	adv_anycast	After the timer expires, anycast VTEP is advertised again.

Install the SLAX script

Copy the SLAX script to the Junos event script directory on each anycast multihoming leaf:

```
/var/db/scripts/event/evpn-core-isolation-recovery.slax
```

Verify the script file is present:

```
file list /var/db/scripts/event/  
file show /var/db/scripts/event/evpn-core-isolation-recovery.slax
```

Because the script imports junos.xml, also verify the import directory exists:

```
file list /var/db/scripts/import/
```

Junos configuration

Use the following configuration on each Junos anycast multihoming leaf. This example keeps the existing UNDERLAY_EXPORT policy unchanged and uses apply-groups to switch between advertise and reject behavior.

```
set policy-options policy-statement ADV-ANYCAST-VTEP term anycast from route-filter 3.1.1.200/32 exact
set policy-options policy-statement ADV-ANYCAST-VTEP term anycast then accept
set policy-options policy-statement ADV-ANYCAST-VTEP term default then next policy

set policy-options policy-statement REJECT-ANYCAST-VTEP term anycast from route-filter 3.1.1.200/32 exact
set policy-options policy-statement REJECT-ANYCAST-VTEP term anycast then reject
set policy-options policy-statement REJECT-ANYCAST-VTEP term default then next policy

set groups adv_anycast protocols bgp group UNDERLAY export [ ADV-ANYCAST-VTEP UNDERLAY_EXPORT ]
set groups reject_anycast protocols bgp group UNDERLAY export [ REJECT-ANYCAST-VTEP UNDERLAY_EXPORT ]

set apply-groups adv_anycast

set event-options policy policy_core_isolated events evpn_core_isolated
set event-options policy policy_core_isolated then event-script evpn-core-isolation-recovery.slax arguments timer 0
set event-options policy policy_core_isolated then event-script evpn-core-isolation-recovery.slax arguments del-group adv_anycast
set event-options policy policy_core_isolated then event-script evpn-core-isolation-recovery.slax arguments add-group reject_anycast

set event-options policy policy_core_isolation_cleared events evpn_core_isolation_cleared
set event-options policy policy_core_isolation_cleared then event-script evpn-core-isolation-recovery.slax arguments timer 100
set event-options policy policy_core_isolation_cleared then event-script evpn-core-isolation-recovery.slax arguments del-group reject_anycast
set event-options policy policy_core_isolation_cleared then event-script evpn-core-isolation-recovery.slax arguments add-group adv_anycast

set event-options event-script file evpn-core-isolation-recovery.slax

## Native core-isolation interface action used in the lab
set interfaces ae10 network-isolation-profile g1
set protocols network-isolation group g1 detection hold-time up 50000
set protocols network-isolation group g1 detection service-tracking core-isolation
set protocols network-isolation group g1 service-tracking-action link-down

## Fast convergence and access-side stability knobs
set protocols bgp group UNDERLAY bfd-liveness-detection minimum-interval 100
set protocols bgp group UNDERLAY bfd-liveness-detection multiplier 3
set protocols bgp group UNDERLAY bfd-liveness-detection session-mode automatic
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 bfd-liveness-detection session-mode automatic
set interfaces ae10 aggregated-ether-options lacp periodic fast
set protocols l2-learning global-mac-table-aging-time 1200
set protocols l2-learning platform-parameters no-mac-flush-on-aa-ae-down
```

Evpn-core-isolation-recovery.slax

```
root@Leaf-2b:RE:0% cd /var/db/scripts/event/
root@Leaf-2b:RE:0% ls -l
total 0
lrwxr-xr-x 1 root wheel 90 Apr 15 06:24 evpn-core-isolation-recovery.slax -> /packages/mnt/junos-routing-
scripts/var/db/scripts/event/evpn-core-isolation-recovery.slax

root@Leaf-2b:RE:0% cat evpn-core-isolation-recovery.slax
/*
 * Event script for EVPN core isolation recovery - Simple & Reliable Version
 * This script handles the evpn_core_isolation_cleared event
 * Uses direct RPC calls for maximum reliability
 */

version 1.1;

ns junos = "http://xml.juniper.net/junos/*/junos";
ns xnm = "http://xml.juniper.net/xnm/1.1/xnm";
ns jcs = "http://xml.juniper.net/junos/commit-scripts/1.0";

import "../import/junos.xsl";

param $timer = "";
param $del-group = "";
param $add-group = "";

match / {
  mvar $timer-delay = 60;
  /* Validate timer argument non-numeric (NaN), and non-positive values. */
  if ($timer) {
    var $n = number($timer);
    /* XPath 1.0 trick: NaN is not equal to itself */
    if (not($n = $n)) {
      expr jcs:syslog("warning", "timer argument is not numeric: " _ $timer _ " - using default " _ $timer-delay _
"s");
      set $timer-delay = 60;
    } else if ($n < 0) {
      expr jcs:syslog("warning", "timer argument not positive: " _ $n _ " - using default " _ $timer-delay _ "s");
      set $timer-delay = 60;
    } else {
      set $timer-delay = $n;
      expr jcs:syslog("info", "Using timer-delay from argument: " _ $timer-delay _ "s");
    }
  } else {
    expr jcs:syslog("info", "No timer argument provided, using default: " _ $timer-delay _ "s");
  }

  /* Prepare variables for group names (no defaults - these are required) */
  mvar $del-group-name = "";
  mvar $add-group-name = "";

  if ($del-group) {
    set $del-group-name = $del-group;
    expr jcs:syslog("info", "Using del-group argument: " _ $del-group-name);
  } else {
    expr jcs:syslog("warning", "No del-group argument provided");
  }

  if ($add-group) {
    set $add-group-name = $add-group;
    expr jcs:syslog("info", "Using add-group argument: " _ $add-group-name);
  } else {
    expr jcs:syslog("warning", "No add-group argument provided");
  }

  /* If either required argument is missing, intentionally produce a commit error
  so callers/operators can detect the mis-invocation via logs/alarms. */
  if (not($del-group) or not($add-group)) {
    <event-script-results> {
      <output> "ERROR: Missing required argument(s): del-group and/or add-group. Forcing commit error to signal
failure.";
    }
  }
}
```

```

var $rpc-connection = jcs:open();
if ($rpc-connection) {
    /* Deliberately issue an invalid commit RPC to generate an xnm:error */
    var $bad-commit = <commit-configuration> {
        <_invalid> "missing-required-args";
    };

    var $bad-response = jcs:execute($rpc-connection, $bad-commit);
    if ($bad-response//xnm:error) {
        <output> "Intentional commit error produced as expected: " _ $bad-response//xnm:error/message;
    } else {
        <output> "WARNING: Intentional commit did not return an error (unexpected)";
    }

    expr jcs:close($rpc-connection);
} else {
    <output> "ERROR: Could not open RPC connection to force commit error";
}
}
} else {
    <event-script-results> {
        /* Log the event reception */
        <output> "EVPN core isolation cleared event received - starting recovery process";

        /* Start timer countdown */
        <output> "Starting " _ $timer-delay _ " second timer before applying configuration";

        expr jcs:sleep($timer-delay);

        <output> "Timer completed. Applying configuration changes now.";

        /* Use direct RPC approach */
        var $rpc-connection = jcs:open();

        if ($rpc-connection) {
            <output> "Connection established. Applying configuration...";

            /* Create the configuration change using arguments */
            var $config-change = <load-configuration action="merge"> {
                <configuration> {
                    /* delete the named group */
                    <apply-groups delete="delete"> $del-group-name;
                    /* add the named group */
                    <apply-groups> $add-group-name;
                }
            };

            /* Execute the load-configuration RPC */
            var $load-response = jcs:execute($rpc-connection, $config-change);

            if ($load-response//xnm:error) {
                <output> "Load configuration failed: " _ $load-response//xnm:error/message;
            } else {
                <output> "Configuration loaded successfully";

                /* Commit the configuration */
                var $commit-rpc = <commit-configuration> {
                    <log> "Event script: EVPN core isolation recovery - restored anycast after " _ $timer-delay _ "s
delay; del-group=" _ $del-group-name _ ", add-group=" _ $add-group-name;
                };

                var $commit-response = jcs:execute($rpc-connection, $commit-rpc);

                if ($commit-response//xnm:error) {
                    <output> "Commit failed: " _ $commit-response//xnm:error/message;
                } else {
                    <output> "SUCCESS: Configuration committed - " _ $del-group-name _ " deleted, " _ $add-group-name
_ " set";
                }
            }

            expr jcs:close($rpc-connection);
        } else {

```

```

    <output> "ERROR: Could not establish connection to commit configuration";
  }

  <output> "EVPN core isolation recovery script completed";
}
}
}

```

Verification and Troubleshooting

Verification

When the feature is newly enabled, do not start troubleshooting with only end-to-end ping. First prove that the underlay, EVPN control plane, route import, VTEP resolution, and local host adjacency are all correct.

Underlay checks

On EOS Leaf-1a/Leaf-1b:

```

Leaf-1a# show ip route 3.1.1.200
VRF: default
  B E      3.1.1.200/32 [20/0] via 171.0.1.0, Ethernet51/1
                        via 171.0.1.10, Ethernet56/1

```

On Junos spine:

```

root@Spine-1-qfx5120> show route 3.1.1.200
inet.0: 36 destinations, 40 routes (35 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both
3.1.1.200/32      *[BGP/170] 6d 05:54:10, localpref 100
                  AS path: 65003 I, validation-state: unverified
                  > to 171.0.1.5 via et-0/0/50.0
                  > to 171.0.1.7 via et-0/0/51.0
                  [BGP/170] 6d 05:54:10, localpref 100
                  AS path: 65002 I, validation-state: unverified
                  > to 171.0.1.5 via et-0/0/50.0

```

On Junos Leaf-2a/Leaf-2b:

```

root@Leaf-2a> show interfaces lo0.0 terse
Interface           Admin Link Proto  Local                Remote
lo0.0                up    up    inet    3.1.1.2              --> 0/0
                    3.1.1.200            --> 0/0

root@Leaf-2a> show route 3.1.1.200
inet.0: 14 destinations, 19 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
3.1.1.200/32      *[Direct/0] 6d 05:55:57
                  > via lo0.0
                  [BGP/170] 6d 05:54:32, localpref 100
                  AS path: 65022 65003 I, validation-state: unverified

```

```

> to 171.0.1.4 via et-0/0/1.0
[BGP/170] 6d 05:54:25, localpref 100
AS path: 65022 65003 I, validation-state: unverified
> to 171.0.1.14 via et-0/0/0.0

```

:vxlan.inet.0: 9 destinations, 9 routes (9 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

```

3.1.1.200/32      *[Direct/0] 6d 05:55:53
                  > via lo0.0

```

EVPN control-plane checks

On EOS:

```
Leaf-1a#show bgp evpn summary
```

BGP summary information for VRF default

Router identifier 151.0.0.1, local AS number 65001

Neighbor Status Codes: m - Under maintenance

Description	Neighbor	V AS	MsgRcvd	MsgSent	InQ	OutQ	Up/Down	State	PfxRcd	PfxAcc
Spine-1	3.1.1.101	4 65022	70611	76063	0	0	18d18h	Estab	26	26
Spine-2	3.1.1.102	4 65022	70586	76152	0	0	18d18h	Estab	26	26

Leaf-1a# show bgp evpn route-type imet rd 3.1.1.2:32 detail

BGP routing table information for VRF default

Router identifier 151.0.0.1, local AS number 65001

BGP routing table entry for imet 10032 3.1.1.200, Route Distinguisher: 3.1.1.2:32

Paths: 2 available

65022 65002

3.1.1.200 from 3.1.1.101 (3.1.1.101)

Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP contributor

Extended Community: Route-Target-AS:32:32 TunnelEncap:tunnelTypeVxlan

VNI: 10032, Tunnel encapsulation: VXLAN, 3.1.1.2

PMSI Tunnel: Ingress Replication, MPLS Label: 10032, Leaf Information Required: false, Tunnel ID: 3.1.1.200

65022 65002

3.1.1.200 from 3.1.1.102 (3.1.1.102)

Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor

Extended Community: Route-Target-AS:32:32 TunnelEncap:tunnelTypeVxlan

VNI: 10032, Tunnel encapsulation: VXLAN, 3.1.1.2

PMSI Tunnel: Ingress Replication, MPLS Label: 10032, Leaf Information Required: false, Tunnel ID: 3.1.1.200

Leaf-1a# show bgp evpn route-type mac-ip 10:0e:7e:b3:ef:80 detail

BGP routing table information for VRF default

Router identifier 151.0.0.1, local AS number 65001

BGP routing table entry for mac-ip 10032 100e.7eb3.ef80, Route Distinguisher: 3.1.1.2:32

Paths: 2 available

65022 65002

3.1.1.200 from 3.1.1.102 (3.1.1.102)

Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP contributor

Extended Community: Route-Target-AS:32:32 TunnelEncap:tunnelTypeVxlan

VNI: 10032 ESI: 0000:0000:0000:0000:3232, Tunnel encapsulation: VXLAN, 3.1.1.2

65022 65002

3.1.1.200 from 3.1.1.101 (3.1.1.101)

```

Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
Extended Community: Route-Target-AS:32:32 TunnelEncap:tunnelTypeVxlan
VNI: 10032 ESI: 0000:0000:0000:0000:3232, Tunnel encapsulation: VXLAN, 3.1.1.2
BGP routing table entry for mac-ip 10032 100e.7eb3.ef80 32.1.1.10, Route Distinguisher: 3.1.1.2:32
Paths: 2 available
65022 65002
3.1.1.200 from 3.1.1.102 (3.1.1.102)
Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP contributor
Extended Community: Route-Target-AS:3:3 Route-Target-AS:32:32 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:00:02:00:00:00:11
VNI: 10032 L3 VNI: 3333 ESI: 0000:0000:0000:0000:3232, Tunnel encapsulation: VXLAN, 3.1.1.2
65022 65002
3.1.1.200 from 3.1.1.101 (3.1.1.101)
Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
Extended Community: Route-Target-AS:3:3 Route-Target-AS:32:32 TunnelEncap:tunnelTypeVxlan
EvpnRouterMac:00:02:00:00:00:11
VNI: 10032 L3 VNI: 3333 ESI: 0000:0000:0000:0000:3232, Tunnel encapsulation: VXLAN, 3.1.1.2

```

```

Leaf-1a#show bgp evpn route-type ip-prefix ipv4 rd 3.1.1.2:3 detail
BGP routing table information for VRF default
Router identifier 151.0.0.1, local AS number 65001
BGP routing table entry for ip-prefix 32.1.1.0/24, Route Distinguisher: 3.1.1.2:3
Paths: 2 available
65022 65002
3.1.1.2 from 3.1.1.101 (3.1.1.101)
Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP contributor
Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:9c:8a:cb:05:55:00
VNI: 3333
65022 65002
3.1.1.2 from 3.1.1.102 (3.1.1.102)
Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:9c:8a:cb:05:55:00
VNI: 3333
BGP routing table entry for ip-prefix 32.1.1.1/32, Route Distinguisher: 3.1.1.2:3
Paths: 2 available
65022 65002
3.1.1.2 from 3.1.1.101 (3.1.1.101)
Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP head, ECMP, best, ECMP contributor
Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:9c:8a:cb:05:55:00
VNI: 3333
65022 65002
3.1.1.2 from 3.1.1.102 (3.1.1.102)
Origin IGP, metric -, localpref 100, weight 0, tag 0, valid, external, ECMP, ECMP contributor
Extended Community: Route-Target-AS:3:3 TunnelEncap:tunnelTypeVxlan EvpnRouterMac:9c:8a:cb:05:55:00
VNI: 3333

```

Leaf-1a#show vxlan vtep

Remote VTEPS for Vxlan1:

VTEP	Tunnel Type(s)
3.1.1.2	unicast
3.1.1.3	unicast
3.1.1.200	unicast

Total number of remote VTEPS: 3

```
Leaf-1a#show ip route vrf VRF1_IP_VRF 32.1.1.10
VRF: VRF1_IP_VRF
B E      32.1.1.10/32 [20/0] via VTEP 3.1.1.200 VNI 3333 router-mac 00:02:00:00:00:11
```

VRF and host reachability checks

```
On EOS:
Leaf-1a#show ip route vrf VRF1_IP_VRF 32.1.1.0/24

VRF: VRF1_IP_VRF
B E      32.1.1.0/24 [20/0] via VTEP 3.1.1.2 VNI 3333 router-mac 9c:8a:cb:05:55:00
                        via VTEP 3.1.1.3 VNI 3333 router-mac 18:2a:d3:57:a2:80
```

```
Leaf-1a# show ip route vrf VRF1_IP_VRF 32.1.1.10
VRF: VRF1_IP_VRF
B E      32.1.1.10/32 [20/0] via VTEP 3.1.1.200 VNI 3333 router-mac 00:02:00:00:00:11
```

```
Leaf-1a# show ip arp vrf VRF1_IP_VRF
Address      Age (sec)  Hardware Addr  Interface
31.1.1.10    1:38:29    001c.73d6.16b3  Vlan31, Port-Channel10
```

```
On Junos Leaf-2a/Leaf-2b:
root@Leaf-2a> show route table VRF1_IP_VRF.inet.0

VRF1_IP_VRF.inet.0: 9 destinations, 13 routes (9 active, 0 holddown, 0 hidden)
4.1.1.5/32      *[Direct/0] 6d 06:09:35
> via lo0.2
31.1.1.0/24     *[EVPN/170] 6d 06:08:08
> to 10.92.75.254 via em0.0
31.1.1.10/32    *[EVPN/7] 6d 05:11:18
> to 171.0.1.14 via et-0/0/0.0
> to 171.0.1.4 via et-0/0/1.0
32.1.1.0/24     *[Direct/0] 6d 06:08:01
> via irb.32
[EVPN/170] 6d 06:08:00
> to 171.0.1.14 via et-0/0/0.0
> to 171.0.1.4 via et-0/0/1.0
32.1.1.1/32     *[Local/0] 6d 06:08:01
Local via irb.32
[EVPN/170] 6d 06:08:01
> to 171.0.1.14 via et-0/0/0.0
> to 171.0.1.4 via et-0/0/1.0
32.1.1.10/32    *[EVPN/7] 00:08:40
> via irb.32
```

```
root@Leaf-2a> show arp no-resolve vpn VRF1_IP_VRF
MAC Address      Address      Interface      Flags
```

10:0e:7e:b3:ef:80 32.1.1.10 irb.32 [ae10.0] permanent remote

root@Leaf-2a> show ethernet-switching table instance VRF1_MAC_VRF_AWARE

Ethernet switching table : 1 entries, 1 learned

Routing instance : VRF1_MAC_VRF_AWARE

Ethernet switching table : 1 entries, 1 learned

Routing instance : VRF1_MAC_VRF_AWARE

Vlan	MAC	MAC	GBP	Logical	SVLBNH/	Active
name	address	flags	tag	interface	VENH Index	source
VLAN32	10:0e:7e:b3:ef:80	DLR		ae10.0		

root@Leaf-2a> show evpn database mac-address 10:0e:7e:b3:ef:80 extensive

Instance: VRF1_MAC_VRF_AWARE

VN Identifier: 10032, MAC address: 10:0e:7e:b3:ef:80

State: 0x0

Source: 00:00:00:00:00:00:00:32:32, Rank: 1, Status: Active

Local origin: ae10.0

Remote origin: 3.1.1.3

Remote state: <Mac-Only-Adv>

Remote route distinguisher: 3.1.1.3:32

Mobility sequence number: 0 (minimum origin address 3.1.1.2)

Timestamp: Jun 24 00:59:09.265041 (0x6a3b2bdd)

State: <Local-MAC-Only Local-To-Remote-Adv-Allowed Remote-To-Local-Adv-Done>

Last IPC-Event: add, Last IPC-Status: none, Last IPC-Timestamp: Jun 24 00:48:53.193 2026

MAC advertisement route status: Created

IP address: 32.1.1.10

Flags: <Local-Adv Symmetric-ADV IRB-Intf-Nexthop>

Last IPC-Event: add, Last IPC-Status: none, Last IPC-Timestamp: Jun 24 00:48:53.193 2026

Local origin: ae10.0

Remote origin: 3.1.1.3

Remote state: <Sent-to-l2ald MH-Proxy MH-Proxy-l2ald Symmetric-Adv>

Remote route distinguisher: 3.1.1.3:32

L3 route: 32.1.1.10/32, L3 context: VRF1_IP_VRF (irb.32)

History db:

Time	Event
Jun 24 00:48:32.239 2026	Rmt Add [], RmtPE 3.1.1.3, RD 3.1.1.3:32, ESI 00:00:00:00:00:00:00:32:32, label 10032, gbptag 0, mob-seq 0 op: 0x10024e000 chg: 0x0
Jun 24 00:48:32.239 2026	Rmt Add [], RmtPE 3.1.1.3, RD 3.1.1.3:32, ESI 00:00:00:00:00:00:00:32:32, label 10032, gbptag 0, mob-seq 0, IPv4: 32.1.1.10 op: 0x190044a000 chg: 0x0
Jun 24 00:48:32.239 2026	00:00:00:00:00:00:00:32:32 : Updating output state (change flags 0x400080 <ESI-Peer-Added Anycast-Ip-Chg>)
Jun 24 00:48:32.239 2026	00:00:00:00:00:00:00:32:32 : 32.1.1.10 Updating output state (change flags 0x2024002 <IP-Peer-Added IP-Remote-Com-Change IP-Remote-Symmetric-ADV-CHANGE IP-Anycast-IP-Changed>)
Jun 24 00:48:53.193 2026	Rmt Add [Mhapb], RmtPE 3.1.1.3, RD 3.1.1.3:32, ESI 00:00:00:00:00:00:00:32:32, label 10032, gbptag 0, mob-seq 0, IPv4: 32.1.1.10 op: 0x290044e000 chg: 0x0
Jun 24 00:48:53.193 2026	00:00:00:00:00:00:00:32:32 : Updating output state (change flags 0x400000 <Anycast-Ip-Chg>)
Jun 24 00:48:53.193 2026	00:00:00:00:00:00:00:32:32 : 32.1.1.10 Updating output state (change flags 0x6000 <IP-Remote-MH-Adv-Proxy-Bit IP-Remote-Com-Change>)
Jun 24 00:49:08.595 2026	Lcl Add [Lcl], intf ae10.0, ESI 3232, IPv4: 32.1.1.10 op: 0x480008000 chg: 0x0
Jun 24 00:54:08.287 2026	Lcl Add [Lcl], intf ae10.0, ESI 3232, IPv4: 32.1.1.10 op: 0x480008000 chg: 0x0
Jun 24 00:59:09.265 2026	Lcl Add [Lcl], intf ae10.0, ESI 3232, IPv4: 32.1.1.10 op: 0x480008000 chg: 0x0

Expected behavior after anycast MH is working

- Both Leaf-2a and Leaf-2b should be in apply-groups adv_anycast during normal operation.
- Spines should have reachability to 3.1.1.200/32 through both Junos leafs when both leafs are healthy.
- BFD should be Up for the underlay and overlay BGP sessions; the configs use minimum-interval 100 and multiplier 3.
- For core-isolation recovery, ae10 should return after 50 seconds and the anycast VTEP advertisement should return after 100 seconds.
- The measured core-isolation convergence with anycast was approximately 0.1 seconds for both Leaf-2a and Leaf-2b in the result table.

Check	Expected result
Underlay route to 3.1.1.200	Remote nodes and spines should have reachability to the Junos anycast VTEP.
EOS EVPN next-hop for Junos routes	Should resolve to the anycast VTEP in interop mode.
EOS show vxlan vtep	Should show the Junos anycast VTEP once routes are received and resolved.

Host-1 to Host-2 ping	Should work through L3 VNI 3333 and egress VLAN 32.
Leaf-2a/Leaf-2b peer ELP behavior	Peer should have protection/reroute state using the unique peer reroute address.

Benefits of Anycast Multihoming Compared to Regular ESI

Relationship between anycast and ELP

Anycast multihoming and ELP should be described as independent features that can be combined. Regular ESI multihoming can use ELP-style local repair without representing the pair as a single anycast VTEP. Anycast multihoming can improve remote VTEP representation and BUM handling, but access-link protection is strongest when ELP local repair is also active.

For the benefit comparison, the key message is anycast changes the remote forwarding model; ELP changes the local repair model. The combined solution gives both a clean single-VTEP remote view and fast local repair for egress access-link failures.

This chapter summarizes why the anycast model is useful compared to the regular EVPN ESI model. The goal is not to replace ESI. The goal is to make the multihomed pair look like one logical VTEP to the remote fabric while still preserving per-leaf repair capability inside the pair.

Regular ESI model versus anycast VTEP model

Area	Regular EVPN ESI model	Anycast multihoming model
Remote view of the pair	Remote VTEPs may see Leaf-2a and Leaf-2b as separate egress VTEPs.	Remote VTEPs can treat Leaf-2a/Leaf-2b as one logical egress VTEP, 3.1.1.200.
Known-unicast forwarding	Remote side may depend more directly on individual PE/ESI aliasing behavior.	Remote side sends to the anycast VTEP; underlay ECMP chooses a healthy receiving leaf.
Failure handling	Classic ESI behavior relies on aliasing, mass withdraw, and route convergence.	Local ELP repair can protect an access-link failure first; global repair can optimize later.
BUM traffic	DF election is used to prevent duplicate BUM delivery when multiple PEs can receive BUM copies.	Remote node sends one BUM copy to the anycast VTEP, so classic DF election is not required for steady-state anycast BUM.
Interop simplicity	Remote side must understand multiple ESI peers and their individual VTEPs.	Interop mode presents a simpler anycast VTEP next-hop to EOS.

Operational model	Troubleshooting often focuses on individual egress VTEP choices.	Troubleshooting focuses first on anycast VTEP reachability, then on ELP/reroute and local host attachment.
-------------------	--	--

Main benefits in this lab

- Single logical egress endpoint: EOS can send traffic to 3.1.1.200 instead of treating 3.1.1.2 and 3.1.1.3 as unrelated VTEPs.
- Better load sharing for normal traffic: underlay ECMP can distribute traffic to the Junos pair through the anycast VTEP.
- Fast local access-link repair: if traffic lands on a leaf whose local access link is down, ELP can redirect traffic to the surviving peer.
- Cleaner BUM behavior: remote nodes send one BUM copy to the anycast VTEP instead of sending duplicate copies to both multihoming peers.
- Clear separation of addresses: 3.1.1.200 is the shared anycast VTEP; 3.1.1.2 and 3.1.1.3 remain unique reroute/ELP addresses.
- Improved interop story: in mlag-interop-mode, EOS sees the Junos multihoming pair through the anycast VTEP next-hop.

DF election clarification

In regular EVPN multihoming, DF election is required for BUM because multiple PEs may otherwise forward duplicate BUM frames to the same Ethernet Segment. In the anycast BUM model, the remote VTEP sends only one BUM copy to the anycast VTEP. The underlay delivers that single copy to either Leaf-2a or Leaf-2b. Therefore, classic DF election is not required for steady-state anycast BUM forwarding.

The following command configures port- and ESI-level granularity to support convergence optimization. It does not select a single DF node:

```
set interfaces ae10 esi df-election-granularity per-esi
```

For anycast BUM forwarding, verify that the remote VTEP sends one BUM copy to 3.1.1.200 and that local bias prevents duplicate forwarding when ELP is present.

When regular ESI behavior is still useful

Regular ESI multihoming is still valid and widely used. It is useful when the deployment does not need the anycast VTEP abstraction, when platform support for anycast MH is not available, or when a design intentionally wants classic ESI aliasing and DF behavior. Anycast MH is an enhancement for specific multihoming and interop use cases; it is not a replacement for understanding normal EVPN ESI behavior.

Failover Test Case With and Without Anycast

Failover Test Cases on Junos

The tests focus on the Junos ESI side, where Host-2 is connected to VLAN 32. Test access-link failures by shutting down the port on the remote device. Test node failures by powering off the device in the lab GUI.

Test ID	Failure Type	Device	Interface(s) to shut	Expected Behavior
T1	Access link failure	Leaf-2a	et-0/0/2	Traffic may still land on Leaf-2a. ELP/local repair should redirect through Leaf-2b if supported.

T2	Access link failure	Leaf-2b	et-0/0/8	Traffic may still land on Leaf-2b. ELP/local repair should redirect through Leaf-2a if supported.
T3	Underlay uplink failure	Leaf-2a -> Spine-1	et-0/0/1	Underlay ECMP should use remaining uplink via Spine-2. Expected loss should be low.
T4	Underlay uplink failure	Leaf-2a -> Spine-2	et-0/0/0	Underlay ECMP should use remaining uplink via Spine-1. Expected loss should be low.
T5	Underlay uplink failure	Leaf-2b -> Spine-1	et-0/0/14	Underlay ECMP should use remaining uplink via Spine-2. Expected loss should be low.
T6	Underlay uplink failure	Leaf-2b -> Spine-2	et-0/0/12	Underlay ECMP should use remaining uplink via Spine-1. Expected loss should be low.
T7	Leaf-2a core isolation	Leaf-2a	et-0/0/1 and et-0/0/0	Simulates Leaf-2a losing both spine links. Traffic should move to Leaf-2b. This is closer to global/underlay repair.
T8	Leaf-2b core isolation	Leaf-2b	et-0/0/14 and et-0/0/12	Simulates Leaf-2b losing both spine links. Traffic should move to Leaf-2a.
T9	Leaf node reboot	Leaf-2a	device	Full node failure. Anycast/underlay should remove failed path and use Leaf-2b.
T10	Leaf node reboot	Leaf-2b	device	Full node failure. Anycast/underlay should remove failed path and use Leaf-2a.

Result Table - With Anycast Config

These results reflect the fast-convergence configuration: BFD 100 x 3 on BGP sessions, LACP fast, physical member hold-time, l2-learning no-mac-flush-on-aa-ae-down, and network-isolation hold-time up 50000 ms with event-script recovery timer 100 seconds.

Test ID	Test Case	Device / Interface	Convergence Time
T1	Access link failure	Leaf-2a / et-0/0/2 DOWN	7ms
T1		Leaf-2a / et-0/0/2 UP	0ms
T2	Access link failure	Leaf-2b / et-0/0/8 DOWN	7.8ms

T2		Leaf-2b / et-0/0/8 UP	0ms
T3	Underlay uplink failure	Leaf-2a -> Spine-1 / et-0/0/1 DOWN	0.7ms
T3		Leaf-2a -> Spine-1 / et-0/0/1 UP	0ms
T4	Underlay uplink failure	Leaf-2a -> Spine-2 / et-0/0/0 DOWN	1.1ms
T4		Leaf-2a -> Spine-2 / et-0/0/0 UP	0ms
T5	Underlay uplink failure	Leaf-2b -> Spine-1 / et-0/0/14 DOWN	1.4ms
T5		Leaf-2b -> Spine-1 / et-0/0/14 UP	0ms
T6	Underlay uplink failure	Leaf-2b -> Spine-2 / et-0/0/12 DOWN	1.3ms
T6		Leaf-2b -> Spine-2 / et-0/0/12 UP	0ms
T7	Leaf-2a core isolation	Leaf-2a / et-0/0/1 and et-0/0/0	110ms
T8	Leaf-2b core isolation	Leaf-2b / et-0/0/14 and et-0/0/12	100ms
T9	Leaf node reboot	Leaf-2a / device	70ms
T10	Leaf node reboot	Leaf-2b / device	29ms

Result Table - Without Anycast Config

Test ID	Test Case	Device / Interface	Convergence Time
T1	Access link failure	Leaf-2a / et-0/0/2 DOWN	36ms
T1		Leaf-2a / et-0/0/2 UP	0ms
T2	Access link failure	Leaf-2b / et-0/0/8 DOWN	35ms
T2		Leaf-2b / et-0/0/8 UP	0ms
T3	Underlay uplink failure	Leaf-2a -> Spine-1 / et-0/0/1 DOWN	27ms

T3		Leaf-2a -> Spine-1 / et-0/0/1 UP	0ms
T4	Underlay uplink failure	Leaf-2a -> Spine-2 / et-0/0/0 DOWN	28ms
T4		Leaf-2a -> Spine-2 / et-0/0/0 UP	0ms
T5	Underlay uplink failure	Leaf-2b -> Spine-1 / et-0/0/14 DOWN	36ms
T5		Leaf-2b -> Spine-1 / et-0/0/14 UP	0ms
T6	Underlay uplink failure	Leaf-2b -> Spine-2 / et-0/0/12 DOWN	30ms
T6		Leaf-2b -> Spine-2 / et-0/0/12 UP	0ms
T7	Leaf-2a core isolation	Leaf-2a / et-0/0/1 and et-0/0/0	235ms
T8	Leaf-2b core isolation	Leaf-2b / et-0/0/14 and et-0/0/12	188ms
T9	Leaf node reboot	Leaf-2a / device	95ms
T10	Leaf node reboot	Leaf-2b / device	75ms