

Troubleshoot PBR in ACI

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Abbreviations](#)

[PBR History](#)

[Design Considerations](#)

[Background Information](#)

[Scenario: Single VRF in a Single Pod Fabric](#)

[Network Diagram](#)

[Troubleshooting](#)

[IP SLA](#)

[Related Information](#)

Introduction

This document describes how to troubleshoot Application Centric Infrastructure (ACI) environments with Policy-Based Redirect (PBR) in a single pod fabric.

Prerequisites

Requirements

For this article, it is recommended that you have general knowledge of these topics:

- ACI concepts: Access Policies, Endpoint Learning, Contracts and L3out

Components Used

This troubleshooting exercise was made on ACI version 6.0(8f) using second generation Nexus switches N9K-C93180YC-EX and N9K-C93240YC-FX2.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Abbreviations

- BD: Bridge domain
- EPG: Endpoint group
- Class ID: Tag that identifies an EPG
- PBR node: L4-L7 device that is used for a PBR destination

- Consumer connector: PBR node interface facing the consumer side
- Provider connector: PBR node interface facing the provider side

PBR History

Version	Major Features
2.0(1m)	<ul style="list-style-type: none"> • Service graphs provide the PBR feature.
3.x and earlier	<ul style="list-style-type: none"> • Multi-node policy-based redirect (PBR) support • PBR Resilient Hashing
3.2(x)	<ul style="list-style-type: none"> • One-node Firewall PBR is supported in Multi-Site environments.
4.0(x)	<ul style="list-style-type: none"> • Two-node Firewall PBR is supported in Multi-Site environments.
4.2(1)	<ul style="list-style-type: none"> • A backup policy for creating standby PBR nodes is now supported.
4.2(3)	<ul style="list-style-type: none"> • Filter-from-contract option is available in the Service Graph template using GUI.
5.0(1)	<ul style="list-style-type: none"> • L3Outs are supported on all provider sides of the service nodes. • Active-active deployment/ECMP paths are supported for Layer 1/Layer 2 PBR devices.
5.2(1)	<ul style="list-style-type: none"> • A PBR destination can now be in an L3Out. • You can track service nodes using the HTTP URI. • Service graph managed and hybrid modes are no longer supported. • Dynamic PBR node mac address is supported.
6.0(1)	<ul style="list-style-type: none"> • Weight-Based Symmetric Policy-Based Redirect (PBR)

Design Considerations

- PBR works with both physical and virtual appliances
- PBR can be used between L3Out EPG and EPGs, between EPGs, and between L3Out EPGs. PBR is not supported if L2Out EPG is part of the contract
- PBR is supported in Cisco ACI Multi-Pod, Multi-Site, and Remote Leaf environments.
- The Cisco ACI fabric must be the gateway for the servers and for the PBR node
- The L4-L7 device must be deployed in go-to mode (routed mode)
- PBR node is not supported on FEX switches
- A dedicated Bridge domain is needed for PBR node
- Dynamic MAC address for the PBR node is now supported using health groups.
- It's recommended to enable GARP-based detection on the PBR node bridge domain
- Common default filter that includes ARP, ethernet traffic, and other non-IP traffic must not be used

for PBR

- The PBR node can be between VRF instances or within one of the VRF instances. For this topology, PBR node is not supported to be on a third VRF, has to be configured in either the consumer or the provider VRF

Background Information

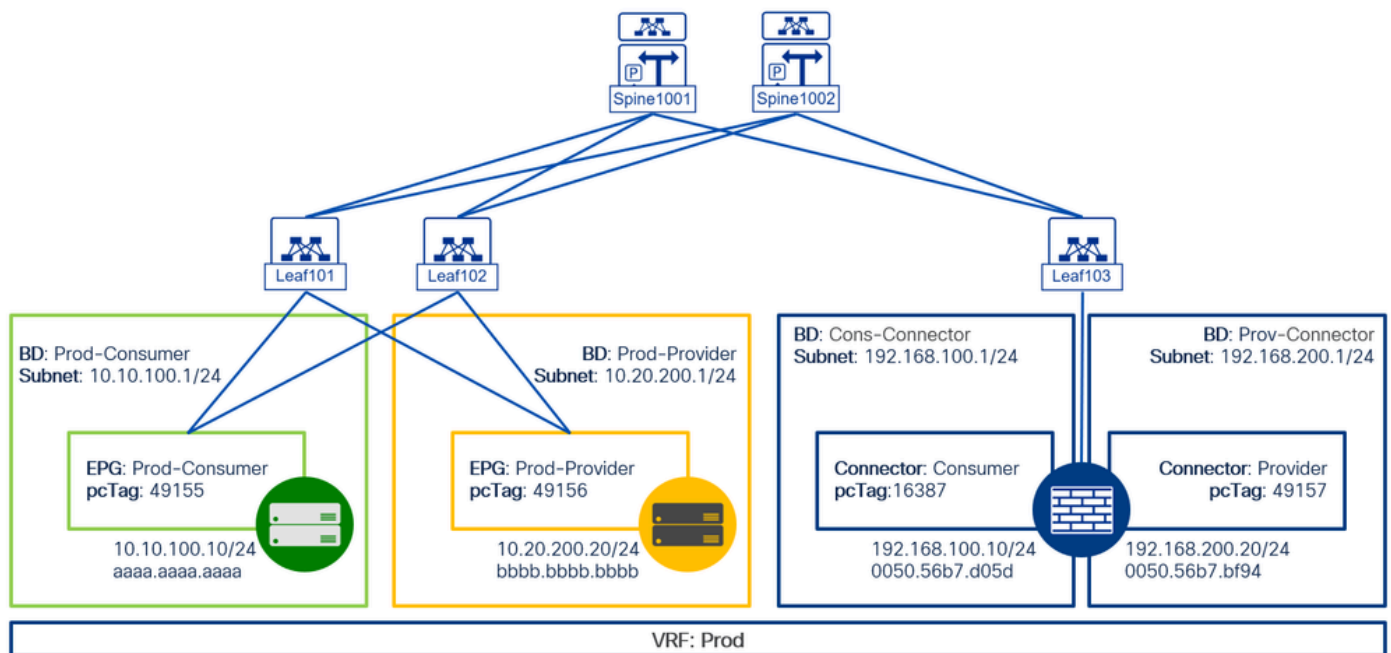
Deeper explanations of ELAM and Ftriage can be found in the CiscoLive On-Demand library in session [BRKDCN-3900b](#).

Additionally, all configuration guidelines can be found in the white paper [Cisco Application Centric Infrastructure Policy-Based Redirect Service Graph Design White Paper](#).

Scenario: Single VRF in a Single Pod Fabric

Network Diagram

Physical Topology:



Troubleshooting

Step 1: Faults

ACI generates a fault when there is an issue with the configuration or policy interactions. Specific faults have been identified for the PBR rendering process in the event of failure:

F1690: Configuration is invalid due to:

- ID allocation failure

This fault signifies that the encapsulated VLAN for the service node is unavailable. For instance, there could be no dynamic VLAN available in the VLAN pool associated with the Virtual Machine Manager (VMM) domain utilized by the Logical Device.

Resolution: Verify the VLAN pool within the domain employed by the Logical Device. If the Logical Device interface is within a physical domain, also inspect the encapsulated VLAN configuration. These settings can be found under **Tenant > Services > L4-L7 > Devices** and **Fabric > Access Policies > Pools > VLAN**.

Conversely, if the Logical Device interface resides within a virtual domain and connects to your ESXi hosts via a trunk interface, ensure that the **trunking port** option is enabled.

General

Name: TZ-PBR-Device

Alias:

Service Type: Firewall

Device Type: VIRTUAL

Trunking Port: ☒

VMM Domain: VMware/UCS-VCENTER

Promiscuous Mode: ☐

Context Aware: Multiple Single

Function Type: GoThrough GoTo L1 L2

- No device context found for LDev

This fault indicates that the Logical Device is not locatable for Service Graph rendering. For instance, there could be no Device Selection Policy that matches the contract associated with the Service Graph.

Resolution: Verify that a Device Selection Policy is defined. The Device Selection Policy specifies selection criteria for a service device and its connectors, based on the contract name, Service Graph name, and node name within the Service Graph. This can be found under **Tenant > Services > L4-L7 > Device Selection Policy**.

Properties



Contract Name: TZ-PBR-Contract

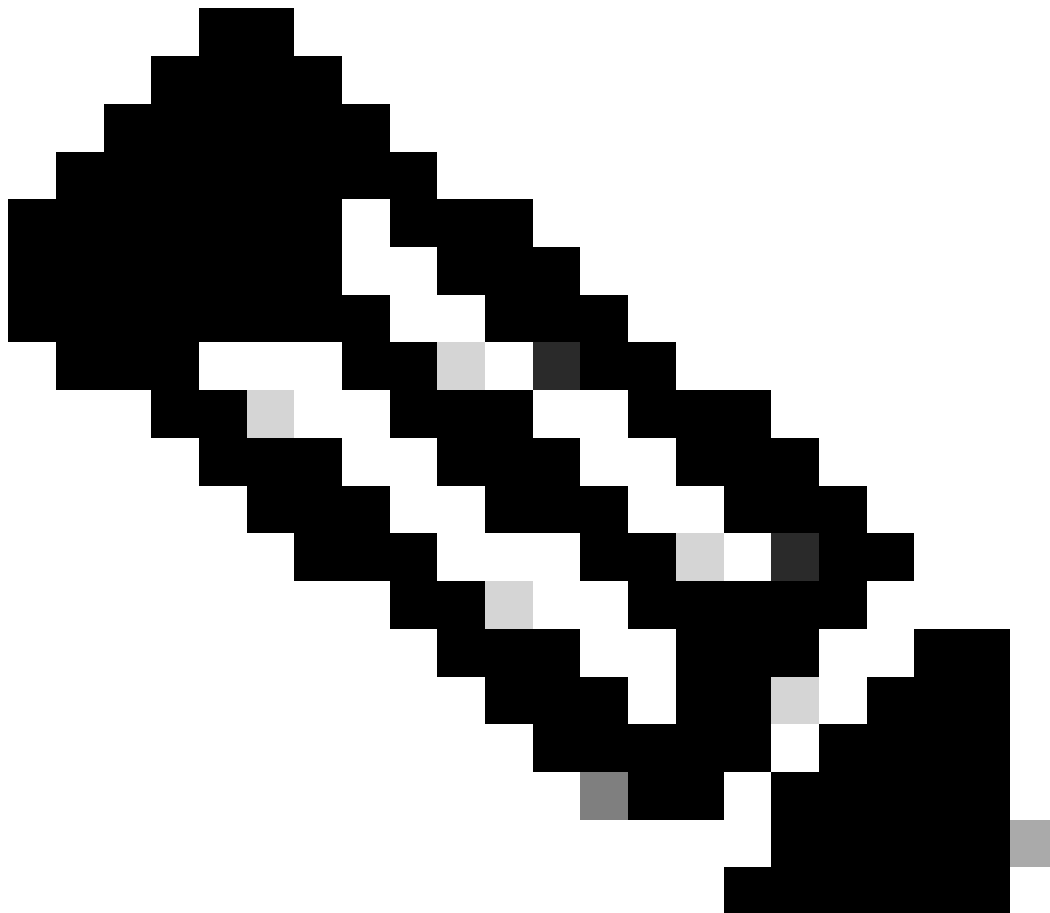
Graph Name: TZ-PBR-SG

Node Name: N1

Alias:

Context Name:

Devices:  



Note: When you deploy a Service Graph template, ACI pre-selects the bridge domain for the source EPG. You must change this bridge domain for the PBR consumer connector. The same applies for provider connector.

- No BD found

This fault indicates that the Bridge Domain (BD) for the service node cannot be located. For example, the BD is not specified in the Device Selection Policy.

Resolution: Ensure that the BD is specified in the Device Selection Policy and that the connector name is correct. This configuration is located under **Tenant > Services > L4-L7 > Devices Selection Policy > [Contract + SG] > [Consumer | Provider]**.

Properties

Connector Name: consumer

Cluster Interface: TZ-PBR-Cluster

Associated Network: Bridge Domain L3Out

Bridge Domain: Cons-Connector

Preferred Contract Group: Exclude

- LIf has no relation to CIf and LIf has an invalid CIf
- No cluster interface found

These faults indicate that the device has no relationship with the cluster interfaces.

Resolution: Ensure that the Layer 4 to Layer 7 (L4-L7) device configuration includes a specified concrete interface selector. This configuration is located under **Tenant > Services > L4-L7 > Devices > [Device] > Cluster Interfaces**.

Logical Interface - Cluster Interface - TZ-PBR-Cluster

Properties

Name: TZ-PBR-Cluster

Configuration Issues:

Concrete Interfaces:


Device Interface
TZ-FW([Out])
TZ-Firewall([In])

- Invalid service redirect policy

This fault signifies that the PBR policy has not been applied despite the redirection being activated on the service function within the Service Graph.


Resolution: Ensure that the PBR policy is configured within the Device Selection Policy settings. This configuration is located under **Tenant > Services > L4-L7 > Devices Selection Policy > [Contract + SG] > [Consumer | Provider]**.

Logical Interface Context - consumer




Properties

Connector Name: consumer

Cluster Interface: TZ-PBR-Cluster 


Associated Network: Bridge Domain L3Out


Bridge Domain: Cons-Connector 


Preferred Contract Group: Exclude

Permit Logging: ☐

L3 Destination (VIP): ☒

L4-L7 Policy-Based Redirect: TZ-PBR-Consumer 

L4-L7 Service EPG Policy: select an option 

Custom QoS Policy: select a value 

F0759: graph-rendering-failure - "Service graph for **tenant** < **tenant** > could not be instantiated. Function **node** < **node** > configuration is invalid."

The service graph for the specified tenant could not be instantiated due to an invalid configuration of the function node name.

This error suggests that there are configuration issues related to the aforementioned conditions.

Furthermore, during initial deployments, this fault can temporarily arise and then be promptly resolved. This occurs due to the rendering process that the ACI undergoes to deploy all policies.

Resolution: Investigate any additional faults that have been reported and resolve them accordingly.

F0764: configuration-failed - "L4-L7 Devices configuration < **device** > for tenant < **tenant** > is invalid."

The service graph for the specified tenant could not be instantiated due to an invalid configuration of the PBR Device policy.

This error suggests that there are configuration issues related to the aforementioned conditions.

Resolution: Investigate any additional faults that have been reported and resolve them accordingly.

F0772: configuration-failed - "L4-L7 Devices configuration < cluster > for L4-L7 Devices < device > for tenant < tenant > is invalid."

The service graph for the specified tenant could not be instantiated due to an invalid configuration of the PBR Device Cluster interface selection.

This error suggests that there are configuration issues related to the aforementioned conditions.

Resolution: Investigate any additional faults that have been reported and resolve them accordingly.

Step 2: Source and Destination Endpoint learning

Ensure that your source and destination endpoints are recognized within the fabric, which necessitates basic configuration:

- A Virtual Routing and Forwarding (VRF) object.
- A Bridge Domain (BD) object with Unicast Routing enabled. While enabling IP Data Plane learning is considered best practice, it is not mandatory.
- An Endpoint Group (EPG) object, applicable to both virtual and physical domains.
- Access Policies: Verify that your access policy configuration chain is complete and that no faults are reported on the EPG.

To confirm endpoint learning on the correct EPG and interface, execute this command on the leaf where the endpoint is learned, also known as compute leaf:

```
<#root>
```

```
show system internal epm endpoint [ ip | mac ] [ x.x.x.x | eeee.eeee.eeee ]
```

```
<#root>
```

```
Leaf101#
```

```
show system internal epm endpoint ip 10.10.100.10
```

```
MAC :
```

```
aaaa.aaaa.aaaa
```

```
::: Num IPs : 1
```

```
IP# 0 : 10.10.100.10
```

```
::: IP# 0 flags : ::: l3-sw-hit: No  
Vlan id : 57 ::: Vlan vnid : 10865 :::
```

```
VRF name : TZ:Prod
```

```
BD vnid : 16056291 ::: VRF vnid : 2162692  
Phy If : 0x16000008 ::: Tunnel If : 0  
Interface :
```

```
port-channel9
```

```
Flags : 0x80004c05 :::
```

```
sclass : 49155
```

```
::: Ref count : 5
```


EP Create Timestamp : 02/18/2025 15:00:18.767228
EP Update Timestamp : 02/18/2025 15:04:57.908343
EP Flags : local|VPC|IP|MAC|sclass|timer|

::::

Leaf101#

This command allows you to identify the pcTag (sclass) associated with the EPG where the endpoint is classified, as well as retrieve the interface, VRF scope, and MAC address information.

If you do not know the location of your source or destination endpoint, you can always assist yourself with this command on the APIC:

<#root>

show endpoint [ip | mac] [x.x.x.x | eeee.eeee.eeee]

<#root>

APIC#

show endpoint ip 10.10.100.10

Legends:

(P):Primary VLAN

(S):Secondary VLAN

Dynamic Endpoints:

Tenant : TZ

Application : TZ

AEPg : Prod-Consumer

End Point MAC	IP Address	Source	Node	Interface

AA:AA:AA:AA:AA:AA	10.10.100.10			
		learned, vmm		
101 102	vpc VPC-ESX-169			
	vlan-2673	not-applicable	2025-02-18T15:16:40.	

Total Dynamic Endpoints: 1

Total Static Endpoints: 0

APIC#

In the GUI, the **EP Tracker** feature can be accessed navigating to **Operations > EP Tracker** for endpoint monitoring and management.

SystemTenantsFabricVirtual NetworkingAdminOperationsAppsIntegrations

Visibility & Troubleshooting | Capacity Dashboard | EP Tracker | Visualization

EP Tracker

End Point Search

10.10.100.10

Search

Learned At	Tenant	Application	EPG	IP
1/101-1/102, vPC: VPC-ESX-169 (learned,vmm)	TZ	TZ	Prod-Consumer	10.10.100.10

With the information gathered from source and destination endpoints, you can now focus on PBR policy deployment.

Step 3: Redirect Contract

PBR is integrated within the Service Graph framework. Consequently, a Service Graph template must be deployed and configured on both the source and destination switches in accordance with a contract. Utilizing the pcTags information collected in the previous step, you can ascertain whether an Endpoint Group (EPG) is being redirected to a Service Graph group by executing this command.

<#root>

```
show zoning-rule scope [ vrf_scope ]
```

In the zoning rules, these rules must be considered:

- 1. Source EPG to Destination EPG with an associated redirect group
- 2. Source PBR-node shadow EPG to Source EPG
- 3. Destination EPG to Source EPG with an associated redirect group. This group can be identical to or different from the previous configuration.
- 4. Destination PBR-node shadow EPG to Destination EPG

<#root>

Leaf101#

```
show zoning-rule scope 2162692
```

Rule ID	SrcEPG	DstEPG	FilterID	Dir	operSt	Scope	Name	Action
4565	49155	49156	default	bi-dir	enabled	2162692		redir(destgrp-8)
4565	49156	49155	default	uni-dir-ignore	enabled	2162692		redir(destgrp-9)
4973	16387	49155	default	uni-dir	enabled	2162692		permit
4564	49157	49156	default	uni-dir	enabled	2162692		permit

Leaf101#

To verify the pcTag of the Shadow EPGs created during the Policy-Based Routing (PBR) deployment process, navigate to **Tenants > [TENANT_NAME] > Services > L4-L7 > Deployed Graph Instance > [**

SG_NAME] > Function Node - N1.

Function Node - N1

Policy Faults History

Properties

Name: N1
Function Type: GoTo
Devices: TZ-PBR-Device

Cluster Interfaces:

Name	Concrete Interfaces	Encap
TZ-PBR-Cluster	TZ-FW/[Out], TZ-Firewall/[In]	unknown

Function Connectors:

Name	Encap	Class ID	L3OutPBR Service pcTag
consumer	vlan-2675	16387	any
provider	vlan-2674	49157	any

- Contract parser

The script correlates zoning rules, filters, statistics, and EPG names. You can safely execute this script directly on an ACI leaf or APIC. When run on the APIC, it collects concrete objects across all leaf switches, which can take several minutes for large policy deployments.

Starting with ACI version 3.2, the `contract_parser` is bundled within the image and available on the leaf. Simply enter `contract_parser.py` from the iBash shell.

<#root>

Leaf101#

```
contract_parser.py --sepg 49155
```

Key:

```
[prio:RuleId] [vrf:{str}] action protocol src-epg [src-l4] dst-epg [dst-l4] [flags][contract:{str}] [hi
```

```
[7:4999] [vrf:TZ:Prod] log,
```

```
redir
```

```
ip tn-TZ/ap-TZ/epg-
```

```
Prod-Consumer(49155)
```

```
tn-TZ/ap-TZ/epg-
```

```
Prod-Provider(49156)
```

```
[contract:uni/tn-TZ/brc-
```

```
TZ-PBR-Contract
```

```
] [
```

```
hit=81
```

```
]
```

```
destgrp-8
```

```
vrf:TZ:Prod ip:
192.168.100.10
mac:
00:50:56:B7:D0:5D
bd:uni/tn-TZ/
BD-Cons-Connector
```

```
Leaf101#
```

This command provides details such as the contract action, source and destination EPGs, the contract name in use, and hit count.

Step 4: Redirection Group

Having identified the redirect group based on the contract applied to the zoning rules, the next step is to determine the IP and MAC addresses of the device(s) targeted for redirection. To assist with this, execute the command:

```
<#root>
```

```
show service redir info group [ destgrp_ID ]
```

```
<#root>
```

```
Leaf101#
```

```
show service redir info group 8
```

```
=====
LEGEND
```

```
TL: Threshold(Low) | TH: Threshold(High) | HP: HashProfile | HG: HealthGrp | BAC: Backup-Dest | TRA: Tr
```

```
=====
```

GrpID	Name	destination	HG-name	BAC W	operSt	operStQual	TL	TH
-------	------	-------------	---------	-------	--------	------------	----	----

8	destgrp-8	dest-[
---	-----------	--------	--	--	--	--	--	--

```
192.168.100.10
```

```
]-[vxlan-
```

```
2162692
```

```
] Not attached N 1
```

```
enabled
```

```
no-oper-grp 0 0 sym no no
```

```
Leaf101#
```

```
Leaf101# show service redir info group 9
```

```
=====
LEGEND
```

```
TL: Threshold(Low) | TH: Threshold(High) | HP: HashProfile | HG: HealthGrp | BAC: Backup-Dest | TRA: Tr
```

```
=====
```

```

GrpID Name      destination      HG-name      BAC W      operSt      operStQual      TL      TH
=====
9      destgrp-9 dest-[
192.168.200.20
]-[vxlan-
2162692
] Not attached   N      1
enabled
no-oper-grp 0      0      sym no      no
Leaf101#

```

This command enables us to determine the operational status (OperSt) of our redirection group, the IP address configured in the L4-L7 PBR section, and the VNID of the VRF associated with the PBR-node bridge domains. You now need to determine the configured MAC address:

```

<#root>
show service redir info destinations ip [ PBR-node IP ] vnid [ VRF_VNID ]

<#root>
Leaf101#
show service redir info destination ip 192.168.100.10 vnid 2162692

=====
LEGEND
TL: Threshold(Low) | TH: Threshold(High) | HP: HashProfile | HG: HealthGrp | BAC: Backup-Dest | TRA: Tr
=====
Name      bdVnid      vMac      vrf      operSt      operStQual      HG-
=====
dest-[
192.168.100.10
]-[vxlan-
2162692
] vxlan-
15826939

00:50:56:B7:D0:5D

TZ:Prod
enabled
no-oper-dest Not attached
Leaf101#
Leaf101# show service redir info destination ip 192.168.200.20 vnid 2162692

```

=====

LEGEND

TL: Threshold(Low) | TH: Threshold(High) | HP: HashProfile | HG: HealthGrp | BAC: Backup-Dest | TRA: Tr

=====

Name	bdVnid	vMac	vrf	operSt	operStQual	HG-
=====	=====	=====	=====	=====	=====	=====
dest-[
192.168.200.20						
]-[vxlan-						
2162692						
] vxlan-						
16646036 00:50:56:B7:BF:94						
TZ:Prod						
enabled						
no-oper-dest Not attached						
Leaf101#						

In addition to the previously mentioned details, this command provides valuable insights, including the VRF name, BD VNID, and the configured MAC address of the PBR-node.



Note: It is important to note that both the IP and MAC addresses are user-configured at this stage, which means that typographical errors can occur during the L4-L7 Policy-Based Routing definition.

Step 5: PBR-node not receiving any traffic.

A prevalent issue encountered with PBR forwarding is the absence of traffic reaching the PBR node. A frequent cause of this issue is an incorrectly specified MAC address within the L4-L7 Policy-Based Routing configuration.

To verify the accuracy of the MAC address configured in the L4-L7 Policy-Based Routing, execute the previously utilized command from Step 2. This command can be executed on the leaf switch designated as the service leaf, where the node is expected to be learned.

<#root>

```
show system internal epm endpoint [ ip | mac ] [ x.x.x.x | eeee.eeee.eeee ]
```

<#root>

Leaf103#

show system internal epm endpoint ip 192.168.100.10

MAC :

0050.56b7.d05d

::: Num IPs : 1

IP# 0 :

192.168.100.10

::: IP# 0 flags : ::: l3-sw-hit: Yes ::: flags2 :

dp-lrn-dis

Vlan id : 71 ::: Vlan vnid : 10867 ::: VRF name : TZ:Prod

BD vnid : 15826939 ::: VRF vnid :

2162692

Phy If : 0x16000008 ::: Tunnel If : 0

Interface : port-channel19

Flags : 0x80004c25 ::: sclass : 16387 ::: Ref count : 5

EP Create Timestamp : 02/19/2025 12:07:44.065032

EP Update Timestamp : 02/19/2025 15:27:03.400086

EP Flags : local|vPC|peer-aged|IP|MAC|sclass|timer|

::::

Leaf103#

.....

Leaf103#

show system internal epm endpoint ip 192.168.200.20

MAC :

0050.56b7.bf94

::: Num IPs : 1

IP# 0 :

192.168.200.20

::: IP# 0 flags : ::: l3-sw-hit: Yes ::: flags2 :

dp-lrn-dis

Vlan id : 60 ::: Vlan vnid : 10866 ::: VRF name : TZ:Prod

BD vnid : 16646036 ::: VRF vnid :

2162692


```
Phy If : 0x16000008 ::: Tunnel If : 0
Interface : port-channel19
Flags : 0x80004c25 ::: sclass : 49157 ::: Ref count : 5
EP Create Timestamp : 02/19/2025 13:51:03.377942
EP Update Timestamp : 02/19/2025 15:28:34.151877
EP Flags : local|vPC|peer-aged|IP|MAC|sc|class|timer|

:::
```

Leaf103#

Verify that the MAC address recorded in the EPM table matches the one configured in the service redirection group. Even minor typographical errors must be corrected to ensure proper traffic routing to the PBR node destination.

Step 6: Traffic Flow.

- FTRIAGE

A CLI tool for the APIC designed to automate the configuration and interpretation of ELAM processes end-to-end. The tool allows users to specify a particular flow and the leaf switch where the flow originates. It sequentially executes ELAMs on each node to analyze the forwarding path of the flow. This tool is especially beneficial in complex topologies where the packet path is not easily discernible.

<#root>

APIC #

```
ftriage -user admin route -sip 10.10.100.10 -dip 10.20.200.20 -ii LEAF:101,102
```

Starting ftriage

Log file name for the current run is: ftlog_2025-02-25-10-26-05-108.txt

```
2025-02-25 10:26:05,116 INFO /controller/bin/ftriage -user admin route -sip 10.10.100.10 -dip 10.20.200.20
Request password info for username: admin
Password:
2025-02-25 10:26:31,759 INFO ftriage: main:2505 Invoking ftriage with username: admin
2025-02-25 10:26:34,188 INFO ftriage: main:1546 Enable Async parallel ELAM with 2 nodes
2025-02-25 10:26:57,927 INFO ftriage: fcls:2510
```

LEAF101

```
: Valid ELAM for asic:0 slice:0 srcid:64 pktid:1913
2025-02-25 10:26:59,120 INFO ftriage: fcls:2863
```

LEAF101

```
: Signal ELAM found for Async lookup
2025-02-25 10:27:00,620 INFO ftriage: main:1317 L3 packet
```

Seen on LEAF101

Ingress:

Eth1/45 (Po9)

Egress: Eth1/52 Vnid: 2673

2025-02-25 10:27:00,632 INFO ftriage: main:1372 LEAF101: Incoming Packet captured with [
SIP:10.10.100.10, DIP:10.20.200.20
]
...
2025-02-25 10:27:08,665 INFO ftriage: main:480 Ingress
BD(s) TZ:Prod-Consumer

2025-02-25 10:27:08,666 INFO ftriage: main:491 Ingress
Ctx: TZ:Prod Vnid: 2162692

...
2025-02-25 10:27:45,337 INFO ftriage: pktrec:367 LEAF101:
traffic is redirected

...
2025-02-25 10:28:10,701 INFO ftriage: unicast:1550 LEAF101:
traffic is redirected
to vnid:15826939
mac:00:50:56:B7:D0:5D
via tenant:TZ
graph:TZ-PBR-SG
contract:
TZ-PBR-Contract

...
2025-02-25 10:28:20,339 INFO ftriage: main:975
Found peer-node SPINE1001
and IF: Eth1/1 in candidate list
...
2025-02-25 10:28:39,471 INFO ftriage: main:1366
SPINE1001
: Incoming Packet captured with Outer [SIP:10.2.200.64, DIP:10.2.64.97] Inner [
SIP:10.10.100.10, DIP:10.20.200.20
]
2025-02-25 10:28:39,472 INFO ftriage: main:1408
SPINE1001
: Outgoing packet's Vnid:
15826939

2025-02-25 10:28:58,469 INFO ftriage: fib:524
SPINE1001: Proxy in spine

```

...
2025-02-25 10:29:07,898 INFO ftriage: main:975

Found peer-node LEAF103

. and IF: Eth1/50 in candidate list
...
2025-02-25 10:29:35,331 INFO ftriage: main:1366

LEAF103

: Incoming Packet captured with Outer [SIP:10.2.200.64, DIP:10.2.200.64] .... Inner [
SIP:10.10.100.10, DIP:10.20.200.20

]
...
2025-02-25 10:29:50,277 INFO ftriage: ep:128 LEAF103: pbr traffic with dmac:
00:50:56:B7:D0:5D

2025-02-25 10:30:07,374 INFO ftriage: main:800 Computed egress encap string
vlan-2676

2025-02-25 10:30:13,326 INFO ftriage: main:535 Egress

Ctx TZ:Prod

2025-02-25 10:30:13,326 INFO ftriage: main:536 Egress BD(s):

TZ:Cons-Connector

...
2025-02-25 10:30:18,812 INFO ftriage: misc:908 LEAF103: caller unicast:581

EP if(Po19)

    same as egr if(Po19)
2025-02-25 10:30:18,812 INFO ftriage: misc:910 LEAF103: L3 packet caller unicast:668 getting
bridged

    in SUG
2025-02-25 10:30:18,813 INFO ftriage: main:1822 dbg_sub_nexthop function returned values on node LEAF10
2025-02-25 10:30:19,378 INFO ftriage: acigraph:794 : Ftriage Completed with hunch: matching service dev
APIC #

```

- ELAM:

The Embedded Logic Analyzer Module (ELAM) is a diagnostic tool that enables users to establish specific conditions in hardware to capture the initial packet or frame meeting those criteria. When a capture is successful, the ELAM status is indicated as triggered. Upon triggering, the ELAM is disabled, allowing for a data dump to be collected, which facilitates the analysis of the numerous forwarding decisions executed by the ASIC of the switch for that packet or frame. ELAM operates at the ASIC level, ensuring that it does not affect the CPU or other resources of the switch.

Structure of the command syntax. This structure was collected from the book [Troubleshoot ACI Intra-Fabric](#)

[Forwarding Tools](#)

```
vsh_lc [This command enters the line card shell where ELAMs are running]
debug platform internal <asic> elam asic 0 [refer to the ASICs table]
```

Set conditions to trigger

```
trigger reset [ensures no existing triggers are running]
trigger init in-select <number> out-select <number> [determines what information about a packet is captured]
set outer/inner [sets conditions]
start [starts the trigger]
status [checks if a packet is captured]
```

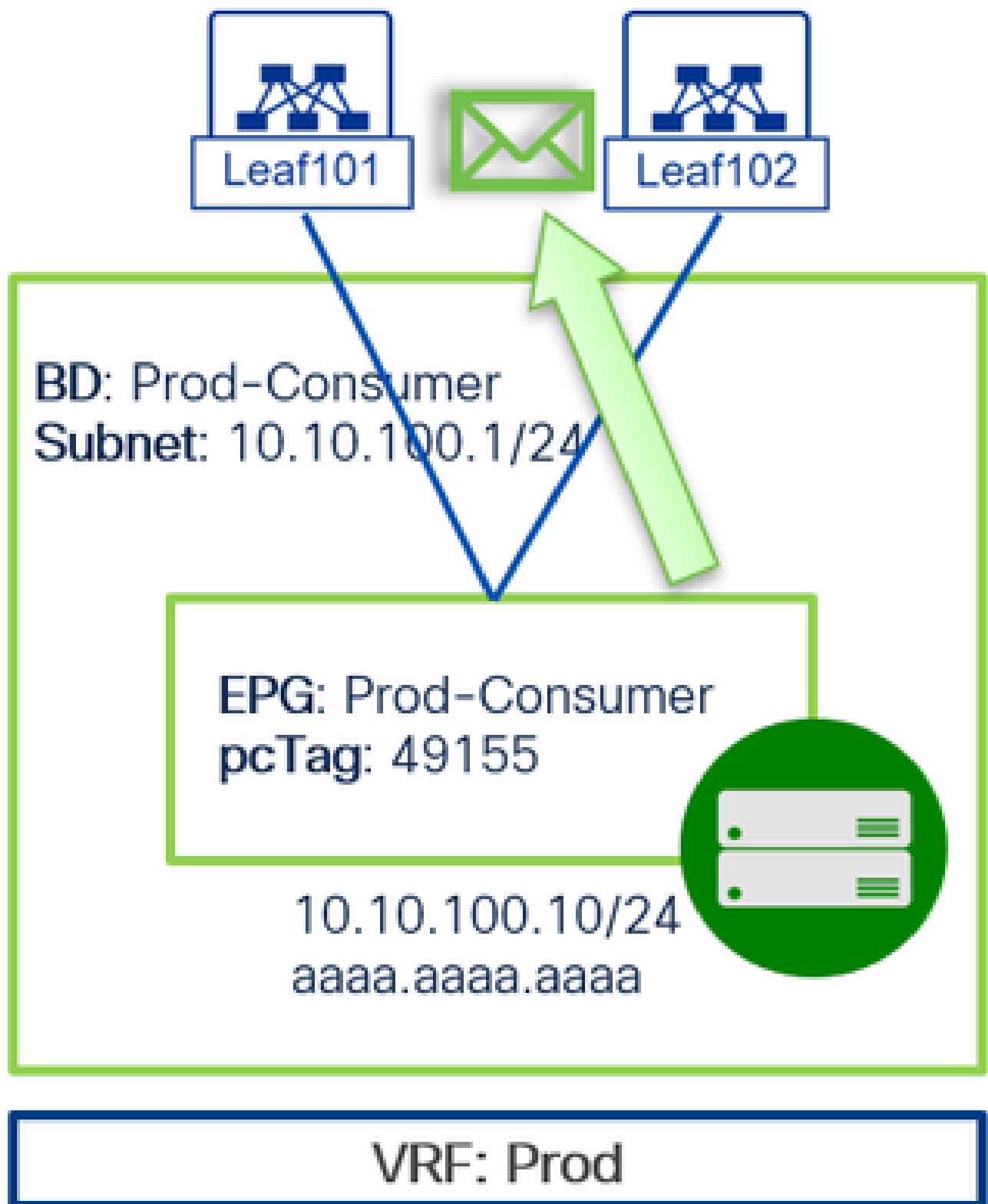
Generate the dump containing the packet analysis.

```
ereport [display detailed forwarding decision for the packet]
```

- Traffic flow

It is crucial to comprehend the traffic flow across all the devices in question. The Ftriage tool provides an excellent summary of this flow. However, for a detailed step-by-step validation and to gain deeper insights into the packet reception process, you can execute Embedded Logic Analyzer Module (ELAM) at each point within the network topology.

1. Ingress traffic occurs at the compute leaf where the source server is learned. In this specific scenario, as the source is positioned behind a vPC interface, ELAM must be configured on the vPC peers. This is necessary because the physical interface selected by the hashing algorithm is indeterminate.



```
<#root>
```

```
LEAF101#
```

```
vsh_lc
```

```
module-1#
```

```
debug platform internal tah elamasic 0
```

module-1(DBG-elam)#

trigger reset

module-1(DBG-elam)#

trigger init in-select 6 out-select 1

module-1(DBG-elam-inse16)#

reset

module-1(DBG-elam-inse16)#

set outer ipv4 src_ip 10.10.100.10 dst_ip 10.20.200.20

module-1(DBG-elam-inse16)#

start

module-1(DBG-elam-inse16)#

status

ELAM STATUS

=====

Asic 0 Slice 0 Status

Triggered

Asic 0 Slice 1 Status Armed

module-1(DBG-elam-inse16)#

ereport

=====

Trigger/Basic Information

=====

...

Incoming Interface : 0x40(0x40)

>>> Eth1/45

...

Outer L2 Header

Destination MAC : 0022.BDF8.19FF >>> Bridge-domain MAC address

Source MAC : AAAA.AAAA.AAAA

802.1Q tag is valid : yes(0x1)
CoS : 0(0x0)

Access Encap VLAN : 2673

(0xA71)

Outer L3 Header

L3 Type : IPv4
IP Version : 4
DSCP : 0
IP Packet Length : 84 (= IP header(28 bytes) + IP payload)
Don't Fragment Bit : set
TTL : 64
IP Protocol Number : ICMP
IP CheckSum : 6465(0x1941)

Destination IP : 10.20.200.20

Source IP : 10.10.100.10

Contract Lookup Key

IP Protocol : ICMP(0x1)
L4 Src Port : 2048(0x800)
L4 Dst Port : 7345(0x1CB1)

sclass (src pcTag) : 49155(0xC003) >>> Prod-Consumer

EPG

dclass (dst pcTag) : 49156(0xC004)

>>> Prod-Provider EPG

src pcTag is from local table : yes

>>> EPGs are known locally

derived from a local table on this node by the lookup of src IP or MAC
Unknown Unicast / Flood Packet : no
If yes, Contract is not applied here because it is flooded

Sideband Information

ovector : 176(0xB0) >>> Eth1/52

```
Ovec in "show plat int hal 12 port gpd"
```

```
Opcode : OPCODE_UC
```

```
sug_luc_latch_results_vec.luc3_0.
```

```
service_redir: 0x1 >>> Service Redir 0x1 = PBR was applied
```

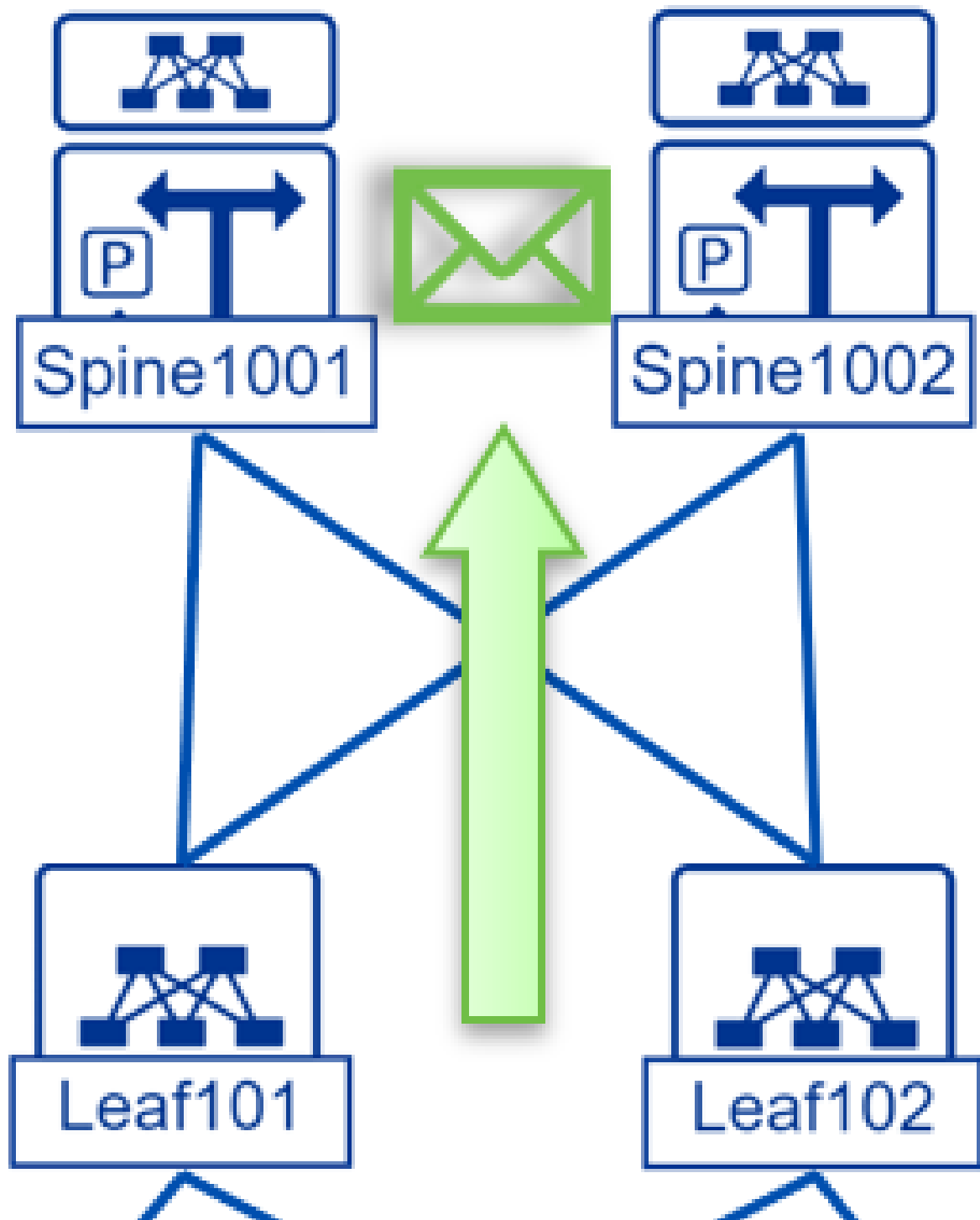
Based on the information provided, it is evident that the packet is being redirected via Policy-Based Routing (PBR), as the service_redir option is enabled. Additionally, you retrieve the sclass and dclass values. In this particular scenario, the switch recognizes the dclass. However, if the destination endpoint is not present in the EPM table, the dclass value defaults to 1.

Furthermore, the ingress interface is determined by the SRCID, and the egress interface is identified by the vector values. These values can be translated into a front port by executing this command at the vsh_lc level:

```
<#root>
```

```
show platform internal hal 12 port gpd
```

2. The subsequent step in the flow involves reaching the spine switch to map the destination MAC address to the PBR node. Since the traffic is encapsulated in VXLAN headers, executing ELAM on a spine or remote leaf requires using in-select 14 to properly decode the encapsulation.



```
<#root>
```

```
SPINE1001#
```

```
vsh_lc
```

```
module-1#
```

```
debug platform internal roc elam asic 0
```

module-1(DBG-elam)#

trigger reset

module-1(DBG-elam)#

trigger init in-select 14 out-selec 0

module-1(DBG-elam-inse114)#

reset

module-1(DBG-elam-inse114)#

set inner ipv4 src_ip 10.10.100.10 dst_ip 10.20.200.20

module-1(DBG-elam-inse114)#

start

module-1(DBG-elam-inse114)#

status

ELAM STATUS

=====

Asic 0 Slice 0 Status Armed

Asic 0 Slice 1 Status Armed

Asic 0 Slice 2 Status Triggered

Asic 0 Slice 3 Status Armed

module-1(DBG-elam-inse114)#

ereport

=====

Trigger/Basic Information

=====

Incoming Interface :

0x48(0x48) >>> Eth1/1

(Slice Source ID(Ss) in "show plat int hal l2 port gpd")

Packet from vPC peer LEAF : yes

Packet from tunnel (remote leaf/avs) : yes

Outer L2 Header

Destination MAC : 000D.0D0D.0D0D

Source MAC : 000C.0C0C.0C0C

Inner L2 Header

Inner Destination MAC : 0050.56B7.D05D >>> Firewall MAC

Source MAC : AAAA.AAAA.AAAA

Outer L3 Header

L3 Type : IPv4
DSCP : 0
Don't Fragment Bit : 0x0
TTL : 32
IP Protocol Number : UDP
Destination IP : 10.2.64.97
Source IP : 10.2.200.64

Inner L3 Header

L3 Type : IPv4
DSCP : 0
Don't Fragment Bit : 0x1
TTL : 63
IP Protocol Number : ICMP
Destination IP : 10.20.200.20

Source IP : 10.10.100.10

Outer L4 Header

L4 Type : iVxLAN
Don't Learn Bit : 1
Src Policy Applied Bit : 1
Dst Policy Applied Bit : 1
sclass (src pcTag) : 0xc003 >>> pcTag 49155 (Prod-Consumer)

VRF or BD VNID : 15826939(0xF17FFB) >>> BD: Prod-Consumer

Sideband Information

Opcode : OPCODE_UC

bky_elam_out_sidebnd_no_spare_vec.

ovector_idx: 0x1F0 >>> Eth1/10

From the previous output, it is evident that the destination MAC address is rewritten to the MAC address of the firewall. Subsequently, a COOP lookup is performed to identify the destination publisher of the MAC, and the packet is then forwarded to the corresponding interface of the switch.

You can simulate this lookup on the spine by executing this command, utilizing the Bridge Domain VNID and the MAC address of the firewall:

<#root>

SPINE1001#

```
show coop internal info repo ep key 15826939 0050.56B7.D05D | egrep "Tunnel|EP" | head -n 3
```

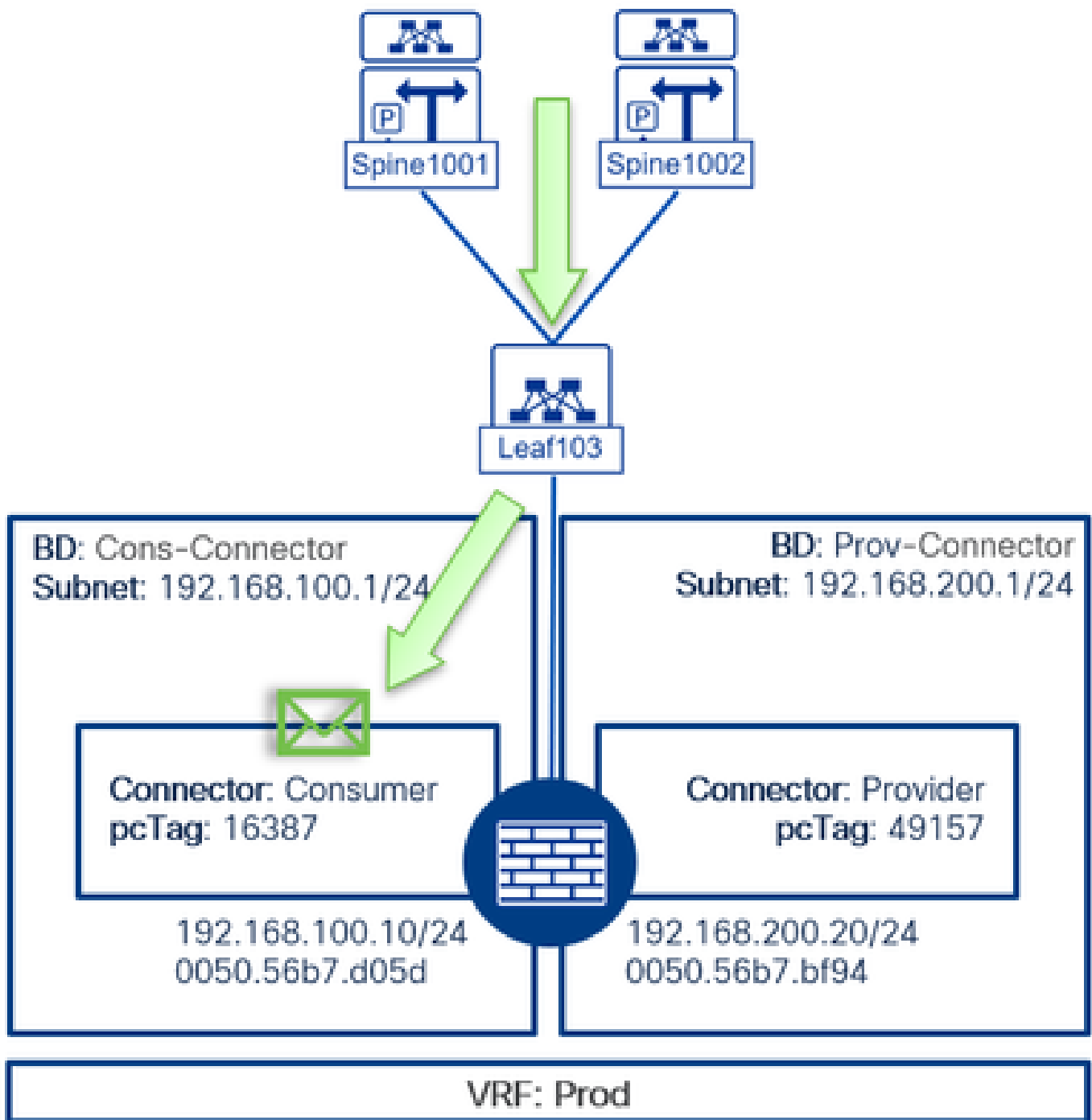
EP bd vnid : 15826939

EP mac : 00:50:56:B7:D0:5D

Tunnel nh : 10.2.200.66

SPINE1001#

3. Traffic reaches the service leaf where the MAC address of the firewall is recognized and subsequently forwarded to the PBR node.



```
<#root>
```

```
MXS2-LF101#
```

```
vsh_lc
```

```
module-1#
```

```
debug platform internal tah elam asic 0
```

```
module-1(DBG-elam)#
```

```
trigger reset
```

module-1(DBG-elam)#

trigger init in-select 14 out-select 1

module-1(DBG-elam-inse114)#

reset

module-1(DBG-elam-inse114)#

set inner ipv4 src_ip 10.10.100.10 dst_ip 10.20.200.20

module-1(DBG-elam-inse114)#

start

module-1(DBG-elam-inse114)#

status

ELAM STATUS

=====

Asic 0 Slice 0 Status Armed

Asic 0 Slice 1 Status Triggered

module-1(DBG-elam-inse114)#

ereport

=====

Trigger/Basic Information

=====

Incoming Interface : 0x0(0x0) >>> Eth1/17

(Slice Source ID(Ss) in "show plat int hal 12 port gpd")

Packet from vPC peer LEAF : yes

Packet from tunnel (remote leaf/avs) : yes

Outer L2 Header

Destination MAC : 000D.0D0D.0D0D

Source MAC : 000C.0C0C.0C0C

Inner L2 Header

Inner

Destination MAC : 0050.56B7.D05D >>> Firewall MAC

Source MAC : AAAA.AAAA.AAAA

Outer L3 Header

L3 Type : IPv4
DSCP : 0
Don't Fragment Bit : 0x0
TTL : 32
IP Protocol Number : UDP
Destination IP : 10.2.200.66
Source IP : 10.2.200.64

Inner L3 Header

L3 Type : IPv4
DSCP : 0
Don't Fragment Bit : 0x1
TTL : 63
IP Protocol Number : ICMP

Destination IP : 10.20.200.20

Source IP : 10.10.100.10

Outer L4 Header

L4 Type : iVxLAN
Don't Learn Bit : 1
Src Policy Applied Bit : 1
Dst Policy Applied Bit : 1

sclass (src pcTag) : 0xc003 >>> pcTag 49155 (Prod-Consumer)

VRF or BD VNID : 15826939(0xF17FFB) >>> BD: Prod-Consumer

Contract Lookup Key

IP Protocol : ICMP(0x1)
L4 Src Port : 2048(0x800)
L4 Dst Port : 50664(0xC5E8)

sclass (src pcTag) : 49155(0xC003) >>> Prod-Consumer EPG

dclass (dst pcTag) : 16387(0x4003) >>> Consumer connector EPG

src pcTag is from local table : no
derived from group-id in iVxLAN header of incoming packet
Unknown Unicast / Flood Packet : no
If yes, Contract is not applied here because it is flooded

Sideband Information

ovector

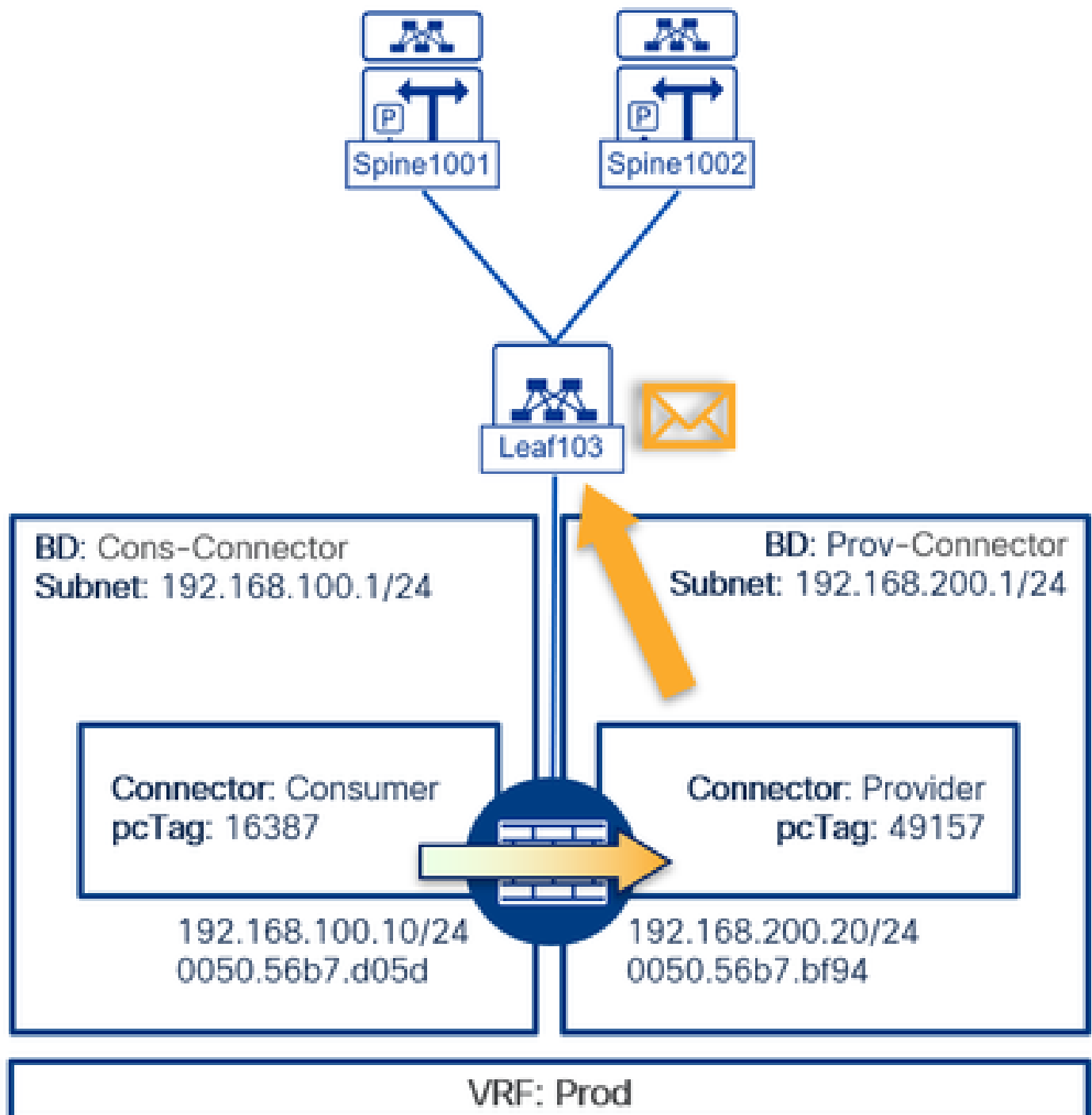
:

64(0x40) >>> Eth1/45

Ovec in "show plat int ha1 12 port gpd"

Opcode : OPCODE_UC

4. For the packer being returned by the PBR node, first, this one has to make its own diligence and have the VRF, interface or VLAN changed. The packet is then going to be forwarded back to ACI on the Provider connector:



<#root>

LEAF103#

vsh_lc

module-1#

debug platform internal tah elam asic 0

module-1(DBG-elam)#

trigger reset

module-1(DBG-elam)#

trigger init in-select 6 out-select 1

module-1(DBG-elam-insel6)#

reset

module-1(DBG-elam-insel6)#

set outer ipv4 src_ip 10.10.100.10 dst_ip 10.20.200.20

module-1(DBG-elam-insel6)#

set outer l2 src_mac 0050.56b7.bf94

module-1(DBG-elam-insel6)#

start

module-1(DBG-elam-insel6)#

stat

ELAM STATUS

=====

Asic 0 Slice 0 Status Triggered

Asic 0 Slice 1 Status Armed

module-1(DBG-elam-insel6)#

ereport

=====
Trigger/Basic Information
=====

...

Incoming Interface : 0x40(0x40)

>>> Eth1/45

...

Outer L2 Header

Destination MAC : 0022.BDF8.19FF

Source MAC : 0050.56B7.BF94

802.1Q tag is valid : yes(0x1)

CoS : 0(0x0)

Access Encap VLAN : 2006(0x7D6)

Outer L3 Header

L3 Type : IPv4

IP Version : 4

DSCP : 0

IP Packet Length : 84 (= IP header(28 bytes) + IP payload)

Don't Fragment Bit : set

TTL : 62

IP Protocol Number : ICMP

IP CheckSum : 46178(0xB462)

Destination IP : 10.20.200.20

Source IP : 10.10.100.10

Contract Lookup Key

IP Protocol : ICMP(0x1)

L4 Src Port : 2048(0x800)

L4 Dst Port : 37489(0x9271)

sclass (src pcTag) : 49157(0xC005) >>> Provider connector EPG

dclass (dst pcTag) : 49156(0xC004)

>>> Prod-Provider EPG

src pcTag is from local table : yes

derived from a local table on this node by the lookup of src IP or MAC

Unknown Unicast / Flood Packet : no

If yes, Contract is not applied here because it is flooded

Sideband Information

ovector : 176(0xB0) >>> Eth1/52

Ovec in "show plat int ha1 12 port gpd"

Opcode : OPCODE_UC

sug_luc_latch_results_vec.luc3_0.

service_redir: 0x0

5. The remaining network traffic adheres to the established steps mentioned so far, wherein packets return to the spine switches to determine the ultimate server destination, based on the coop lookup. Subsequently, packets are directed to the compute leaf switch that has the local learning flag and disseminated this information to the COOP table on the spines. The ELAM execution for steps 2 and steps 3 are consistent for the distribution of the packet towards its final destination. It is essential to validate these the destination EPG pcTag and the egress interface to ensure accurate delivery.

IP SLA

IP SLA is utilized to assess the operational status and performance of network paths. It aids in ensuring that traffic is routed efficiently according to the defined policies, based on real-time network conditions. In ACI, PBR leverages IP SLA to make informed routing decisions. If the IP SLA metrics indicate that a path is under performing, PBR can reroute traffic through alternate paths that meet the required performance criteria.

Starting on version 5.2(1) you can enable Dynamic MAC tracking for IP SLA, this is useful for scenarios where a PBR-nodes fails over and changes the MAC address for the same IP address, in static deployments, a change in the Policy-Based Redirect policy needs to be done every time the PBR-node changes its MAC address to continue sending traffic. With IP SLA, the MAC address used in this policy relays to the probe response. Different probes can be used to determine if the PBR node is healthy or not, at the moment of this writing, these include: ICMP, TCP, L2Ping and HTTP.

The general configuration of an IP SLA policy must look like:

IP SLA Monitoring Policy - tz-ipSLA



Properties

Name: tz-ipSLA

Description: optional

SLA Type:

ICMP

TCP

L2Ping

HTTP

SLA Frequency (sec): 60

Detect Multiplier: 3

Request Data Size (bytes): 28

Type of Service: 0

Operation Timeout (milliseconds): 900

Threshold (milliseconds): 900

Traffic Class Value: 0

The IP SLA policy has to be mapped to the PBR-node IP by means of a Health Group.





Create L4-L7 Redirect Health Group





Name: tz-HG

Description: optional

If IP SLA to dynamically discover the MAC address is used, this field cannot be empty, but set to all 0s:





Properties

IP: 192.168.100.10

Destination Name:

Description:

MAC:

Additional IPv4/IPv6:

Pod ID:

Weight:

Redirect Health Group:

Virtual Dynamic MAC of Service Node: 00:00:00:00:00:00

Implicit Service VRF VNID: 0

Show Usage Close Submit

Once a Health Group associates with the PBR-node IP via the L3 Destination in the L4-L7 Policy Based Redirect policy, set thresholds to define IP SLA behavior upon unreachability. Specify a minimum number of active L3 Destinations required to maintain redirection. If the count of live PBR nodes falls under or exceeds the threshold percentage, the entire group is affected by the **Threshold Down Action** selected, stopping redistribution. This approach supports bypassing the PBR-node during troubleshooting without impacting traffic flow.

Threshold Enable: ☒

Min Threshold Percent (percentage):

Max Threshold Percent (percentage):

Threshold Down Action:

bypass action deny action permit action

Health Groups bind together all the IPs defined in L3 Destinations of a single L4-L7 Policy Base Redirect policy or multiple L4-L7 Policy Base Redirect policies so long the same Health Group policy is used.

```
Leaf101# show service redir info health-group aperezos::tz-HG
=====
LEGEND
TL: Threshold(Low) | TH: Threshold(High) | HP: HashProfile | HG: HealthGrp | BAC: Backup-Dest | TRA: Tr
=====
HG-Name HG-OperSt HG-Dest HG-Dest-OperSt
=====
aperezos::tz-HG enabled dest-[192.168.100.10]-[vxlan-2162692]] up
                        dest-[192.168.200.20]-[vxlan-2162692]] up
Leaf101#
```

Related Information

- [Cisco Application Centric Infrastructure Policy-Based Redirect Service Graph Design White Paper](#)
- [ACI Layer 4-7 Policy-Based Redirect \(PBR\) Deep Dive and Tips \(Cisco Live Presentation\)](#)
- [Troubleshoot IP SLA on Multipod PBR](#)