

**Figure 6-10** Domain View Updated Automatically with AAEP Association

Example 6-1 shows the APIC CLI equivalents for the configurations outlined in this section.

**Example 6-1** CLI Equivalents for VLAN Pool, Domain, and AAEP Configurations

[Click here to view code image](#)

```
apic1# show running-config vlan-domain DCACI-Domain
  vlan-domain DCACI-Domain type phys
    vlan-pool DCACI-VLANs
      vlan 910-920
    exit
```

## Note

AAEP configuration cannot be performed via the APIC CLI, but the existence of AAEP objects can be verified via MOQuery.

## Policies and Policy Groups

This chapter has made multiple references to the configuration of individual ports, port channels, and vPCs, but it has not discussed the objects and relationships that enable policy assignment to switch ports.

Some of the most important constructs of interest in configuring a switch port are interface policies, interface policy groups, switch policies, and switch policy groups.

## Interface Policies and Interface Policy Groups



In ACI, configuration parameters that dictate interface behavior are called **interface policies**. Examples of interface policies include port speeds, enabled or disabled protocols or port level features, and monitoring settings.

It is common for interface policies to be defined when an ACI fabric is initialized unless automation is employed to dynamically create objects that define desired interface policies as part of an interface configuration script.

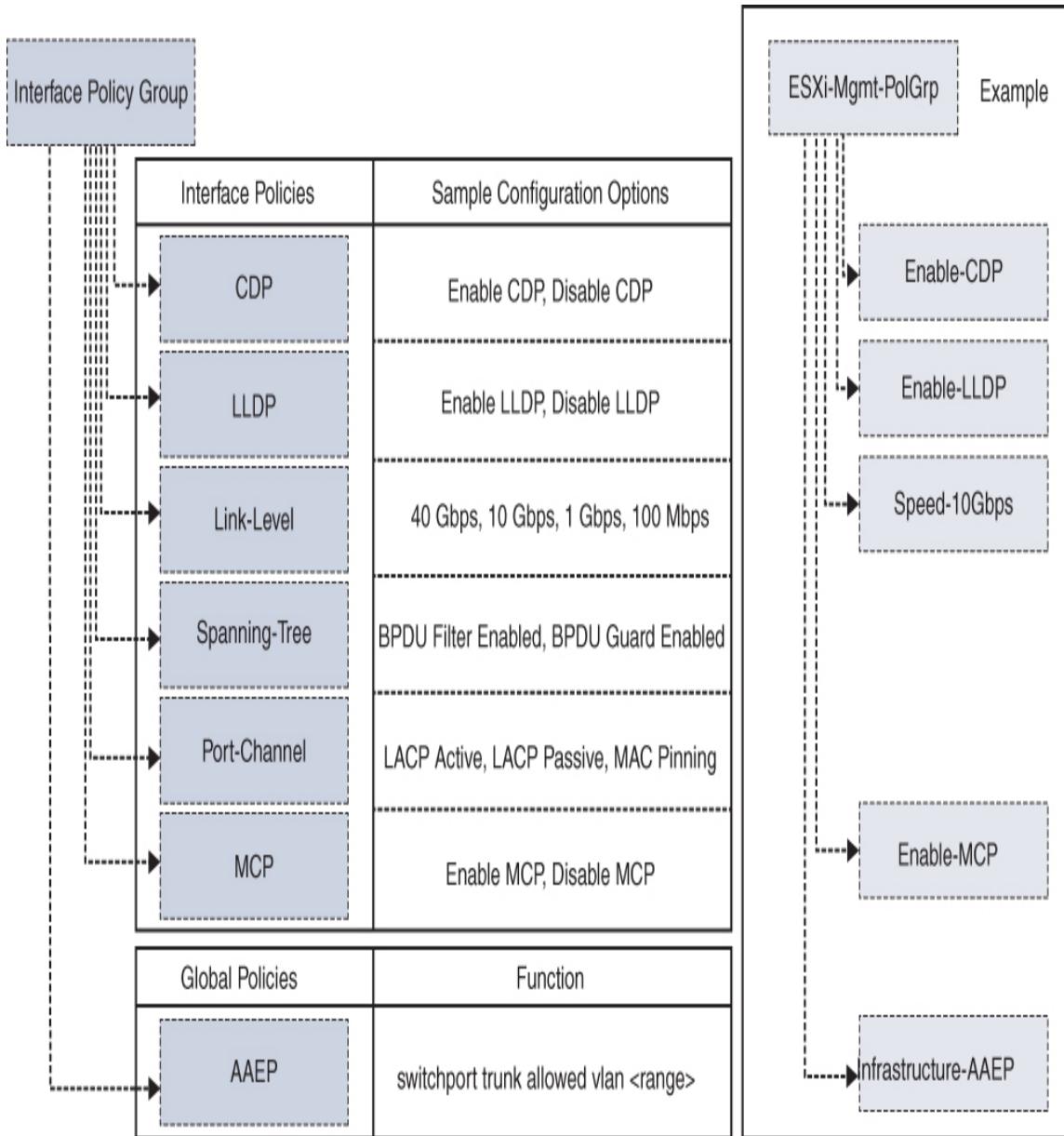
For instance, when an ACI fabric is initialized, an administrator may create an interface policy called LLDP-Enable, which enables the LLDP protocol. The administrator may also create another interface policy called LACP-Active

to enable link aggregation via LACP. Interface policies do not take effect by themselves; they need to be assigned to an interface policy group to be applicable to switch ports. When the two interface policies LLDP-Enable and LACP-Active are applied together to an interface policy group and assigned to a set of ports, the newly configured ports then enable LLDP and attempt to form an LACP port channel.



An ***interface policy*** group is a port configuration template that aligns with link types. Each individual physical interface or link aggregation within ACI derives two critical configuration components from an interface policy group: The first is a collection of interface policies, and the second is an AAEP. Some types of interface policy groups are fully reusable, and others are semi-reusable.

[Figure 6-11](#) illustrates the relationships between and functions of interface policies and interface policy groups.



**Figure 6-11** *Interface Policies and Interface Policy Groups*

As illustrated in this figure, you can think of an interface policy group as having two components. The first component is the collection or grouping of interface policies applicable to a set of ports that therefore dictates the features (for example, control plane policing, storm control), protocols (for example, Cisco Discovery Protocol, Link Layer

Discovery Protocol), and other link aspects (for example, speed) applied to relevant ports. The second component of an interface policy group is the AAEP, which dictates the domains associated with the interface policy group and therefore determines the tenants and VLAN IDs that can be encapsulated on the relevant ports.

In NX-OS, interface configuration lines related to Cisco Discovery Protocol (CDP) and storm control would be considered interface policies. An NX-OS configuration line like **switchport trunk allowed vlan 100-200** dictates the range of acceptable VLANs and therefore would be equivalent in function to an AAEP.

**Table 6-6** describes the types of interface policy groups available for ACI leaf switches at the time of this writing.



**Table 6-6** Types of Interface Policy Groups in ACI

Interface Policy Group Type	Description
Leaf access port policy group	Fully reusable interface configuration templates used for individual (non-aggregated) non-fabric ports on a leaf switch.

Interface Policy Group Type	Description
Port channel interface policy group	A single-switch port channel configuration template that can be applied to non-fabric ports on a leaf switch.
VPC interface policy group	A vPC configuration template used to create a port aggregation across two switches that are in the same vPC domain.
PC/VPC override policy group	Where an administrator reuses PC or VPC interface policy groups across multiple leafs or vPC domains, an override policy group can override the settings applied on an individual leaf or vPC domain.
Leaf breakout port group	Some ACI switches support breaking out high-bandwidth ports into lower-bandwidth ports using breakout cabling. This enables configuration of high-bandwidth ports for breakouts.

Interface Policy Group Type	Description
Fibre Channel (FC) interface policy group	A fully reusable interface policy group that allows the selection of an FC interface policy and AAEP for connectivity to SAN-accessing servers or a Fibre Channel Forwarder (FCF).
FC port channel interface policy group	A pseudo-reusable FC port channel interface policy group that allows the selection of an FC interface policy, a port channel policy, and an AAEP for connectivity to SAN-accessing servers or an FCF.

**Key Topic**

Not all interface policy groups are fully reusable. Leaf access port policy groups and FC interface policy groups are reusable without caveats because they do not necessitate link aggregation.

If a port channel interface policy group has already been used to deploy a port channel on a leaf switch, reuse of the PC interface policy group on that leaf results in the newly configured ports being added to the previously created bundle. A similar situation occurs when a vPC interface

policy group is used to create a vPC and the interface policy group is reused across the same vPC domain.

Therefore, if a port channel interface policy group has already been deployed to a leaf and the intent is to create a new port channel on the leaf, you can create and deploy a new PC interface policy group to the leaf. Likewise, if a vPC interface policy group has been deployed to a vPC switch pair, you should not reuse the vPC interface policy group unless the intent is to join new links to the previously created vPC.

[Table 6-7](#) presents the most commonly used types of interface policies available in ACI.



**Table 6-7** Types of Interface Policies Available in ACI

Inte rfac e Poli cy Typ e	Description of Settings
Link level	Determines a port link speed, auto-negotiation status, forward error correction (FEC), and link debounce interval.

CDP	Allows the creation of policies that enable or disable CDP.
LLDP	Allows the creation of policies that enable or disable LLDP.
NetFlow	Allows the creation of NetFlow monitors, NetFlow records, or NetFlow exporters for traffic and flow data collection at the interface level.
Port channel	Enables the creation of policies involving port aggregations, such as Link Aggregation Control Protocol (LACP) and static port channels. Additional options, including MAC pinning and explicit failover order, are available for virtual environments.
Spanning Tree	Allows the creation of policies that enable BPDU Guard, BPDU Filtering, or both.
Storm control	Enables the creation of policies that can prevent traffic disruptions on physical interfaces caused by a broadcast, multicast, or unknown unicast traffic storm.

MCP	Allows the creation of policies that enable or disable <b><i>MisCabling Protocol (MCP)</i></b> , which is a loop-prevention protocol in ACI. MCP can be applied on both physical Ethernet interfaces and port channel interfaces. MCP needs to be enabled globally for MCP interface policies to be applied.
CoPP (Control Plane Policing)	Control Plane Policing (CoPP) interface policies protect ACI switches by setting limits on the number of packets per second the switch may process in CPU when received on a link. CoPP policies are applied on a per-protocol basis.
L2 interface policy	L2 interface policies govern policies related to VLAN scopes, Q-in-Q encapsulation, and Reflexive Relay functionality. Later chapters address these policies.

As a supplement to the previous list, [Table 6-8](#) describes some of the less commonly used interface policy types available in ACI as of the time of this writing. Even though these interface policy types are not used as often, they still technically fall under the umbrella of access policies and therefore may be considered within the scope of the DCACI 300-620 exam.

**Table 6-8** Additional Interface Policy Types Available in ACI

## **Int Description of Settings**

**erf**

**ac**

**e**

**Po**

**lic**

**y**

**Ty**

**pe**

<p>Pri ori ty flo w co ntr ol</p>	<p>Enables or disables priority-based flow control (PFC) or sets it to automatic. PFC is a QoS PAUSE mechanism often used to ensure lossless Fibre Channel forwarding over an Ethernet medium.</p>
<p>Fib re Ch an nel int erf ac e</p>	<p>Sets the port speed, mode (F or NP), trunking status, and receive buffer credit size for Fibre Channel interfaces.</p>

Po E	Sets a Power over Ethernet (PoE) policy that can be applied to switch ports for direct phone or wireless attachment to an ACI fabric. Use of PoE in ACI fabrics is not very common.
Por t ch an nel me mb er	Defines a common policy that applies to one or more member interfaces within a port channel bundle. For example, LACP port priorities can be configured in a port channel member policy to help determine which ports should be put in standby mode when not all ports configured in the bundle can be moved into a forwarding state. LACP fast timers are also configured in a port channel member interface policy.
Da ta pla ne pol ici ng	Data plane policing (DPP) policies manage bandwidth consumption on ACI fabric access interfaces. DPP policies can apply to egress traffic, ingress traffic, or both. DPP monitors the data rates for a particular interface. When traffic exceeds user-configured values, marking or dropping of packets occurs immediately.
Por t se cur ity	Port security policies protect the ACI fabric from being flooded with unknown MAC addresses by limiting the number of MAC addresses per port.

MA Cs ec	MACsec is an IEEE 802.1AE standards-based Layer 2 hop-by-hop encryption that provides data confidentiality and integrity for media access independent protocols. MACsec interface policies can be used on host-facing links to secure switch-to-endpoint communication via MACsec.
D W DM	When a DWDM optic is inserted into an ACI switch, the port defaults to DWDM channel 32 and the corresponding frequency and wavelength. Using a DWDM interface policy, an administrator can change the DWDM channel used on a port.
Fir ew all	An interface policy used for implementation of the ACI Distributed Firewall in an ACI Virtual Edge environment.
80 2.1 X po rt aut he nti cat ion	802.1X port authentication interface policies allow administrators to restrict unauthorized endpoints from connecting to the network through ACI switch ports.

Slow drain	Slow drain interface policies manage FCoE traffic congestion by specifying the actions ACI should take if congestion is detected on an interface.
------------	---

### Note

Although some types of interface policies do have default values, it is highly recommended that you create and use explicit interface policies as much as possible.

[Chapter 7](#) covers the configuration of interface policies and interface policy groups.

## Planning Deployment of Interface Policies

Remember that all interface policies are reusable. While administrators usually deploy interface policy groups when new physical infrastructure is introduced into the data center, they tend to plan and configure a large set of interface policies at the time of initial fabric deployment. If a specific use arises for additional interface policies, the administrator can add the new interface policy to the deployment.

[Table 6-9](#) shows a basic sample collection of interface policies an administrator might configure at the time of fabric initialization. The data in this table is for learning purposes only and should not be interpreted as a recommendation for policy naming.

**Table 6-9** Sample Interface Policies Configured During Fabric Initialization

Interface Configuration Settings Selected for Policy	Policy Name
CDP-Enable	Sets CDP to enabled
CDP-Disable	Sets CDP to disabled
LLDP-Enable	Sets LLDP to enabled
LLDP-Disable	Sets LLDP to disabled
MCP-Enable	Sets MCP to enabled
MCP-Disable	Sets MCP to disabled

Interface Configuration Settings Selected for Policy Policy Name	
40-Gbps	A link-level policy that sets the port speed to 40 Gbps. All other settings may remain set to the defaults.
10-Gbps	A link-level policy that sets the port speed to 10 Gbps. All other settings may remain set to the defaults.
1-Gbps	A link-level policy that sets the port speed to 1 Gbps. All other settings may remain set to the defaults.
100-Mbps-Full	A link-level policy that sets the port speed to 100 Mbps, with auto-negotiation set to full.
LACP-Active	A port channel policy that sets LACP to active with suspend individual port, graceful convergence, and fast select hot standby ports enabled.

Interface Configuration Settings Selected for Policy	
Policy Name	
Static-On	A port channel policy used for the creation of static port channels.

## Switch Policies and Switch Policy Groups

Just like interfaces, switches at times require custom policies. An example of a custom policy might be specific CoPP settings or a vPC domain peer dead interval modification. Custom switch policies are defined using switch policies and are grouped together for allocation via switch policy groups.

ACI does not require that custom switch policies be defined and allocated to switches.

Configuration parameters that dictate switch behavior are called *switch policies*. A *switch policy group* is a switch configuration template that includes a set of switch policies for allocation to one or more switches in an ACI fabric. Switch policies and switch policy groups are usually configured during fabric initialization and are fully reusable.



**Table 6-10** outlines the most commonly deployed switch policies available in ACI as of the time of writing.



**Table 6-10** Most Commonly Deployed Switch Policies in ACI

Switch Policy	Description of Settings
CoPP (leaf and spine)	Enables the modification of switch Control Plane Policing (CoPP) profiles to allow a more lenient or more strict profile compared to the default CoPP switch profile. If the predefined CoPP profiles are not sufficient, a custom CoPP switch profile can be configured and allocated to switches.
BFD	Enables the configuration of global IPv4 and IPv6 Bidirectional Forwarding Detection (BFD) policies in the fabric to provide subsecond failure detection times in the forwarding path between ACI switches.

## Switch Description of Settings Traffic Policy

NetFlow node	Allows the configuration of NetFlow timers that specify the rate at which flow records are sent to the external collector.
Forwarding scaling profile	<p>This policy provides different scalability options, including the following:</p> <ul style="list-style-type: none"><li>■ <b>Dual Stack:</b> Provides scalability of up to 12,000 endpoints for IPv6 configurations and up to 24,000 endpoints for IPv4 configurations.</li><li>■ <b>High LPM:</b> Provides scalability similar to Dual Stack except that the longest prefix match (LPM) scale is 128,000, and the policy scale is 8000.</li></ul>

Switch Policies	Description of Settings
	<ul style="list-style-type: none"><li>■ <b>IPv4 Scale:</b> Enables systems with no IPv6 configurations to increase scalability to 48,000 IPv4 endpoints.</li><li>■ <b>High Dual Stack:</b> Provides scalability of up to 64,000 MAC endpoints and 64,000 IPv4 endpoints. IPv6 endpoint scale can be 24,000/48,000, depending on the switch hardware model.</li></ul>

[Table 6-11](#) describes some less commonly modified ACI switch policies.

**Table 6-11** Additional Switch Policies in ACI

## **Switch Description of Settings**

### **Spanning Tree Policies**

Spanning Tree	Enables configuration of certain spanning-tree policies, such as MST region policies.
Fibre Channel Node	Sets parameters related to the FCoE functionality of the switch, including disruptive load balancing and FIP keepalive intervals.
Fibre Channel SAN	Specifies FC map values, Error Detect Timeout (EDT) values, and Resource Allocation Timeout (RAT) values for an NPV leaf switch targeted to support FCoE connectivity.

PoE	Controls the overall default power consumption of a node switch, in milliwatts. Further interface-level policies, such as the PoE VLAN, need to also be configured to enable PoE power delivery to devices such as IP phones.
Fast link failover	Reduces the data plane outage resulting from a fabric link failure to less than 200 milliseconds. This feature requires EX leaf switches or newer switches. Enabling this feature on leaf uplinks prevents the use of these links for port mirroring.
CoPP prefilter	To protect against DDoS attacks, a CoPP prefilter profile can filter access to authentication services based on specified sources and TCP ports. When a CoPP prefilter profile is deployed on a switch, control plane traffic is denied by default. Only the traffic specified in the CoPP prefilter profile is permitted.

802.1X	802.1X authentication allows administrators to restrict unauthorized endpoints from connecting to the network through ACI switch ports. By default, leaf switches use the OOB (out-of-band) management IP address to source packets to a RADIUS server for 802.1X authentication. If you wish to use an in-band management IP address for communication with the 802.1X server, you need to configure an 802.1X node authentication policy and associate the RADIUS provider group to it.
Equipped policies	The SSD monitoring feature enables administrators to override the preconfigured thresholds for the SSD lifetime parameters. Faults are generated in ACI when the SSD reaches some percentage of the configured thresholds. These faults enable network operators to monitor and proactively replace any switch before the switch fails due to SSD lifetime parameter values becoming exceeded.

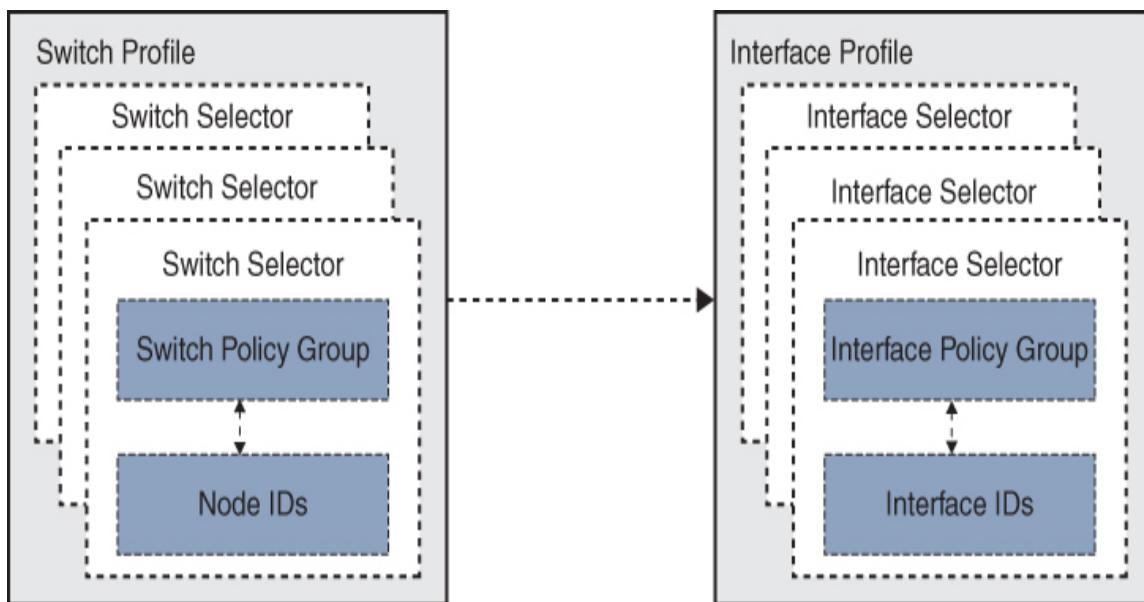
## Note

As demonstrated in [Chapter 4, “Exploring ACI,”](#) administrators assign vPC domain IDs and vPC domain policies to leaf switch pairs from the switch policies folder. These two switch policies cannot be allocated to switches using switch policy groups and are therefore not discussed here.

[Chapter 7](#) includes configuration examples for switch policies and switch policy groups.

## Profiles and Selectors

Once administrators create interface policy groups, they need to assign them to one or more ports. The port mapping occurs under an interface profile using an object called an *interface selector*, as shown in [Figure 6-12](#). The interface profile contains port mappings but not switch mappings. The switch mappings are determined through associations between interface profiles and switch profiles.

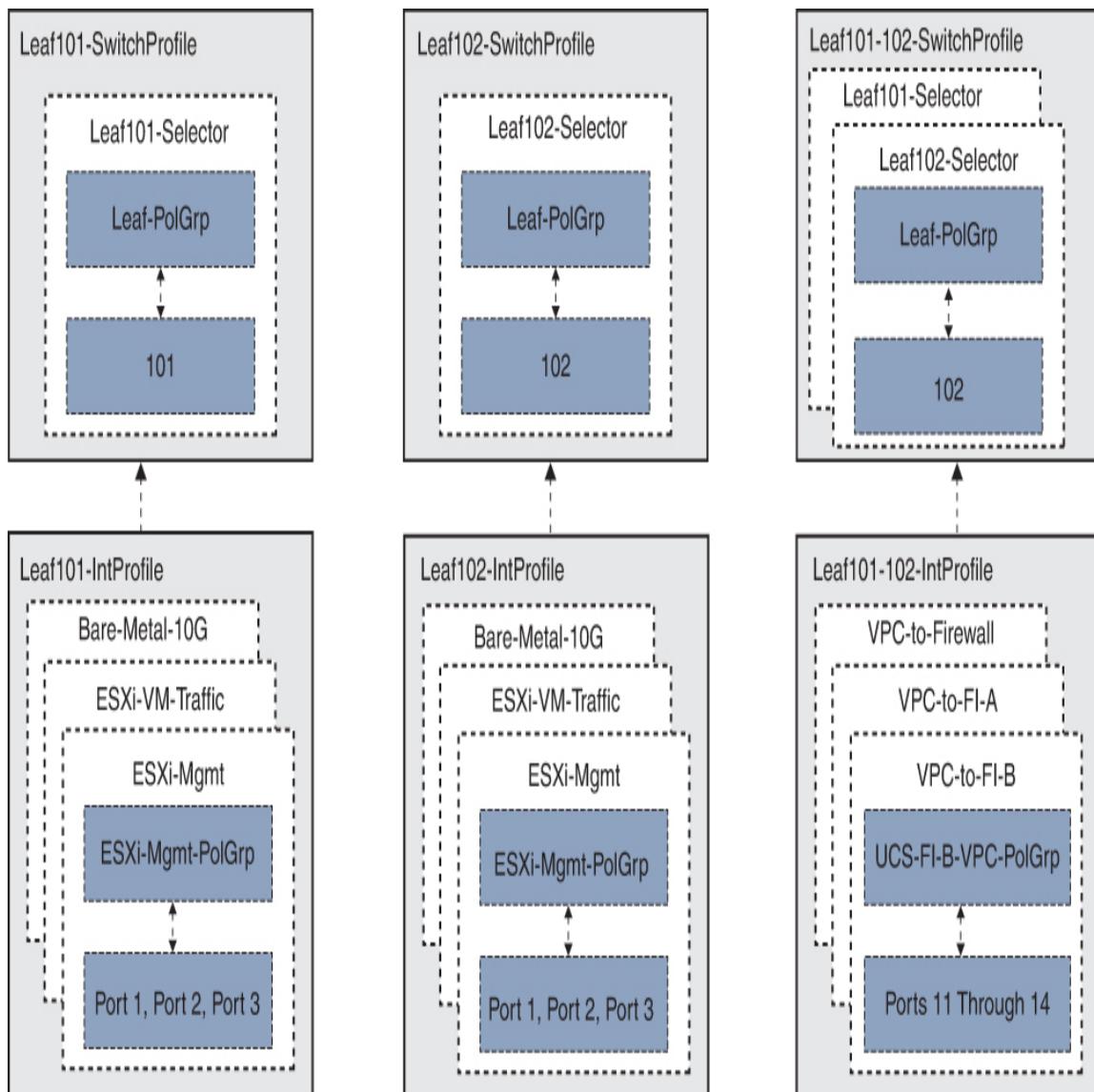


**Figure 6-12** Mapping Interface Policy Groups to Physical Ports

[Figure 6-13](#) provides some context for how this might be deployed in practice.

[Figure 6-13](#) shows that an administrator creates an interface selector called ESXi-Mgmt under an interface profile named Leaf101-IntProfile and maps an interface policy group named ESXi-Mgmt-PolGrp to Ports 1, 2, and 3. The

administrator creates a separate interface selector with the same name and port assignments under an interface profile called Leaf102-IntProfile. The administrator then associates Leaf101-IntProfile and Leaf102-IntProfile to switch profiles Leaf101-SwitchProfile and Leaf102-SwitchProfile, respectively.



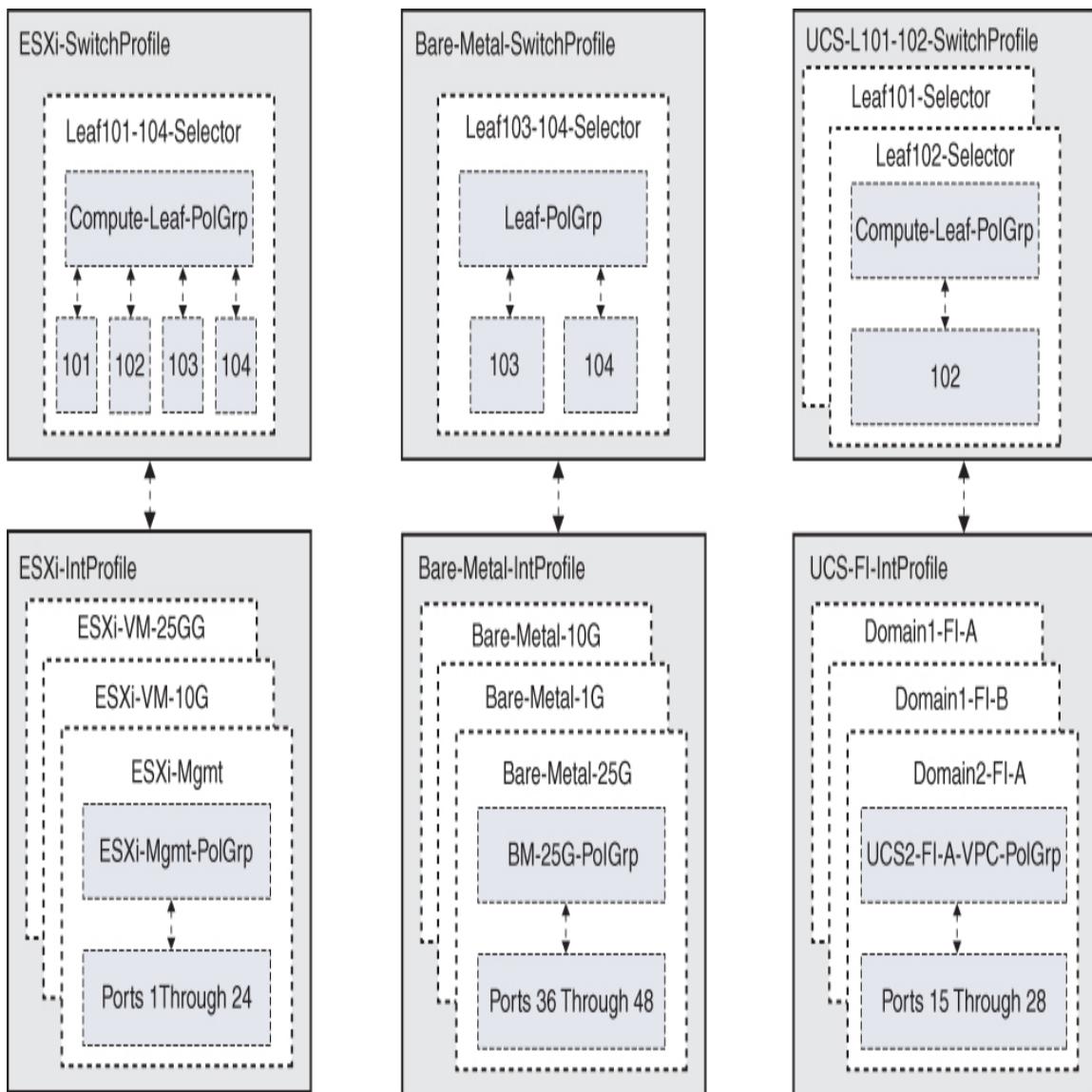
**Figure 6-13 A Sample Port Configuration Design in ACI**

The administrator has also created an interface profile named Leaf101-102-IntProfile. She makes several interface

selector mappings on the interface profile. The interface selector VPC-to-FI-B maps an VPC interface policy group called UCS-FI-B-VPC-PolGrp to Ports 11 through 14. The administrator associates the interface profile with a single switch profile named Leaf101-102-SwitchProfile, which has switch selectors referencing both node IDs 101 and 102. This configures Ports 11 through 14 on both (leaf) Node 101 and Node 102 into a vPC. The use of a switch profile referencing both vPC peers is not the only way this type of configuration can be accomplished. The administrator could have just as well mapped the VPC interface policy group to both Leaf101-SwitchProfile and Leaf102-SwitchProfile to attain the same result.

Note that in this example, all eight ports are collectively bundled into a single virtual port channel because a single vPC interface policy group has been used. If the intent were for four separate vPCs to be created, four separate vPC interface policy groups would be needed. For the eight ports to be correctly aggregated into a vPC, it is important that the switches also be configured in the same vPC domain.

[Figure 6-14](#) shows a slightly different interpretation of interface profiles compared to the example in [Figure 6-13](#). With the interpretation depicted earlier, a single interface profile needs to be created for each individual switch, and a separate interface profile needs to be created for each vPC switch pair. Such an approach enables the creation of interface profiles at the time of switch deployment; there is then no need to create interface profiles when port assignments are being made. Under the interpretation shown in [Figure 6-14](#), however, separate interface profiles may be used for each interface configuration use case.



**Figure 6-14** Example of Separate Interface Profiles for Each Interface Use Case

The interface profile presented in [Figure 6-14](#) assumes that all port assignments for all ESXi hypervisors in the network will be allocated to a single interface profile called **ESXi-IntProfile**. Multiple switch profiles reference the interface profile. With this approach, exactly the same port assignments are made on all switches whose switch profiles reference the given interface profile.

## Note

The examples presented in this section are just that: examples. There is no recommended approach for interface and switch profile design. An important benefit of profiles is their flexibility. You need to understand what an interface profile does and consider the benefits and drawbacks of any given approach and decide which approach will work best in any given environment. Although the approaches outlined are not mutually exclusive, it usually makes sense to stick to a single approach in a given environment. For example, if a defined set of switch ports will always be dedicated to a given use case across large numbers of switches and if ports can be preconfigured, it might make sense to consider an approach similar to the one outlined in [Figure 6-14](#). This also enables quick audits of all port assignments related to the specific platform for which the interface profiles were defined. On the other hand, if quick auditability of port assignments on any given switch in the environment is most important and the use of scripting is not desirable, it might be more feasible to reach this goal by using an approach similar to the one outlined in [Figure 6-13](#).

## Note

Note that even though ACI allows multiple switch profiles to reference a given switch node ID, ACI does not allow the assignment of different switch policy groups to a given switch.

[Table 6-12](#) summarizes the types of profiles and selectors covered in this section.



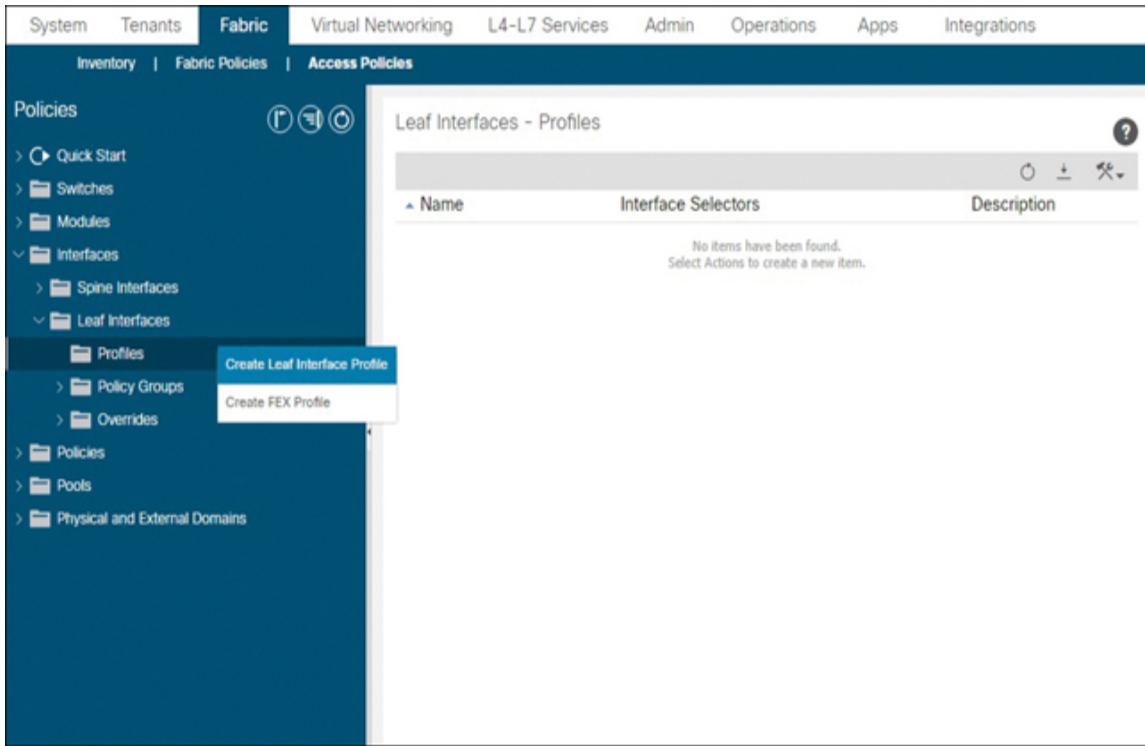
**Table 6-12** Access Policy Profiles and Selectors

O bj ec t N a m e	D efinition
<b>In te rf ac e pr o fil e</b>	An interface profile is a collection of interface mappings that gets bound to switch IDs through its association with one or more switch profiles.
<b>In te rf ac e se le</b>	An interface selector is a child object of an interface profile that ties an interface policy group to one or more port IDs. Since switch associations are determined by switch profiles and not interface profiles, interface selectors only determine port ID associations and not the list of switches to which the interface policy groups should be assigned.

<b>ct</b> <b>or</b>	
<b>S</b> <b>wi</b> <b>tc</b> <b>h</b> <b>pr</b> <b>o</b> <b>fil</b> <b>e</b>	A switch profile is a collection of switch policy group-to-node ID mappings that binds policy to switch IDs using switch selectors. Switch profiles reference interface profiles and deploy the port configurations defined in the interface profiles to switches to which the switch profile is bound. There are two types of switch profiles: leaf profiles and spine profiles.
<b>S</b> <b>wi</b> <b>tc</b> <b>h</b> <b>se</b> <b>le</b> <b>ct</b> <b>or</b>	A switch selector is a child object of a switch profile that associates a switch policy group to one or more node IDs.

## Configuring Switch Profiles and Interface Profiles

To configure an interface profile, navigate to the Access Policies menu, double-click Interfaces, open Leaf Interfaces, right-click Profiles, and select Create Leaf Interface Profile, as shown in [Figure 6-15](#).



**Figure 6-15** Navigating to the Leaf Interface Profile Creation Wizard

In the Create Leaf Interface Profile wizard, type in an object name and click Submit, as illustrated in [Figure 6-16](#). Note that interface selectors can be directly configured from within this wizard if desired.

Create Leaf Interface Profile

Name: Leaf101-102-IntProfile

Description: optional

Interface Selectors:

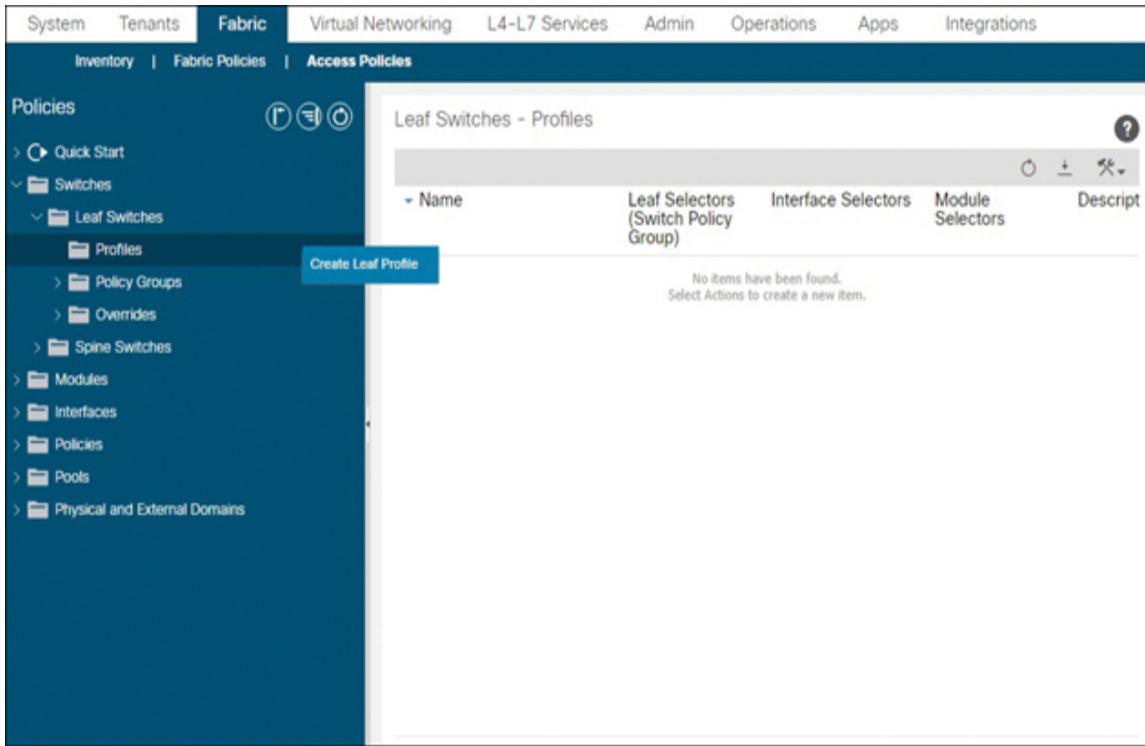
Name	Type

Cancel Submit

The dialog box is titled "Create Leaf Interface Profile". It contains three input fields: "Name" with the value "Leaf101-102-IntProfile", "Description" with the value "optional", and "Interface Selectors". Below the description field is a table with two columns: "Name" and "Type". A single row is present in the table. At the bottom right are "Cancel" and "Submit" buttons.

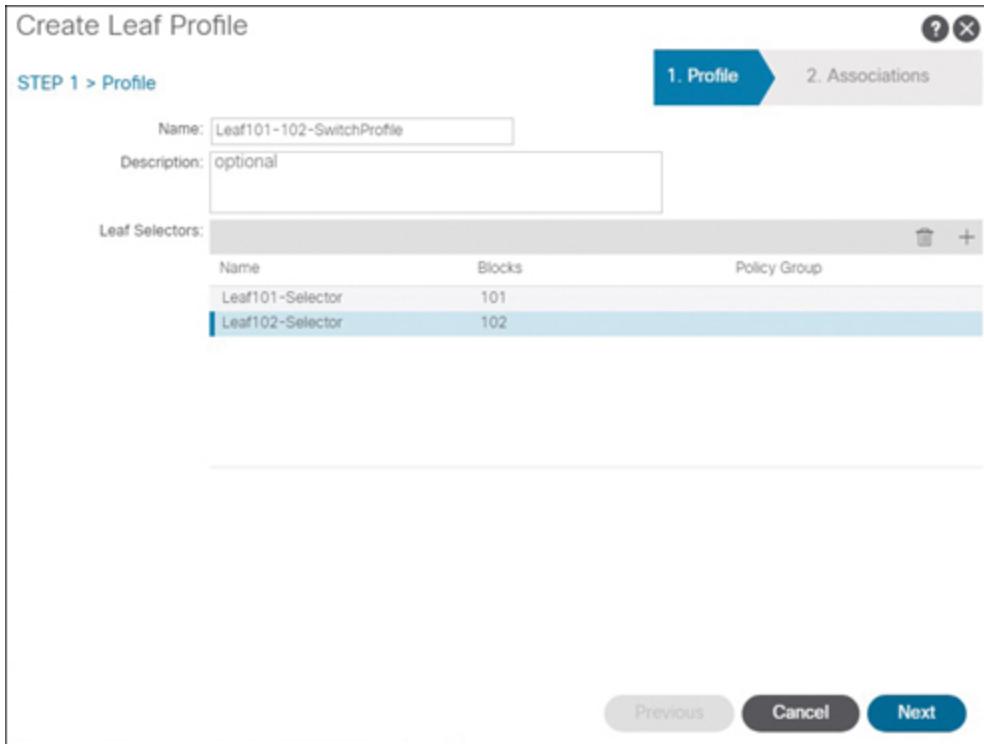
**Figure 6-16** Configuring a Leaf Interface Profile

To create a switch profile, navigate to the Access Policies menu, double-click Switches, open Leaf Switches, right-click Profiles, and select Create Leaf Profile, as shown in [Figure 6-17](#).



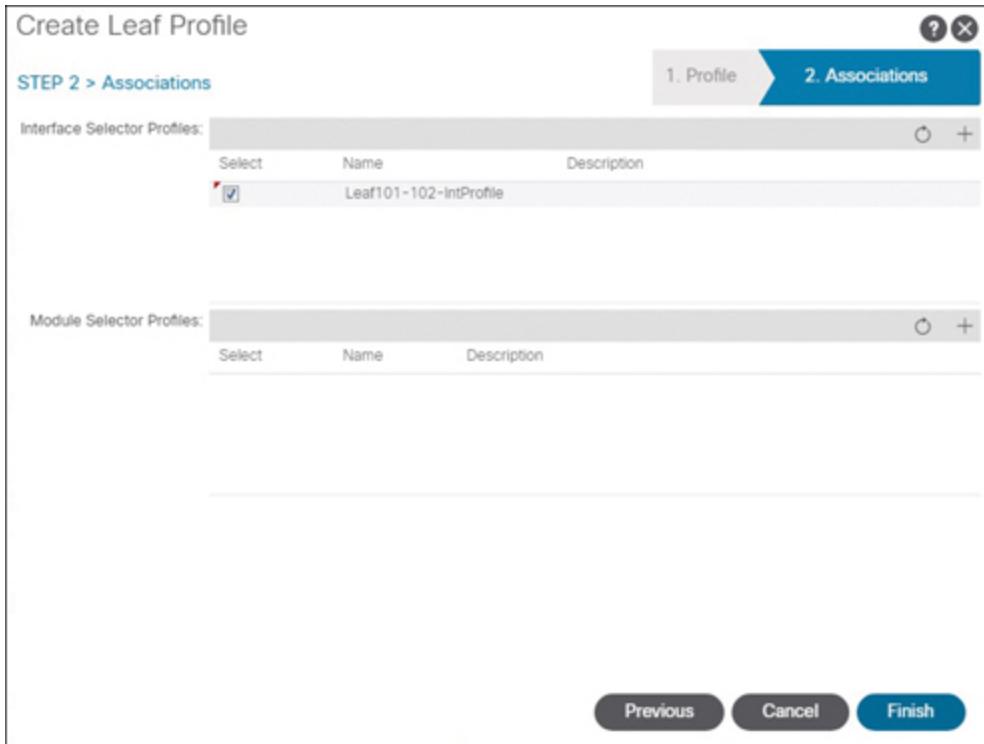
**Figure 6-17** Navigating to the Leaf Switch Profile Creation Wizard

In the Create Leaf Profile wizard, type in the switch profile name and associate the switch profile with node IDs through the configuration of switch selectors, which are shown with the label Leaf Selectors in [Figure 6-18](#). Then click Next.



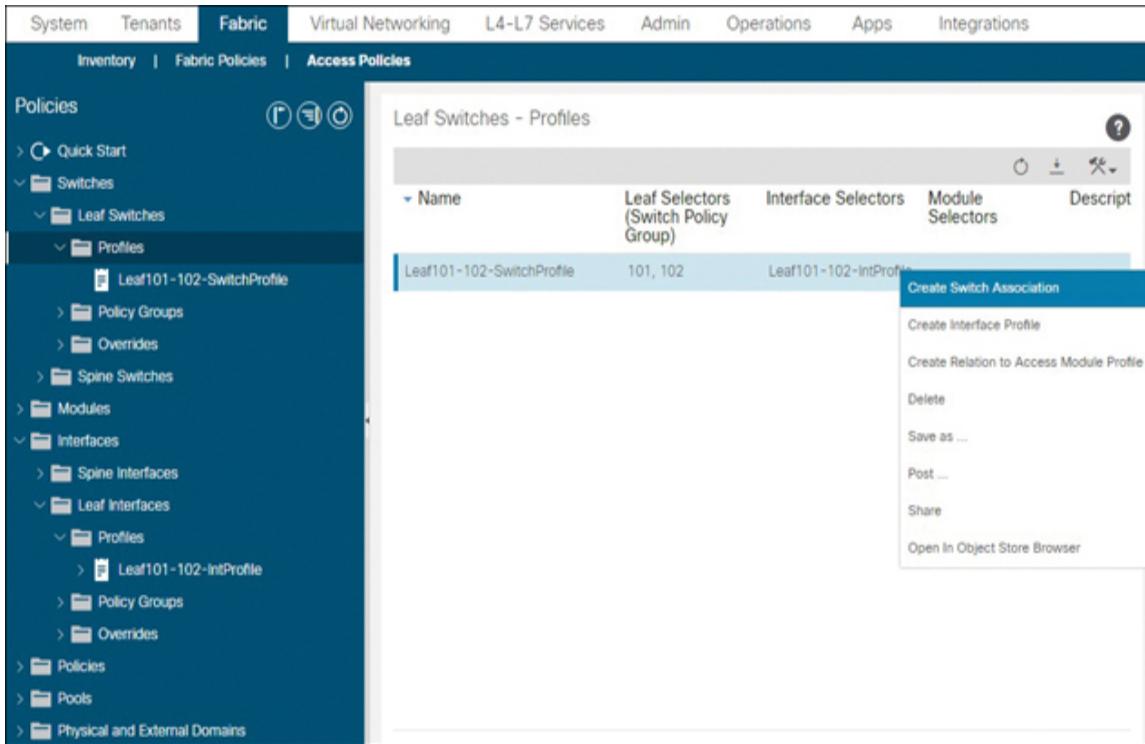
**Figure 6-18** *Associating a Switch Profile to Node IDs Using Switch Selectors*

Finally, associate interface profiles to the new switch profile. In the window displayed in [Figure 6-19](#), interface profiles are referred to as “leaf interface selectors.” Click Finish after selecting the proper interface profile(s) from the list.



**Figure 6-19** *Associating Interface Profiles to the New Switch Profile*

The screens shown in [Figures 6-15 through 6-19](#) show configuration of the objects also presented on the right side of [Figure 6-13](#). After switch profiles and interface profiles have been created, their association can be confirmed under **Fabric > Access Policies > Switches > Leaf Switches > Profiles**, as shown in [Figure 6-20](#).



**Figure 6-20** Verifying Association of Switch Profiles and Interface Profiles

These configurations can also be done via the APIC CLI, using the commands shown in [Example 6-2](#).

### **Example 6-2** CLI Equivalents for Interface and Switch Configurations

[Click here to view code image](#)

```
apic1# show running-config leaf-interface-profile Leaf101-102-IntProfile
    leaf-interface-profile Leaf101-102-IntProfile
apic1# show running-config leaf-profile Leaf101-102-SwitchProfile
    leaf-profile Leaf101-102-SwitchProfile
        leaf-group Leaf101-Selector
            leaf 101
        leaf-group Leaf102-Selector
```

```
leaf 102
exit
leaf-interface-profile Leaf101-102-IntProfile
```

## Stateless Networking in ACI

The approach of using node IDs, switch profiles, and interface profiles and not tying configurations to physical hardware is called *stateless networking*.

Stateless networking has the benefit of minimizing the time to recover from hardware issues. Sometimes, it also enables expedited data center network migrations.

If a switch needs to be returned to Cisco due to hardware issues, an administrator can easily migrate the switch configurations to a new switch by decommissioning the switch from the Fabric Membership view and commissioning a replacement switch using the previous node ID. All old optics can be reseated into the same port IDs to which they were earlier attached. Cables can then be connected to the previously assigned ports.

Alternatively, switch profiles assigned to the leaf in question can be assigned to a new node ID, and all port configurations carry over to the new node ID, ensuring that administrators do not have to modify port configurations. There are caveats to this approach when virtual port channels are deployed, but a strategy can almost always be identified to expedite the overall process.

If a data center network migration needs to take place and an ACI fabric needs to be upgraded to new hardware, it is most likely that a process can be identified to allow for re-allocation of switch profiles to new node IDs or re-

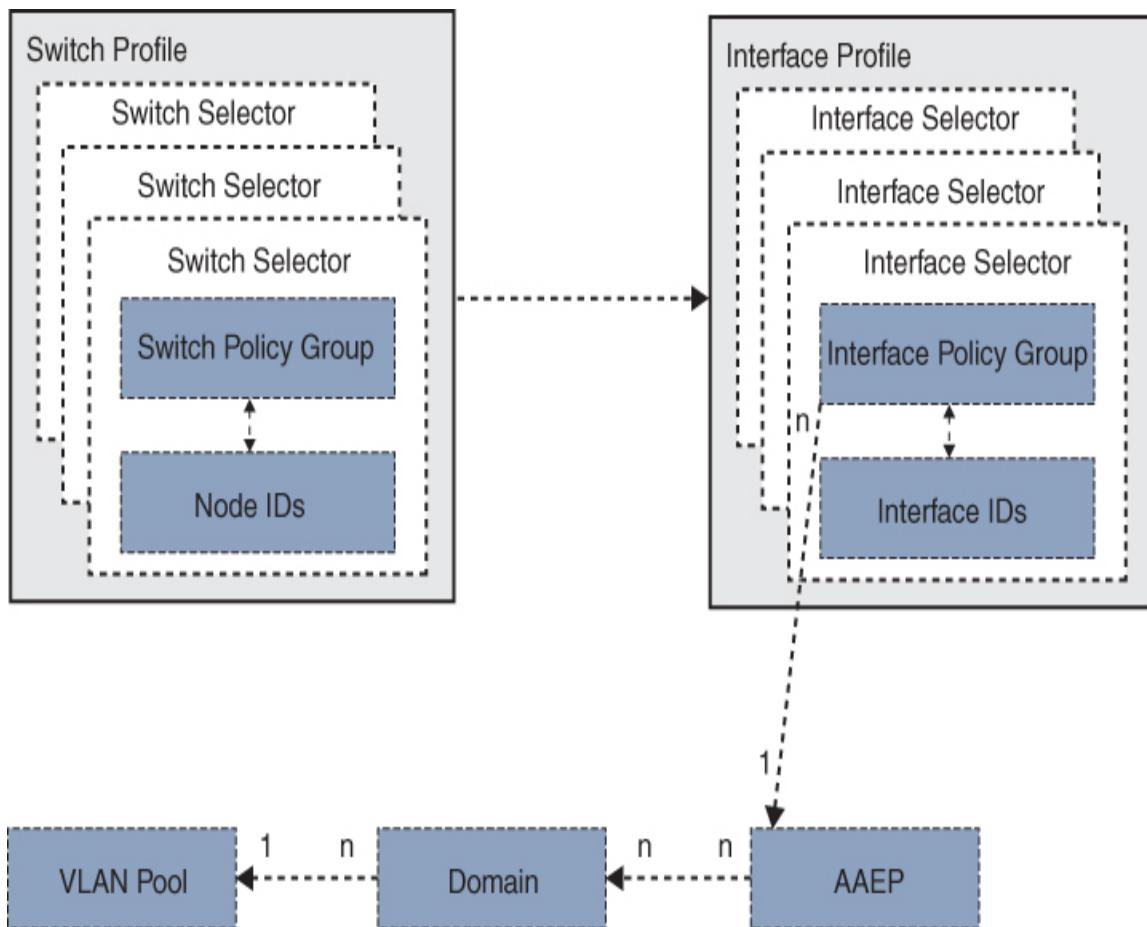
association of interface profiles to new switch profiles to speed up the migration process.

## Bringing It All Together

When learning new concepts, it sometimes helps to have a visual summary of the concepts to aid in learning. This section provides such a visual aid and also aims to draw a bigger picture of how the concepts in this chapter and [Chapter 5, “Tenant Building Blocks,”](#) relate to one another.

## Access Policies Hierarchy in Review

[Figure 6-21](#) provides a visual representation of the objects covered in this chapter and how they relate with one another.



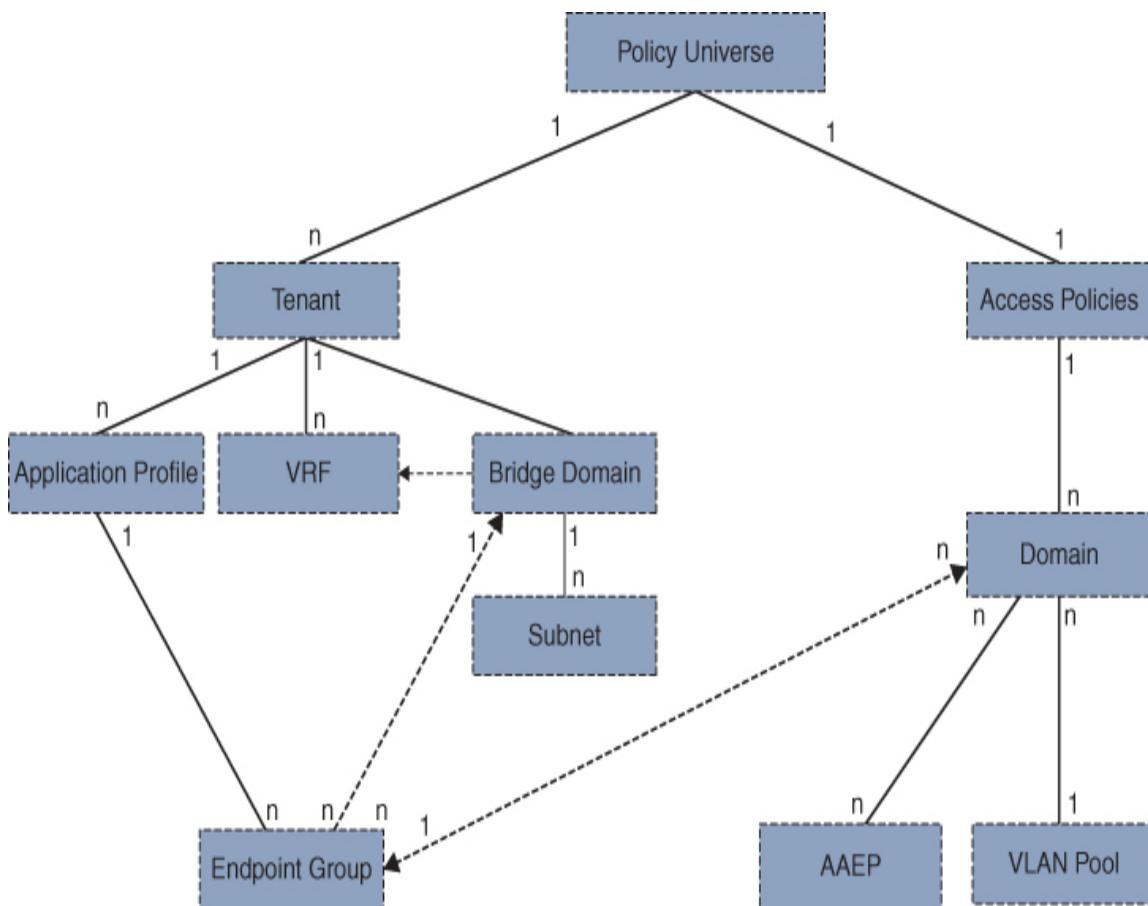
**Figure 6-21** Relationship Between Key Access Policy Objects

Keep in mind that administrators are not limited in the number of times they can instantiate any of the types of objects depicted in [Figure 6-21](#). The one-to-one relationships and one-to-many relationships shown only apply to the relationships between these subobjects in the access policies hierarchy.

One of the key things to remember from this diagram is that multiple interface policy groups can reference a single AAEP. However, any given interface policy group can reference one and only one AAEP. Also remember that a domain can reference no more than one VLAN pool, even though multiple domains can technically share a VLAN pool.

# Access Policies and Tenancy in Review

Figure 6-22 summarizes the critical relationships between the access policies and tenant logical policy model subtrees. It illustrates how domains serve as the central object that tenant administrators use to map EPGs to the underlying physical infrastructure.



**Figure 6-22** How Tenancy Links to Underlying Physical Infrastructure

Figures 6-21 and 6-22 together show some of the most important objects that engineers deal with on a day-to-day basis in ACI. More often than not, when port configuration issues occur, the objects in these two figures and the

configured relationships between these objects should be evaluated first.

## Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: [Chapter 17, “Final Preparation,”](#) and the exam simulation questions in the Pearson Test Prep Software Online.

## Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. [Table 6-13](#) lists these key topics and the page number on which each is found.



**Table 6-13** Key Topics for [Chapter 6](#)

Key Topic Description Element		Page Number
Paragraph	Defines VLAN pools	159
Paragraph	Describes static and dynamic VLAN allocation	159

Paragraph	Describes domains	160
List	Outlines the types of domains in ACI	161
Paragraph	Touches on challenges related to VLAN pool overlap across multiple domains	164
Paragraph	Defines AAEPs	165
Paragraph	Defines interface policies	169
Paragraph	Defines interface policy groups	169
Table 6-6	Describes the types of interface policy groups in ACI	170
Paragraph	Calls out which types of interface policy groups are fully reusable	171
Table 6-7	Describes the types of interface policies available in ACI	171

Paragraph	Defines switch policies and switch policy groups	<a href="#">174</a>
<a href="#">Table 6-10</a>	Describes the most commonly deployed switch policies in ACI	<a href="#">174</a>
<a href="#">Table 6-12</a>	Describes access policy profiles and selectors	<a href="#">179</a>

## Complete Tables and Lists from Memory

Print a copy of [Appendix C, “Memory Tables”](#) (found on the companion website), or at least the section for this chapter, and complete the tables and lists from memory. [Appendix D, “Memory Tables Answer Key”](#) (also on the companion website), includes completed tables and lists you can use to check your work.

## Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

- [VLAN pool](#)
- [static VLAN allocation](#)
- [dynamic VLAN allocation](#)
- [domain](#)
- [physical domain](#)
- [external bridge domain](#)

external routed domain  
Virtual Machine Manager (VMM) domain  
attachable access entity profile (AAEP)  
interface policy  
MisCabling Protocol (MCP)  
interface policy group  
interface profile  
interface selector  
switch profile  
switch selector  
leaf selector  
spine selector

## Chapter 7

# Implementing Access Policies

**This chapter covers the following topics:**

**Configuring ACI Switch Ports:** This section addresses practical implementation of ACI switch port configurations.

**Configuring Access Policies Using Quick Start Wizards:** This section shows how to configure access policies using quick start wizards.

**Additional Access Policy Configurations:** This section reviews implementation procedures for a handful of other less common access policies.

This chapter covers the following exam topics:

- 1.5 Implement ACI policies
  - 1.5.a access
  - 1.5.b fabric

Chapter 6, “Access Policies,” covers the theory around access policies and the configuration of a limited number of objects available under the Access Policies menu. This chapter completes the topic of access policies by covering the configuration of all forms of Ethernet-based switch port connectivity available in ACI.

## “Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 7-1 lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find

the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Questions.”

**Table 7-1** “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Configuring ACI Switch Ports	1–5
Configuring Access Policies Using Quick Start Wizards	6
Additional Access Policy Configurations	7–10

### Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** An administrator has configured a leaf interface, but it appears to have the status out-of-service. What does this mean?
  - a.** The port has a bad transceiver installed.
  - b.** The server behind the port has failed to PXE boot, and the port has been shut down.
  - c.** This status reflects the fact that access policies have been successfully deployed.
  - d.** The port has been administratively disabled.
- 2.** Where would you go to configure a vPC domain in ACI?
  - a.** Fabric > Access Policies > Policies > Switch > Virtual Port Channel default
  - b.** Fabric > Access Policies > Interfaces > Leaf Interfaces > Policy Groups

- c. Fabric > Access Policies > Policies > Switch > VPC Domain
  - d. Fabric > Fabric Policies > Policies > Switch > Virtual Port Channel default
- 3. True or false: To configure an LACP port channel, first create a leaf access port policy group and then add a port channel policy to the interface policy group.
  - a. True
  - b. False
- 4. True or false: To forward traffic destined to an endpoint behind a vPC, switches within the fabric encapsulate each packet twice and forward a copy separately to the loopback 0 tunnel endpoint of each vPC peer.
  - a. True
  - b. False
- 5. True or false: The only way to enable CDP in ACI is through the use of interface overrides.
  - a. True
  - b. False
- 6. True or false: The Configure Interface wizard in ACI can be used to make new port assignments using preconfigured interface policy groups.
  - a. True
  - b. False
- 7. True or false: To configure a fabric extender (FEX), you first create a FEX profile and then configure an access port selector from the parent leaf down to the FEX with the Connected to FEX checkbox enabled.
  - a. True
  - b. False
- 8. Which of the following are valid steps in implementing MCP on all 20 VLANs on a switch? (Choose all that apply.)
  - a. Enable MCP at the switch level.
  - b. Ensure that MCP has been enabled on all desired interfaces through interface policies.
  - c. Select the Enable MCP PDU per VLAN checkbox.
  - d. Enable MCP globally by toggling the Admin State to Enabled and defining a key.

- 9.** True or false: With dynamic port breakouts, a port speed can be lowered, but a dramatic loss occurs in the forwarding capacity of the switch.
- a. True
  - b. False
- 10.** True or false: ACI preserves dot1q CoS bits within packets by default.
- a. True
  - b. False

## Foundation Topics

### Configuring ACI Switch Ports

Put yourself in the shoes of an engineer working at a company that has decided to deploy all new applications into ACI. Looking at a platform with an initial focus on greenfield deployments as opposed to the intricacies of migrations can often lead to better logical designs that fully leverage the capabilities of the solution.

Imagine as part of this exercise that you have been asked to accommodate a newly formed business unit within your company, focusing on multiplayer gaming. This business unit would like to be able to patch its server operating systems independently and outside of regular IT processes and to have full autonomy over its applications with close to zero IT oversight beyond coordination of basic security policies. The business unit thinks it can achieve better agility if it is not bound by processes dictated by IT. Aside from whether deploying a shadow environment alongside a production environment is even desirable, is a setup like this even feasible with ACI? By thinking about this question while reading through the following sections, you may gain insights into how access policies can be used to share underlying infrastructure among tenants in ACI.

### Configuring Individual Ports

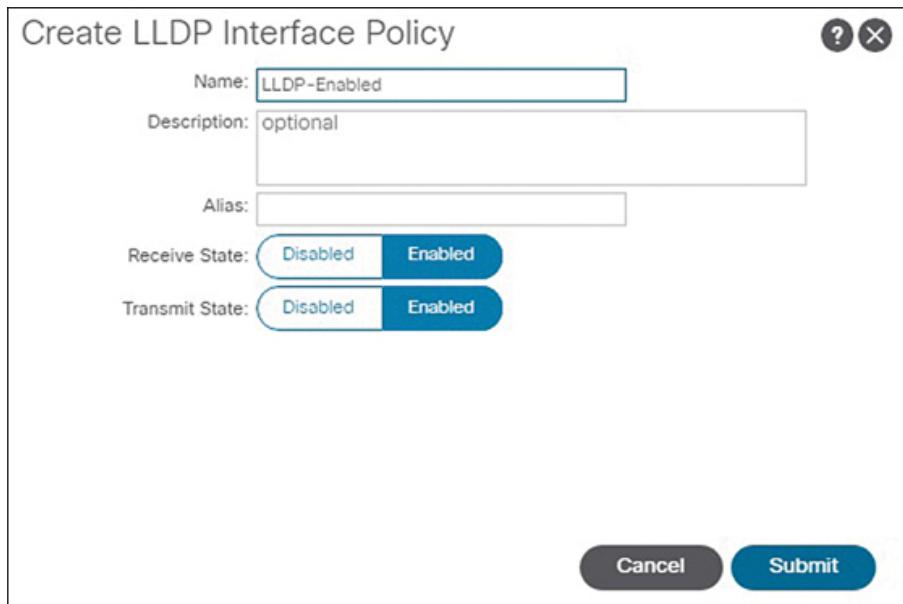
This section shows how to deploy access policies for two new multiplayer gaming servers. Assume that each of these new servers has a single 10 Gbps network card and does not support port channeling. Let's say that the network engineers configuring switch ports for connectivity to these servers want to enable LLDP and CDP to have visibility into host names, if advertised by the servers. They also decide to auto-detect speed and

duplex settings to reduce the need for their team to have to coordinate network card upgrades with the business unit.

### Note

This chapter demonstrates a wide variety of common port configurations through examples. The examples are not meant to imply that implementation of auto-negotiation, LLDP, and CDP toward servers outside an organization's administrative control is a best practice. Where the intent is to convey that something is a best practice, this book explicitly says so.

To configure an interface policy with LLDP enabled, navigate to **Fabric > Access Policies > Policies > Interface**, right-click LLDP Interface, and select Create LLDP Interface Policy. [Figure 7-1](#) shows an interface policy with LLDP enabled bidirectionally.



**Figure 7-1** Configuring an LLDP Interface Policy

It is often good practice to use explicit policies. Auto-negotiation of port speed and duplex settings can be achieved by using a link level policy. To create a link level policy, navigate to **Fabric > Access Policies > Policies > Interface**, right-click Link Level, and select Create Link Level Policy.

Key Topic

Figure 7-2 shows the settings for a link level policy. By default, Speed is set to Inherit, and Auto Negotiation is set to On to allow the link speed to be determined by the transceiver, medium, and capabilities of the connecting server. The **Link Debounce Interval** setting delays reporting of a link-down event to the switch supervisor. The Forwarding Error Correction (FEC) setting determines the error correction technique used to detect and correct errors in transmitted data without the need for data retransmission.

The screenshot shows a configuration dialog titled "Create Link Level Policy". The fields are as follows:

- Name: Speed-Auto
- Description: optional
- Alias: (empty)
- Auto Negotiation: **on** (selected)
- Speed: inherit
- Link debounce interval (msec): 100
- Forwarding Error Correction: Inherit

At the bottom right are "Cancel" and "Submit" buttons.

Key  
Topic

**Figure 7-2** Configuring a Link Level Interface Policy

To create a policy with CDP enabled, navigate to **Fabric > Access Policies > Policies > Interface**, right-click CDP Interface, and select CDP Interface Policy. Figure 7-3 shows an interface policy with CDP enabled.

Create CDP Interface Policy

Name: CDP-Enabled

Description: optional

Alias:

Admin State:  Enabled  Disabled

**Figure 7-3** Configuring a CDP Interface Policy

In addition to interface policies, interface policy groups need to reference a global access policy (an AAEP) for interface deployment. AAEPs can often be reused. [Figure 7-4](#) shows the creation of an AAEP named Bare-Metal-Servers-AAEP. By associating the domain phys as shown in [Figure 7-4](#), you enable any servers configured with the noted AAEP to map EPGs to switch ports using VLAN IDs 300 through 499.

Create Attachable Access Entity Profile

STEP 1 > Profile

1. Profile 2. Association To Interfaces

Name: Bare-Metal-Servers-AAEP

Description: optional

Enable Infrastructure VLAN:

Domains (VMM, Physical or External) To Be Associated	To Interfaces:	Domain Profile	Encapsulation
		Physical Domain - phys	from:vlan-300 to:vlan-499

EPG DEPLOYMENT (Selected EPGs will be displayed on all the interfaces associated.)

Application EPGs	Encap	Primary Encap	Mode

**Figure 7-4** Configuring an AAEP

With interface policies and global policies created, it is time to create an interface policy group to be applied to ports.



To create an interface policy group for individual (non-aggregated) switch ports, navigate to **Fabric > Access Policies > Interfaces > Leaf Interfaces > Policy Groups**, right-click the Leaf Access Port option, and select Create Leaf Access Port Policy Group.

Figure 7-5 shows the association of the interface policies and AAEP created earlier with an interface policy group. Because policy groups for individual ports are fully reusable, a generic name not associated with any one server might be most beneficial for the interface policy group.

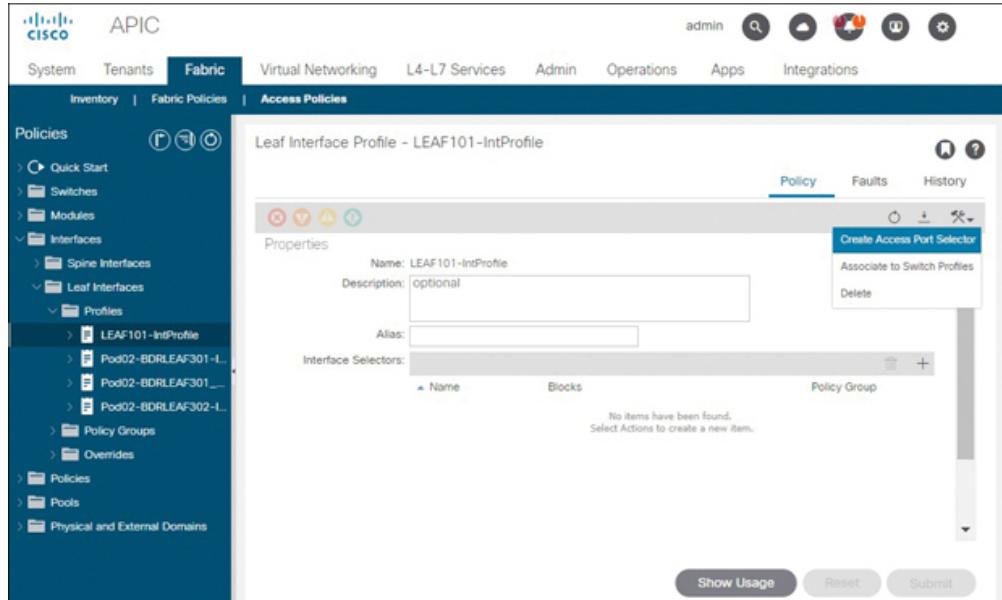
The screenshot shows a configuration dialog titled "Create Leaf Access Port Policy Group". The "Name" field contains "Multiplayer-Gaming-PolGrp". The "Attached Entity Profile" dropdown is set to "Bare-Metal-Servers-A". The "Submit" button is located at the bottom right of the dialog.



**Figure 7-5 Configuring a Leaf Access Port Policy Group**

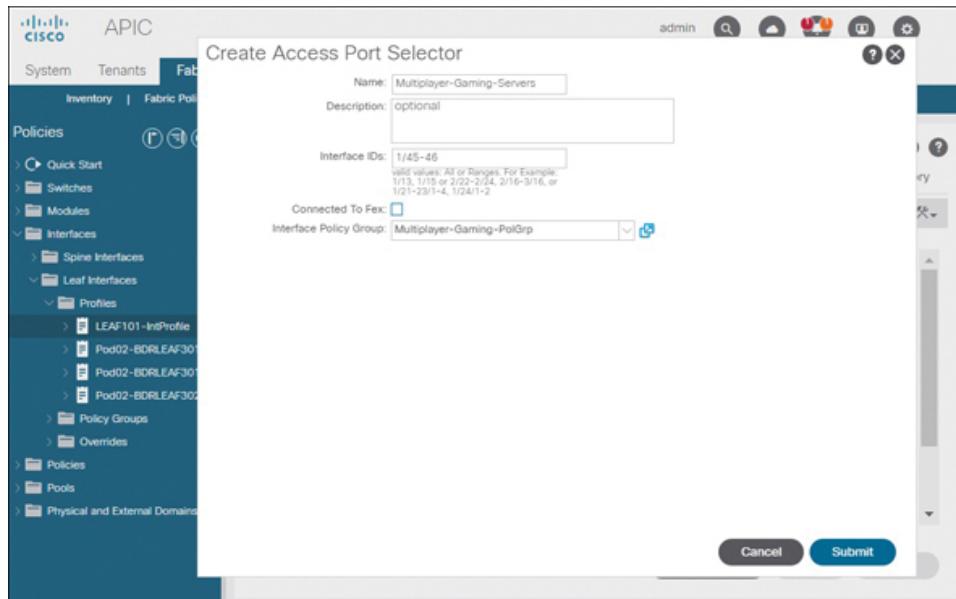
Next, the interface policy group needs to be mapped to switch ports. Let's say a new switch has been procured and will be dedicated to multiplayer gaming servers for the business unit. The switch, which has already been commissioned, has node ID 101 and a switch profile. An interface profile has also been linked with the switch profile.

To associate an interface policy with ports, navigate to the desired interface profile, click on the Tools menu, and select Create Access Port Selector, as shown in [Figure 7-6](#).



**Figure 7-6** Navigating to the Create Access Port Selector Window

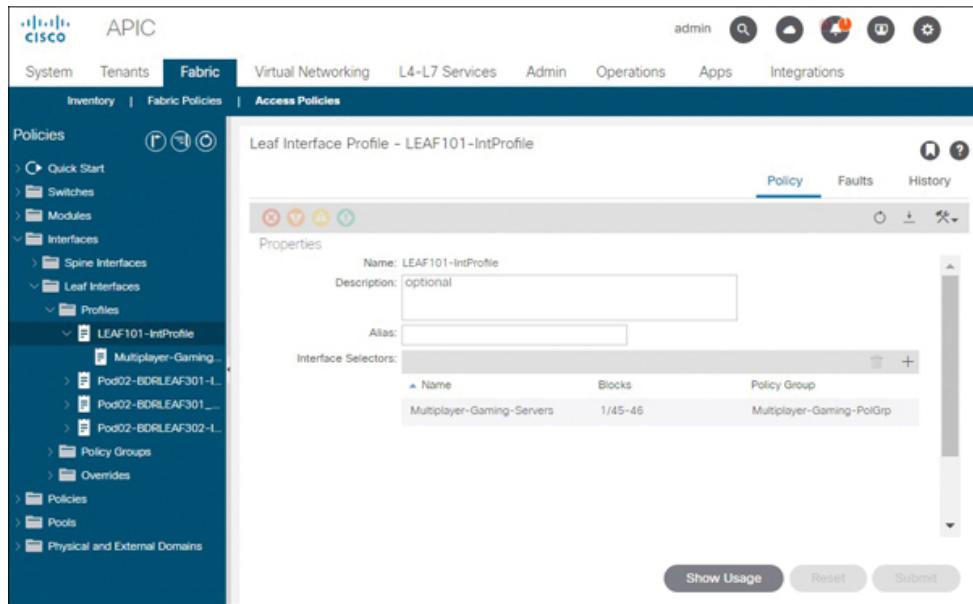
[Figure 7-7](#) demonstrates the association of the new interface policy group with ports 1/45 and 1/46. Since this is a contiguous block of ports, you can use a hyphen to list the ports. After you click Submit, the interface policy group is deployed on the selected switch ports on all switches referenced by the interface profile.



**Key Topic**

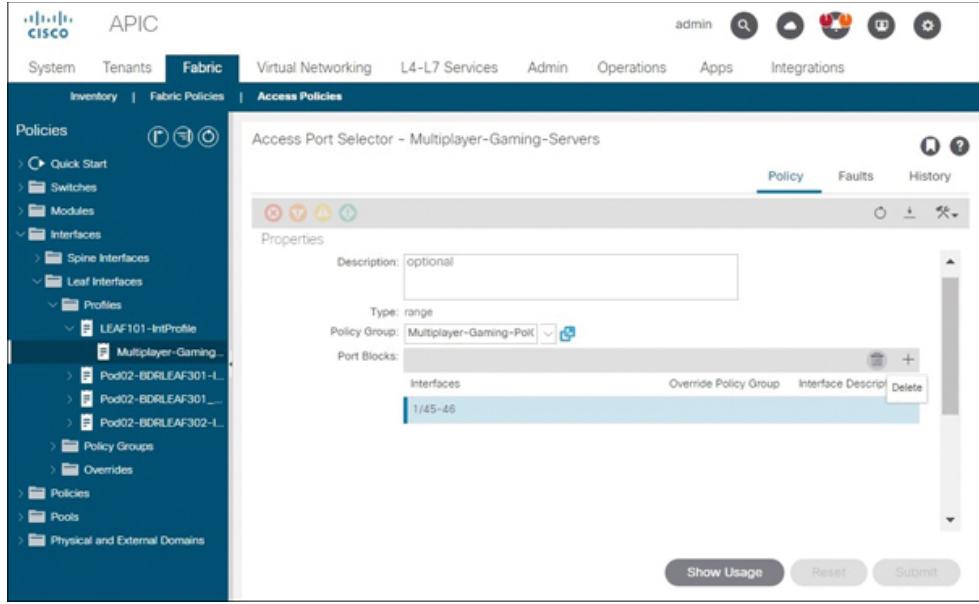
**Figure 7-7 Mapping Ports to an Interface Policy Group**

Back under the leaf interface profile, notice that an entry should be added in the Interface Selectors view (see [Figure 7-8](#)).



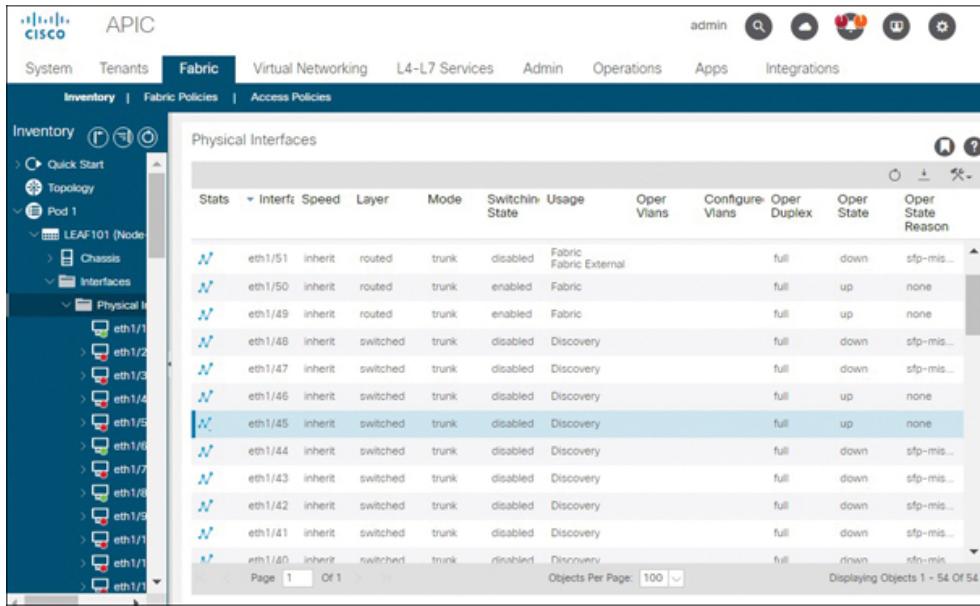
**Figure 7-8 Port Mappings Added to the Interface Selector View**

Double-click the entry to view the Access Port Selector page. As shown in [Figure 7-9](#), ports that are mapped to an interface policy group as a contiguous block cannot be individually deleted from the port block. This might pose a problem if a single port that is part of a port block needs to be deleted and repurposed at some point in the future. Therefore, use of hyphens to group ports together is not always suitable.



**Figure 7-9 Ports Lumped Together in a Port Block**

In the GUI, the operational state of ports can be verified under **Fabric > Inventory > Pod number > Node Name > Interfaces > Physical Interfaces**. According to [Figure 7-10](#), the newly configured ports appear to have the Usage column set to Discovery.



**Figure 7-10 Verifying the Status of Physical Interfaces in the ACI GUI**

[Example 7-1](#) shows how to verify the operational status of ports in the switch CLI.

## **Example 7-1 Verifying Port Status via the ACI Switch CLI**

[Click here to view code image](#)

```
LEAF101# show interface ethernet 1/45-46 status
-----
-----
Port      Name       Status     Vlan      Duplex   Speed    Type
-----
Eth1/45    --        out-of-ser trunk    full     10G     10Gbase-SR
Eth1/46    --        out-of-ser trunk    full     10G     10Gbase-SR
```

What does the status “out-of-service” actually mean? When this status appears for operational fabric downlink ports, it simply means that tenant policies have not yet been layered on top of the configured access policies.

[Table 7-2](#) summarizes port usage types that may appear in the GUI.

**Table 7-2** Port Usages

Po	Description
rt	
Us	
ag	
e	
Blacklist	Blacklist indicates that a port has been disabled either by an administrator or by an APIC having detected anomalies with the port. Anomalies can include wiring errors or switches with nonmatching fabric IDs connecting to the fabric.
Controller	ACI detects an APIC controller attached to the port.

Po	Description
rt	
Us	
ag	
e	
Dis	The port is not forwarding user traffic because no tenant policies have been enabled over the port. This can be due to the lack of an EPG mapping or routing configuration on the port. This is the default state for all fabric downlinks.
EP G	At least one EPG has been correctly associated with the port. This is a valid state even if the port is disabled.
Fab ric	Fabric indicates that a port functions or can potentially function as a fabric uplink for connectivity between leaf and spine switches. By default, fabric ports have Usage set to both Fabric and Fabric External until cabling is attached or a configuration change takes place.
Fab ric Ext al	Fabric External indicates that a port functions as an L3Out, peering with some switch or router outside the fabric. By default, fabric ports have Usage set to both Fabric and Fabric External until cabling is attached or a configuration change takes place.
Infr a	Infra indicates that a port is trunking the overlay VLAN.

The APIC CLI commands shown in [Example 7-2](#) are the equivalent of the configurations completed via the GUI. Notice that the LLDP setting does not appear in the output. This is because not all commands appear in the output of the APIC CLI running configuration. Use the command **show running-config all** to see all policy settings, including those that deviate from default values for a parameter.

**Example 7-2 APIC CLI Configurations Equivalent to the GUI Configurations Demonstrated**

[Click here to view code image](#)

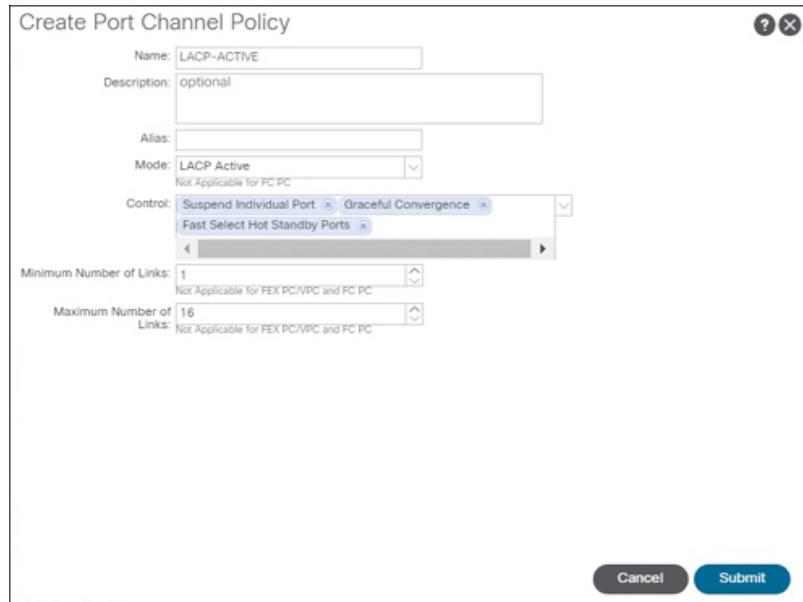
```
APIC1# show run
(...output truncated for brevity...)
template policy-group Multiplayer-Gaming-PolGrp
    cdp enable
    vlan-domain member phys type phys
    exit
leaf-interface-profile LEAF101-IntProfile
    leaf-interface-group Multiplayer-Gaming-Servers
        interface ethernet 1/45-46
        policy-group Multiplayer-Gaming-PolGrp
        exit
    exit
```

**Note**

Switch port configurations, like all other configurations in ACI, can be scripted or automated using Python, Ansible, Postman, or Terraform or using workflow orchestration solutions such as UCS Director.

## Configuring Port Channels

Let's say that the business unit running the multiplayer project wants a server deployed using LACP, but it has purchased only a single leaf switch, so dual-homing the server to a pair of leaf switches is not an option. Before LACP port channels can be deployed in ACI, you need to configure an interface policy with LACP enabled. To do so, navigate to **Fabric > Access Policies > Policies > Interface**, right-click Port Channel, and select Create Port Channel Policy. The window shown in [Figure 7-11](#) appears.



**Figure 7-11** Configuring a Port Channel Interface Policy with LACP Enabled

The function of the Mode setting LACP Active should be easy to understand. [Table 7-3](#) details the most commonly used Control settings available for ACI port channels.

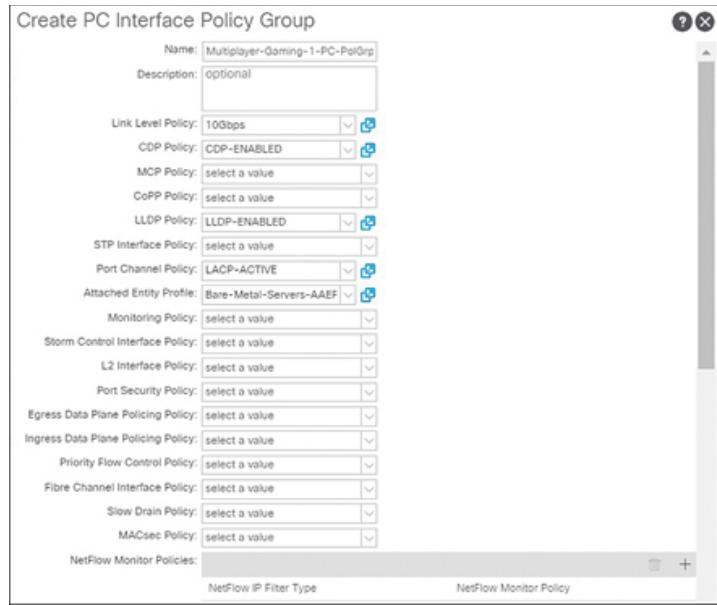


Table 7-3 Common Control Settings for ACI Port Channel Configuration

Cont	Description
rol	
Setti	
ng	

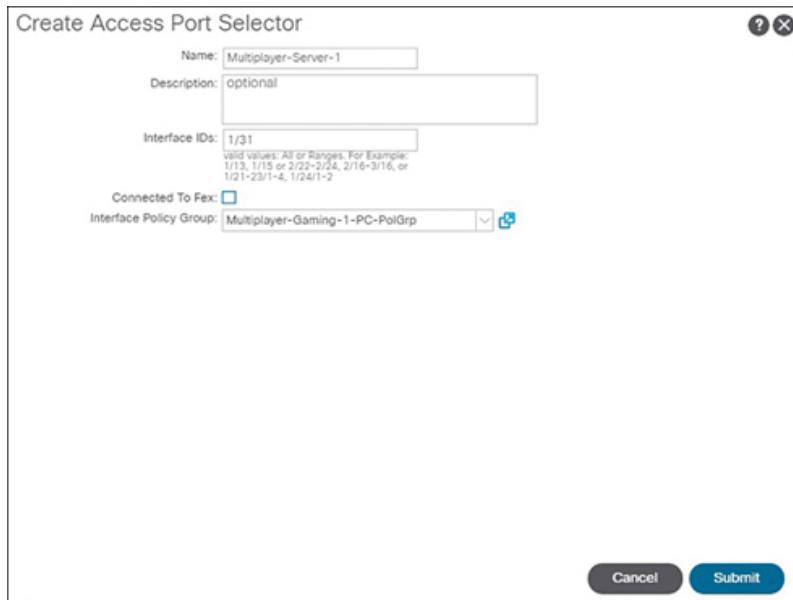
Cont rol Setti ng	Description
Fast Select Hot Standby Ports	This setting enables fast select for hot standby ports. Enabling this feature makes possible the faster selection of a hot standby port when the last active port in the port channel is going down.
Graceful Conversion	This setting ensures optimal failover of links in an LACP port channel if the port channel or virtual port channel configured with this setting connects to Nexus devices.
Suspension and Individual Port	With this setting configured, LACP suspends a bundled port if it does not receive LACP packets from its peer port. When this setting is not enabled, LACP moves such ports into the Individual state.
Symmetric Hashing	With this setting enabled, bidirectional traffic is forced to use the same physical interface, and each physical interface in the port channel is effectively mapped to a set of flows. When an administrator creates a policy with Symmetric Hashing enabled, ACI exposes a new field for selection of a hashing algorithm.

After you create a port channel interface policy, you can create a port channel interface policy group for each individual port channel by navigating to **Fabric > Access Policies > Policies > Interface > Leaf Interfaces > Policy Groups**, right-clicking PC Interface, and selecting Create PC Interface Policy Group. [Figure 7-12](#) shows the grouping of several policies to create a basic port channel interface policy group.



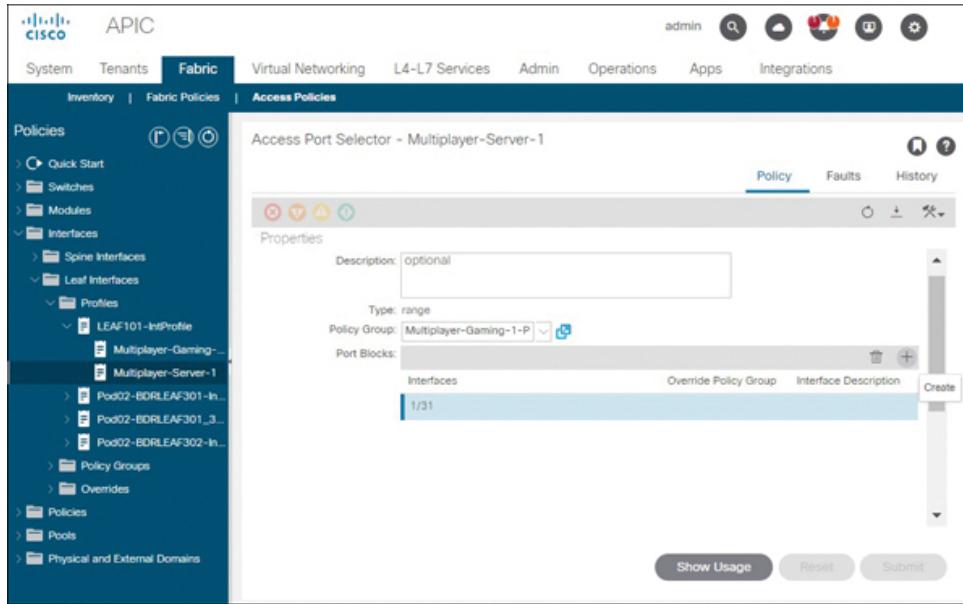
**Figure 7-12** Configuring a Port Channel Interface Policy Group

You use an access selector to associate the interface policy group with the desired ports. If the intent is to configure ports 1/31 and 1/32 without lumping these ports into a single port block, it might make sense to first associate a single port with the port channel interface policy group and then add the next port as a separate port block. [Figure 7-13](#) demonstrates the association of port 1/31 on Leaf 101 with the interface policy group.



**Figure 7-13** Mapping Ports to a Port Channel Interface Policy Group

To add the second port to the port channel, click on the + sign in the Port Blocks section, as shown in [Figure 7-14](#), to create a new port block.



**Figure 7-14** Navigating to the Create Access Port Block Page

Finally, you can add port 1/32 as a new port block, as shown in [Figure 7-15](#).

A screenshot of a 'Create Access Port Block' dialog box. It has fields for 'Interface IDs' containing '1/32', 'Description' with 'optional', and 'Override Policy Group' with 'select an option'. There are 'Cancel' and 'Submit' buttons at the bottom.

**Figure 7-15** Adding a New Port Block to an Access Port Selector

[Example 7-3](#), taken from the Leaf 101 CLI, verifies that Ethernet ports 1/31 and 1/32 have indeed been bundled into an LACP port channel and that

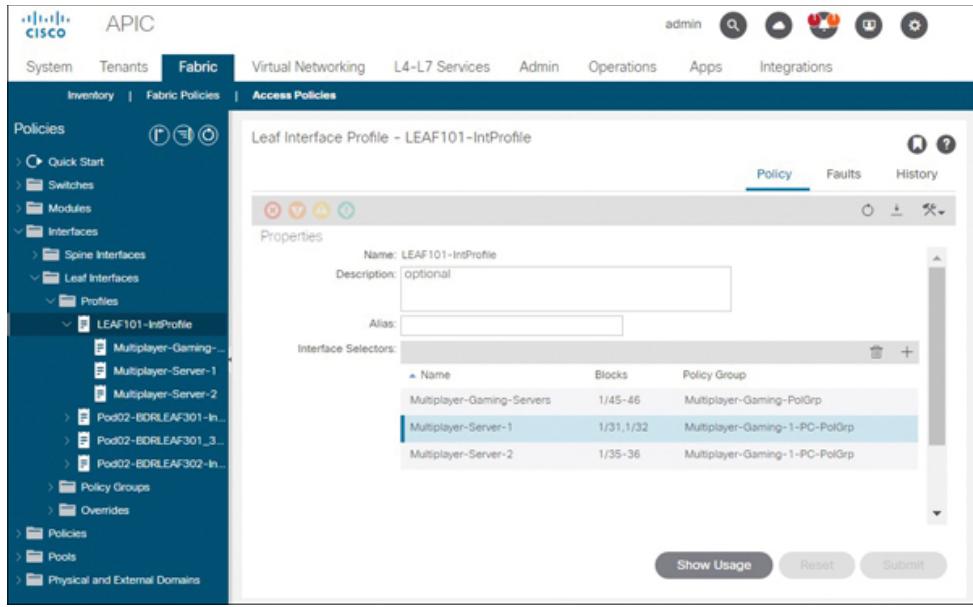
they are up. Why was there no need to assign an ID to the port channel? The answer is that ACI itself assigns port channel IDs to interface bundles.

### **Example 7-3 Switch CLI-Based Verification of Port Channel Configuration**

[Click here to view code image](#)

```
LEAF101# show port-channel summary
Flags:  D - Down          P - Up in port-channel (members)
        I - Individual    H - Hot-standby (LACP only)
        S - Suspended      R - Module-removed
        S - Switched       R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
        F - Configuration failed
-----
-- 
Group Port-      Type     Protocol   Member Ports
      Channel
-----
-- 
1    Po1(SU)      Eth      LACP       Eth1/6(P)    Eth1/8(P)
2    Po2(SU)      Eth      LACP       Eth1/31(P)   Eth1/32(P)
```

You have already learned that port channel interface policy groups should ideally not be reused, especially on a single switch. But why is this the case? [Figure 7-16](#) shows that an administrator has created a new interface selector and has mistakenly associated the same port channel interface policy group with ports 1/35 and 1/36. Note in this figure that using commas to separate the interface IDs leads to the creation of separate port blocks.



**Figure 7-16** Multiple Interface Selectors Referencing a Port Channel Interface Policy Group

The setup in [Figure 7-16](#) would lead to the switch CLI output presented in [Example 7-4](#).

#### **Example 7-4** Interfaces Bundled Incorrectly Due to PC Interface Policy Group Reuse

[Click here to view code image](#)

```
LEAF101# show port-channel summary
Flags:  D - Down          P - Up in port-channel (members)
        I - Individual    H - Hot-standby (LACP only)
        s - Suspended     r - Module-removed
        S - Switched      R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
        F - Configuration failed
-
-
Group Port-          Type       Protocol      Member Ports
      Channel
-
-
1    Po1(SU)         Eth        LACP          Eth1/6(P)      Eth1/8(P)
2    Po2(SU)         Eth        LACP          Eth1/31(P)     Eth1/32(P)
```

Eth1/35(D)

Eth1/36(D)

To the administrator's surprise, ports 1/35 and 1/36 have been added to the previously created port channel. The initial assumption may have been that because a different interface selector name was selected, a new port channel would be created. This is not the case.

[Example 7-5](#) shows the CLI-equivalent configuration of the port channel interface policy group and the assignment of the policy group to ports on Leaf 101.

#### **Example 7-5 APIC CLI Configuration for the Port Channel Interfaces**

[Click here to view code image](#)

```
template port-channel Multiplayer-Gaming-1-PC-PolGrp
    cdp enable
    vlan-domain member phys type phys
    channel-mode active
    speed 10G
    no negotiate auto
    exit
    leaf-interface-profile LEAF101-IntProfile
    leaf-interface-group Multiplayer-Server-1
        interface ethernet 1/31
        interface ethernet 1/32
        channel-group Multiplayer-Gaming-1-PC-PolGrp
        exit
    exit
```

#### **Key Topic**

There is nothing that says you cannot reuse port channel or virtual port channel interface policy groups in new interface selector configurations if the intent truly is to bundle the new interfaces into a previously created port channel or virtual port channel. You may still question whether a port channel interface policy group or a vPC interface policy group can be reused on a different switch or vPC domain. As a best practice, you should avoid reuse of port channel and vPC interface policy groups when creating new port channels and vPCs to minimize the possibility of configuration mistakes.

### Note

You may not have noticed it, but the Control settings selected in the port channel interface policy shown earlier are Suspend Individual Ports, Graceful Convergence, and Fast Select Hot Standby Ports (refer to [Figure 7.11](#)). These settings are the default Control settings for LACP port channel interface policy groups in ACI. Unfortunately, these default Control settings are not always ideal. For example, LACP graceful convergence can lead to packet drops during port channel bringup and teardown when used to connect ACI switches to servers or non-Cisco switches that are not closely compliant with the LACP specification. As a general best practice, Cisco recommends keeping LACP graceful convergence enabled on port channels connecting to Nexus switches but disabling this setting when connecting to servers and non-Nexus switches.

## Configuring Virtual Port Channel (vPC) Domains

When configuring switch ports to servers and appliances, it is best to dual-home devices to switches to prevent total loss of traffic if a northbound switch fails. Some servers can handle failover at the operating system level very well and may be configured using individual ports from a switch point of view, despite being dual-homed. Where a server intends to hash traffic across links dual-homed across a pair of switches, virtual port channeling needs to be configured.

vPC technology allows links that are physically connected to two different Cisco switches to appear to a downstream device as coming from a single device and part of a single port channel. The downstream device can be a switch, a server, or any other networking device that supports Link Aggregation Control Protocol (LACP) or static port channels.

Standalone Nexus NX-OS software does support vPCs, but there are fewer caveats to deal with in ACI because ACI does not leverage peer links. In ACI, the keepalives and cross-switch communication needed for forming vPC domains all traverse the fabric.

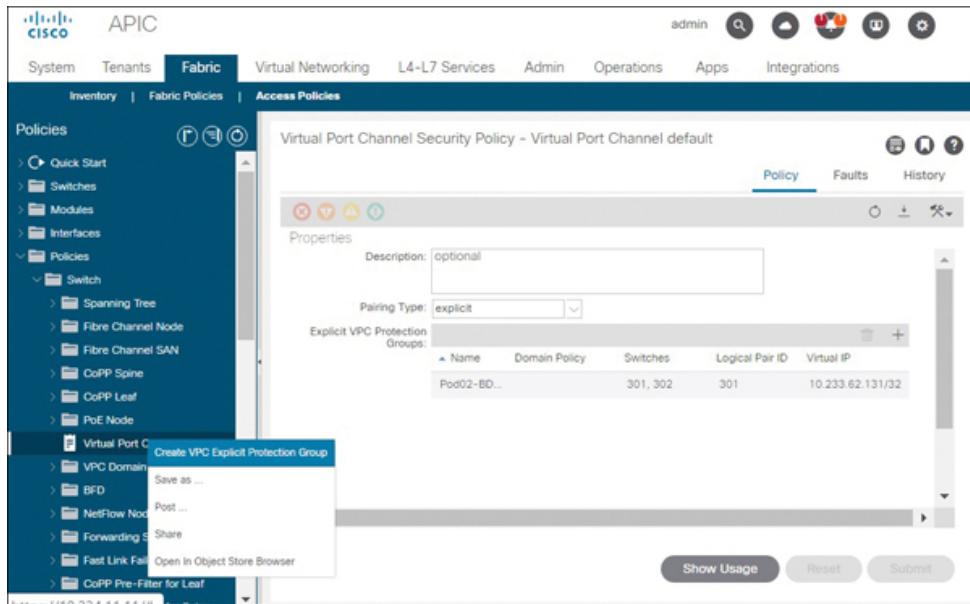
### Note

One limitation around vPC domain configuration in ACI that you should be aware of is that two vPC peer switches joined into a vPC domain must be of the same switch generation. This means you cannot form a vPC domain between a first-generation switch suffixed with TX and a

newer-generation switch suffixed with EX, FX, or FX2. ACI does allow migration of first-generation switches that are in a vPC domain to higher-generation switches, but it typically requires 10 to 20 seconds of downtime for vPC-attached servers.

The business unit running the multiplayer gaming project has purchased three additional switches and can now make use of vPCs in ACI. Before configuring virtual port channels, vPC domains need to be identified.

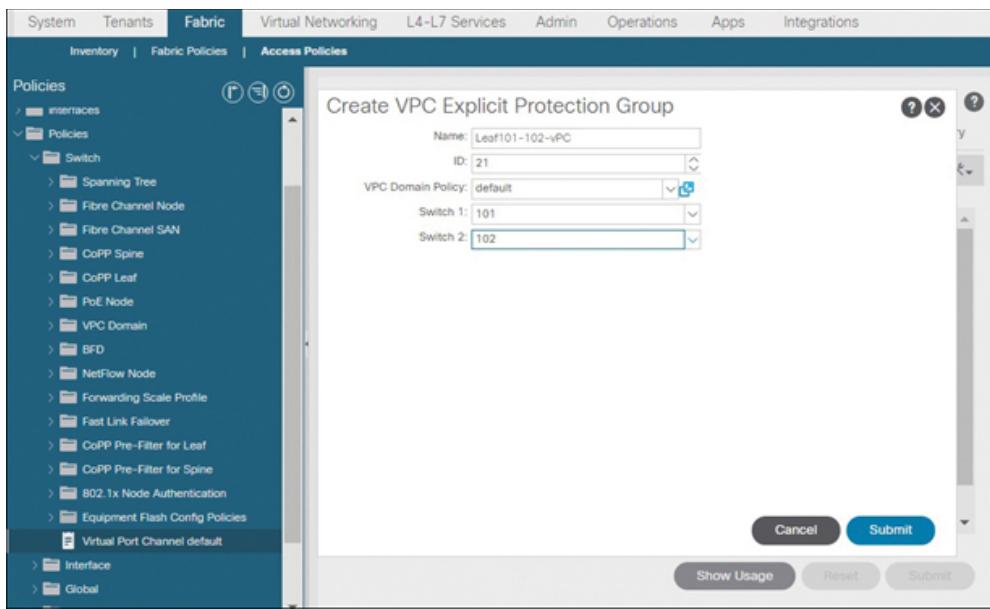
To configure a vPC domain, navigate to **Fabric > Access Policies > Policies > Switch**, right-click Virtual Port Channel Default, and select Create VPC Explicit Protection Group. [Figure 7-17](#) shows how to navigate to the Create VPC Explicit Protection Group wizard.



**Figure 7-17** Navigating to the Create VPC Explicit Protection Group Wizard

[Figure 7-18](#) shows how you can pair together two switches with node IDs 101 and 102 into vPC domain 21 by populating the Name, ID, Switch 1, and Switch 2 fields. Even though populating the Name field is mandatory, it has little impact on the configuration.

**Key Topic**



**Figure 7-18** Configuring a vPC Domain

The only vPC failover parameter that can be tweaked in ACI at the time of writing is the **vPC peer dead interval**, which is the amount of time a leaf switch with a vPC secondary role waits following a vPC peer switch failure before assuming the role of vPC master. The default peer dead interval in ACI is 200 seconds. This value can be tuned between 5 and 600 seconds through configuration of a vPC domain policy, which can then be applied to the vPC explicit protection group.

### Note

As a best practice, vPC domain IDs should be unique across each Layer 2 network. Problems can arise when more than one pair of vPC peer switches attached to a common Layer 2 network have the same vPC domain ID. This is because vPC domain IDs are a component in the generation of LACP system IDs.

The CLI-based equivalent of the vPC domain definition completed in this section is the command **vpc domain explicit 21 leaf 101 102**. Example 7-6 shows CLI verification of Leaf 101 and Leaf 102 having joined vPC domain ID 21. Note that the vPC peer status indicates that the peer adjacency with Leaf 102 has been formed, but the vPC keepalive status displays as Disabled. This is expected output from an operational vPC peering in ACI.

### Example 7-6 Verifying a vPC Peering Between Two Switches

[Click here to view code image](#)

LEAF101# show vpc

### Legend:

(\*) - local vPC is down, forwarding via vPC peer-link

vPC domain id	: 21
Peer status	: peer adjacency formed ok
vPC keep-alive status	: Disabled
Configuration consistency status	: success
Per-vlan consistency status	: success
Type-2 consistency status	: success
vPC role	: primary, operational
secondary	
Number of vPCs configured	: 1
Peer Gateway	: Disabled
Dual-active excluded VLANs	: -
Graceful Consistency Check	: Enabled
Auto-recovery status	: Enabled (timeout = 240
seconds)	
Operational Layer3 Peer	: Disabled

## vPC Peer-link status

<b>id</b>	<b>Port</b>	<b>Status</b>	<b>Active</b>	<b>vlangs</b>
--	-----	-----	-----	-----
-----	-----	-----	-----	-----
1		up		-

## vPC status

id	Port	Status	Consistency	Reason	Active
vlans					
--	-----	-----	-----	-----	-----
-----	-----	-----	-----	-----	-----



From a forwarding perspective, the result of creating a vPC explicit protection group is that ACI assigns a common virtual IP address to the

loopback 1 interface on the two vPC peers. This new IP address functions as a tunnel endpoint within the fabric, enabling all other switches in the fabric to forward traffic to either of the two switches via equal-cost multipathing. For this to work, the two vPC switches advertise reachability of vPC-attached endpoints using the loopback 1 interface, and traffic toward all endpoints that are not vPC attached continues to be forwarded to the tunnel IP addresses of the loopback 0 interfaces.

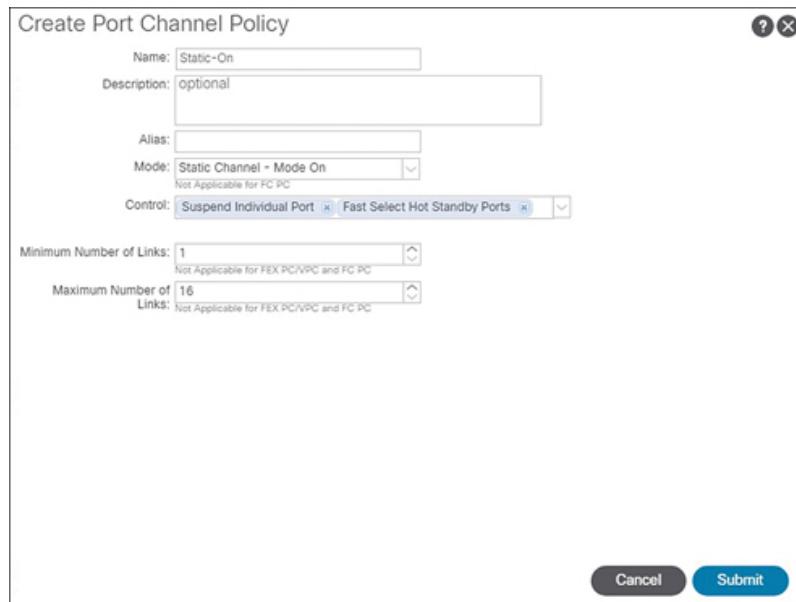
### Note

A vPC domain is a Layer 2 construct. ACI spine switches do not function as connection points for servers and non-ACI switches at Layer 2. Therefore, vPC is not a supported function for spine switches.

## Configuring Virtual Port Channels

Let's say you want to configure a resilient connection to a new multiplayer gaming server that does not support LACP but does support static port channeling. The first thing you need to do is to create a new interface policy that enables static port channeling. [Figure 7-19](#) shows such a policy.

### Key Topic



**Figure 7-19** Configuring an Interface Policy for Static Port Channeling

Next, you can move onto the configuration of a vPC interface policy group by navigating to **Fabric > Access Policies > Policies > Interface > Leaf Interfaces > Policy Groups**, right-clicking VPC Interface, and selecting Create VPC Interface Policy Group. [Figure 7-20](#) shows the configuration of a vPC interface policy group.



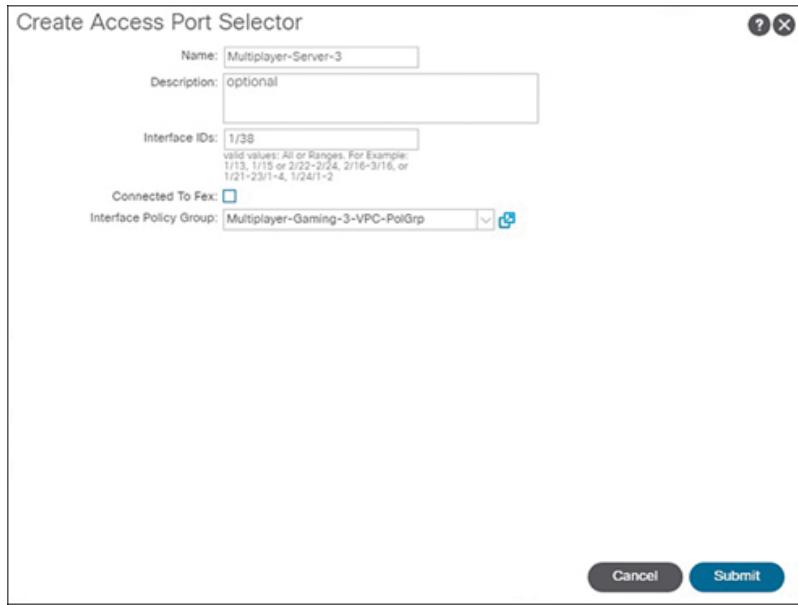
Create VPC Interface Policy Group

Name:	Multiplayer-Gaming-3-VPC-PolGr	?	X
Description:	optional		
Link Level Policy:	10Gbps	?	X
CDP Policy:	CDP-ENABLED	?	X
MCP Policy:	select a value		
CoPP Policy:	select a value		
LLDP Policy:	LLDP-ENABLED	?	X
STP Interface Policy:	select a value		
L2 Interface Policy:	select a value		
Port Security Policy:	select a value		
Egress Data Plane Policing Policy:	select a value		
Ingress Data Plane Policing Policy:	select a value		
Priority Flow Control Policy:	select a value		
Fibre Channel Interface Policy:	select a value		
Slow Drain Policy:	select a value		
MACsec Policy:	select a value		
Attached Entity Profile:	Bare-Metal-Servers-AAEF	?	X
Port Channel Policy:	Static-On	?	X
Monitoring Policy:	select a value		

**Figure 7-20** Configuring a vPC Interface Policy Group

Next, you need to associate the vPC interface policy group with interfaces on both vPC peers. The best way to associate policy to multiple switches simultaneously is to create an interface profile that points to all the desired switches.

[Figure 7-21](#) shows that the process of creating an access port selector for a vPC is the same as the process of configuring access port selectors for individual ports and port channels.



**Figure 7-21** Applying vPC Access Port Selectors to an Interface Profile for vPC Peers

The **show vpc** and **show port-channel summary** commands verify that the vPC has been created. As indicated in [Example 7-7](#), vPC IDs are also auto-generated by ACI.

### Example 7-7 Verifying the vPC Configuration from the Switch CLI

[Click here to view code image](#)

```
LEAF101# show vpc
Legend:
(*) - local vPC is down, forwarding via vPC peer-link

vPC domain id : 21
Peer status : peer adjacency formed ok
vPC keep-alive status : Disabled
Configuration consistency status : success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role : primary, operational
secondary
Number of vPCs configured : 2
Peer Gateway : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled (timeout = 240)
```

```

seconds)
Operational Layer3 Peer : Disabled

vPC Peer-link status
-----
-----
id   Port   Status Active vlans
--   ---   -----
1      up      -
-----

vPC status
-----
-----
id   Port   Status   Consistency   Reason   Active
vlans
--   ---   -----   -----   -----   -----
685  Po3    up       success      success   -
-----
```

**LEAF101# show port-channel summary**

```

Flags:  D - Down          P - Up in port-channel (members)
        I - Individual     H - Hot-standby (LACP only)
        S - Suspended      r - Module-removed
        S - Switched       R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
        F - Configuration failed
-----
```

Group	Port-Channel	Type	Protocol	Member Ports
1	Po1(SU)	Eth	LACP	Eth1/6(P) Eth1/8(P)
2	Po2(SU)	Eth	LACP	Eth1/31(P) Eth1/32(P)
3	Po3(SU)	Eth	NONE	Eth1/38(P)

**Example 7-8** shows the APIC CLI configurations equivalent to the GUI-based vPC configuration performed in this section.

#### **Example 7-8 Configuring a vPC Using the APIC CLI**

[Click here to view code image](#)

```
template port-channel Multiplayer-Gaming-3-VPC-PolGrp
    cdp enable
    vlan-domain member phys type phys
    speed 10G
    no negotiate auto
    exit
leaf-interface-profile LEAF101-102-vPC-IntProfile
    leaf-interface-group Multiplayer-Server-3
        interface ethernet 1/38
        channel-group Multiplayer-Gaming-3-VPC-PolGrp vpc
        exit
    exit
```

The static port channel policy setting does not show up in the configuration. As shown in [Example 7-9](#), by adding the keyword **all** to the command, you can confirm that the setting has been applied.

**Example 7-9** *Using all to Include Defaults Not Otherwise Shown in the APIC CLI*

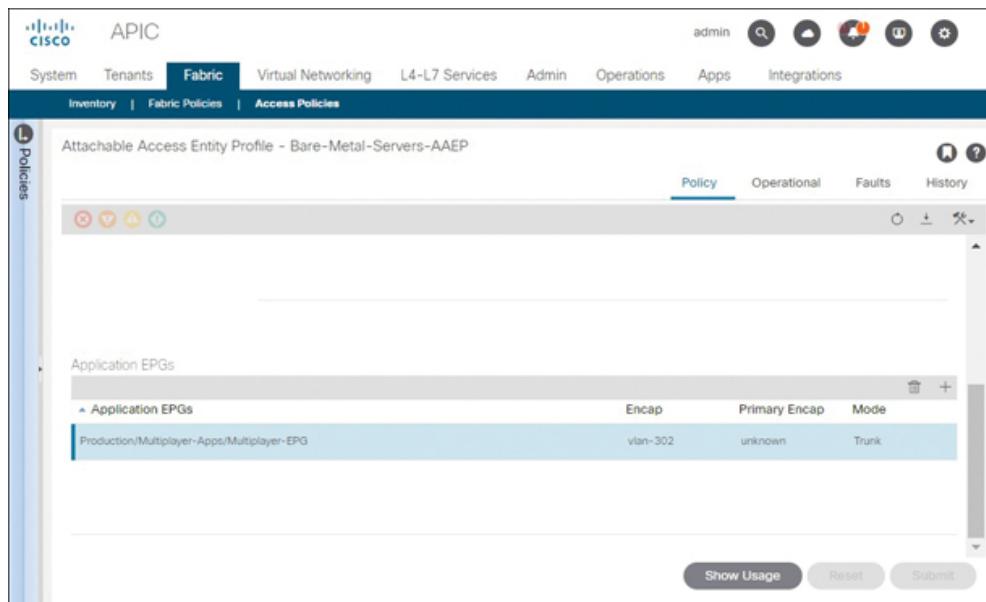
[Click here to view code image](#)

```
APIC1(config)# show running-config all template port-channel
Multiplayer-Gaming-3-VPC-PolGrp
(...output truncated for brevity...)
template port-channel Multiplayer-Gaming-3-VPC-PolGrp
    no description
    lldp receive
    lldp transmit
    cdp enable
    vlan-domain member phys type phys
    channel-mode on
        lacp min-links 1
        lacp max-links 16
        no lacp symmetric-hash
        exit
    mcp enable
    spanning-tree bpdu-filter disable
    spanning-tree bpdu-guard disable
    speed 10G
```

```
no negotiate auto  
exit
```

## Configuring Ports Using AAEP EPGs

Even seasoned ACI engineers are often under the impression that EPG assignments can only be made under the Tenants menu. This is not true. [Figure 7-22](#) shows the mapping of an EPG to VLAN 302. The mappings in this view require that users prefix the VLAN ID with **vlan-**.



**Figure 7-22** Mapping One or More EPGs to All Ports Leveraging a Specific AAEP

[Figure 7-23](#) shows that after making this change, the newly configured ports, which all referenced the AAEP, transition out of the Usage status Discovery to EPG.

Stats	Interfa	Speed	Layer	Mode	Switching State	Usage	Oper Vlans	Configured Vlans	Oper Duplex	Oper State	Oper State Reason
	eth1/51	inherit	routed	trunk	disabled	Fabric External			full	down	sfp-missing
	eth1/50	inherit	routed	trunk	enabled	Fabric			full	up	none
	eth1/49	inherit	routed	trunk	enabled	Fabric			full	up	none
	eth1/48	inherit	switched	trunk	disabled	Discovery			full	down	sfp-missing
	eth1/47	inherit	switched	trunk	disabled	Discovery			full	down	sfp-missing
	eth1/46	inherit	switched	trunk	enabled	EPG	30-31	30-31	full	up	none
	eth1/45	inherit	switched	trunk	enabled	EPG	30-31	30-31	full	up	none
	eth1/44	inherit	switched	trunk	disabled	Discovery			full	down	sfp-missing
	eth1/43	inherit	switched	trunk	disabled	Discovery			full	down	sfp-missing
	eth1/42	inherit	switched	trunk	disabled	Discovery			full	down	sfp-missing
	eth1/41	inherit	switched	trunk	disabled	Discovery			full	down	sfp-missing
	eth1/40	inherit	switched	trunk	disabled	Discovery			full	down	sfp-missing

**Figure 7-23 Ports with AAEP Assignment Transitioned to the EPG State**

As shown in [Example 7-10](#), the ports are no longer out of service. This indicates that tenant policies have successfully been layered on the access policies by using the AAEP.

### **Example 7-10 Operational Ports with an Associated EPG Transition to Connected Status**

[Click here to view code image](#)

```
LEAF101# show interface ethernet 1/45-46, ethernet 1/31-32, ethernet 1/38
status
-----
-----
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Eth1/31	--	connected	trunk	full	10G	10Gbase-SR
Eth1/32	--	connected	trunk	full	10G	10Gbase-SR
Eth1/38	--	connected	trunk	full	10G	10Gbase-SR
Eth1/45	--	connected	trunk	full	10G	10Gbase-SR
Eth1/46	--	connected	trunk	full	10G	10Gbase-SR

## Key Topic

So, what are AAEP EPGs, and why use this method of EPG-to-VLAN assignment? Static path mappings in the Tenants view associate an EPG with a single port, port channel, or vPC. If a set of 10 EPGs need to be associated with 10 servers, a sum total of 100 static path assignments are needed. On the other hand, if exactly the same EPG-to-VLAN mappings are required for the 10 servers, the 10 assignments can be made once to an AAEP, allowing all switch ports referencing the AAEP to inherit the EPG-to-VLAN mappings. This reduces administrative overhead in some environments and eliminates configuration drift in terms of EPG assignments across the servers.

### Note

Some engineers strictly stick with EPG-to-VLAN mappings that are applied under the Tenants menu, and others focus solely on AAEP EPGs. These options are not mutually exclusive. The method or methods selected should be determined based on the business and technical objectives. Methods like static path assignment result in large numbers of EPG-to-VLAN mappings because each port needs to be individually assigned all the desired mappings, but the number of AAEPs in this approach can be kept to a minimum.

In environments that solely center around AAEP EPGs, there are as many AAEPs as there are combinations of EPG-to-VLAN mappings. Therefore, the number of AAEPs in such environments is higher, but tenant-level mappings are not necessary. In environments in which automation scripts handle the task of assigning EPGs to hundreds of ports simultaneously, there may be little reason to even consider AAEP EPGs. However, not all environments center around scripting.

## Implications of Initial Access Policy Design on Capabilities

What are some of the implications of the configurations covered so far in this chapter? The EPG trunked onto the object Bare-Metal-Servers-AAEP resides in the tenant Production. This particular customer wants to manage its own servers, so would it make more sense to isolate the customer's servers and applications in a dedicated tenant? The answer most likely is yes.

If a new tenant were built for the multiplayer gaming applications, the business unit could be provided not just visibility but configuration access to its tenant. Tasks like creating new EPGs and EPG-to-port mappings could then be offloaded to the business unit.

In addition, what happens if this particular customer wants to open up communication between the tenant and a specific subnet within the campus? In this case, a new external EPG may be needed to classify traffic originating from the campus subnet. Creating a new external EPG for L3Outs in already available VRF instances in the Production tenant could force a reevaluation of policies to ensure continuity of connectivity for other applications to the destination subnet. Sometimes, use of a new tenant can simplify the external EPG design and the enforcement of security policies.

Finally, what are the implications of the AAEP and domain design? If central IT manages ACI, there's really nothing to worry about. However, if all bare-metal servers in a fabric indeed leverage a common AAEP object as well as a common domain, how would central IT be able to prevent the gaming business unit from mistakenly mapping an EPG to a corporate IT server? How could central IT ensure that an unintended VLAN ID is not used for the mapping? The answer is that it cannot. This highlights the importance of good AAEP and domain design.

In summary, where there is a requirement for the configuration of a server environment within ACI to be offloaded to a customer or an alternate internal organization or even when there are requirements for complete segmentation of traffic in one environment (for example, production) and a new server environment, it often makes sense to use separate tenants, separate physical domains, and separate non-overlapping VLAN pools. Through enforcement of proper role-based access control (RBAC) and scope-limiting customer configuration changes to a specific tenant and relevant domains, central IT is then able to ensure that any configuration changes within the tenant do not impact existing operations in other server environments (tenants).

## Configuring Access Policies Using Quick Start Wizards

All the configurations performed in the previous section can also be done using quick start wizards. There are two such wizards under the Access Policies view: the Configure Interface, PC, and vPC Wizard and the Configure Interface Wizard.

# The Configure Interface, PC, and VPC Wizard

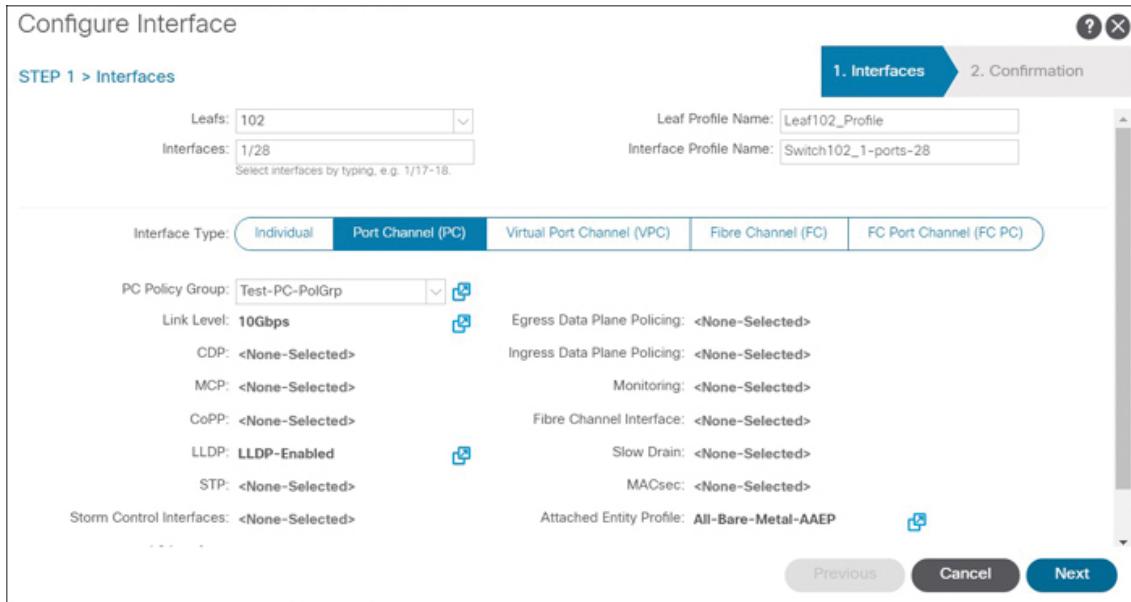
Under **Fabric > Access Policies > Quick Start**, click on Configure Interface, PC, and VPC. The page shown in [Figure 7-24](#) appears. Everything from a switch profile-to-node ID association to interface policies and mapping configurations can be done in this simple view.

The screenshot shows the 'Configure Interface, PC, and VPC' wizard. At the top, it says 'Configured Switch Interfaces' and 'VPC Switch Pairs'. Under 'Configured Switch Interfaces', there's a table with columns: Switches, Interfaces, IF Type, and Attached Device Type. It lists several entries, including '1/34 VPC Bare Metal (VLAN: 244.2...)' and '1/8,1/10,1... VPC L3 (VLANs: 2100-2114...)'. Below this is a section for 'Select Switches To Configure Interfaces' with tabs for 'Quick' and 'Advanced'. The 'Switches' dropdown is set to '102'. The 'Interface Type' dropdown has 'Individual' selected. The 'Switch Profile Name' field is 'Switch102\_Profile'. An 'Interfaces' dropdown shows '1/28' and an 'Interface Selector Name' field contains 'Switch102\_1\_ports\_28'. A 'Create One' button is available for interface policy groups. The right side of the interface is filled with various policy configuration fields like CDP, LLDP, Monitoring, L2 Interface, Port Policy, Egress Data Plane, IPv4 NetFlow Monitor, IPv6 NetFlow Monitor, and Layer2-Switched (CE type) NetFlow Monitor. At the bottom, there are buttons for 'Cancel', 'Save', and 'Submit'.

**Figure 7-24** The Configure Interface, PC, and VPC Wizard

## The Configure Interface Wizard

Under **Fabric > Access Policies > Quick Start**, notice the Configure Interface wizard. Click it to see the page shown in [Figure 7-25](#). This page provides a convenient view for double-checking previously configured interface policy group settings before making port assignments.



**Figure 7-25** View of the Configure Interface Wizard

## Additional Access Policy Configurations

The access policies covered so far in this chapter apply to all businesses and ACI deployments. The sections that follow address the implementation of less common access policies.

## Configuring Fabric Extenders

Fabric extenders (FEX) are a low-cost solution for low-bandwidth port attachment to a parent switch. Fabric extenders are less than ideal for high-bandwidth and low-latency use cases and do not have a lot of appeal in ACI due to feature deficiencies, such as analytics capabilities.

### Note

Ideally, new ACI deployments should not leverage fabric extenders. This book includes coverage of FEX because it is a topic that can appear on the Implementing Cisco Application Centric Infrastructure DCACI 300-620 exam and because not all companies are fortunate enough to be able to remove fabric extenders from their data centers when first migrating to ACI.

Fabric extenders attach to ACI fabrics in much the same way they attach to NX-OS mode switches. However, ACI does not support dual-homing of fabric extenders to leaf switch pairs in an active/active FEX design.

Instead, to make FEX-attached servers resilient to the loss of a single server uplink in ACI, you need to dual-home the servers to a pair of fabric extenders. Ideally, these fabric extenders connect to different upstream leaf switches that form a vPC domain. In such a situation, you can configure vPCs from the servers up to the fabric extenders to also protect server traffic against the failure of a single leaf switch.

There are two steps involved in implementing a fabric extender:

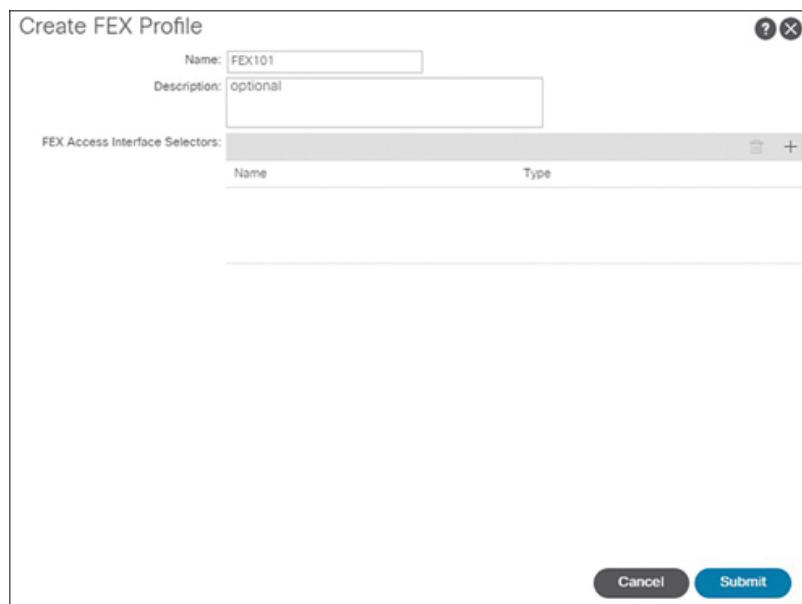
**Step 1.** Configure a FEX profile.

**Step 2.** Associate the FEX profile with the parent switch by configuring access policies down to the fabric extender.



After these two steps have been completed, you can configure FEX downlinks to servers by configuring access port selectors on the newly deployed FEX profile.

Let's say you want to deploy a fabric extender to enable low-bandwidth CIMC connections down to servers. To do so, navigate to **Fabric > Access Policies > Interfaces > Leaf Interfaces**, right-click Profiles, and select Create FEX Profile. The page shown in [Figure 7-26](#) appears. The FEX Access Interface Selectors section is where the CIMC port mappings need to be implemented. Enter an appropriate name for the FEX interface profile and click Submit.

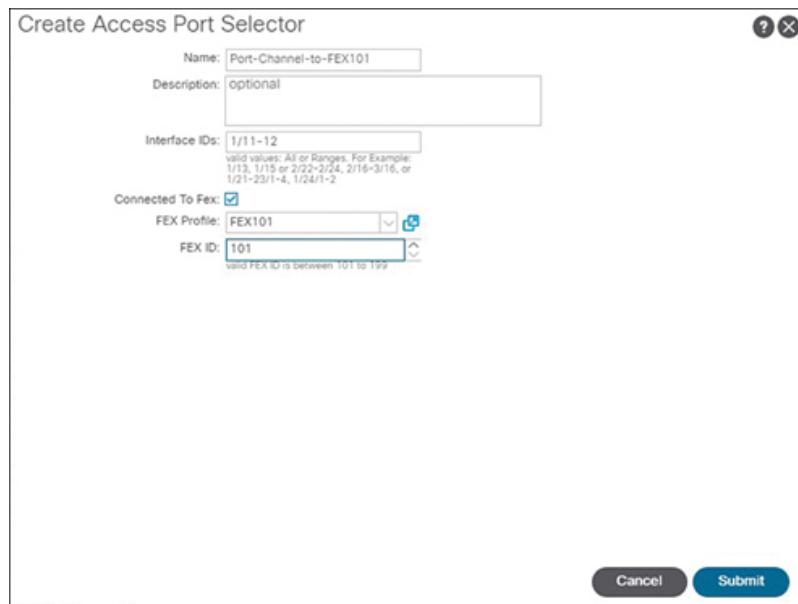


The screenshot shows a dialog box titled "Create FEX Profile". It has fields for "Name" (set to "FEX101") and "Description" (set to "optional"). Below these is a section titled "FEX Access Interface Selectors:" with a table header "Name" and "Type". At the bottom are "Cancel" and "Submit" buttons.

Name	Type

**Figure 7-26** Configuring a FEX Profile

Next, navigate to the interface profile of the parent leaf and configure an interface selector. In [Figure 7-27](#), ports 1/11 and 1/12 on Leaf 101 connect to uplink ports on the new fabric extender. To expose the list of available FEX profiles, enable the Connected to Fex checkbox and select the profile of the FEX connecting to the leaf ports.



**Figure 7-27** Associating a FEX Profile with a Parent Switch

After you click Submit, ACI bundles the selected ports into a static port channel, as indicated by the output NONE in the Protocol column in [Example 7-11](#). The FEX eventually transitions through several states before moving to the Online state.

### Example 7-11 Verifying FEX Association with a Parent Leaf Switch

[Click here to view code image](#)

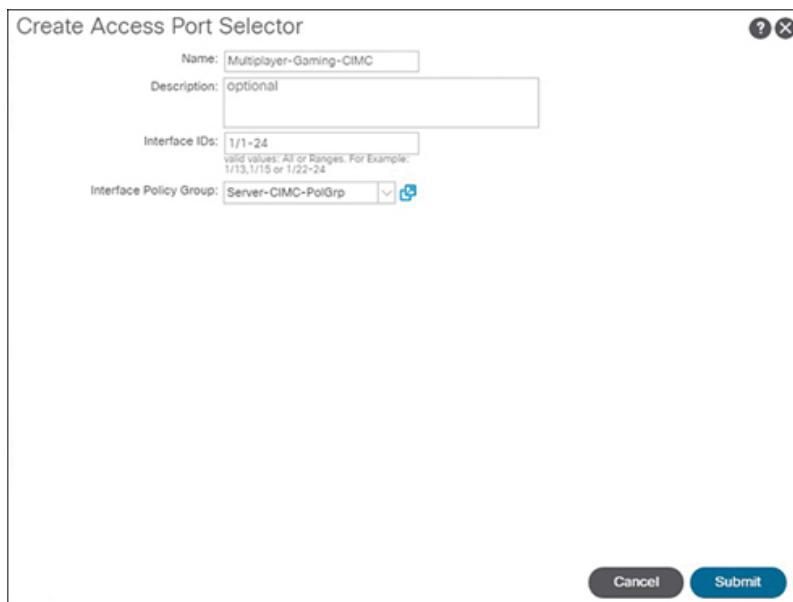
```
LEAF101# show port-channel summary
Flags:  D - Down          P - Up in port-channel (members)
       I - Individual    H - Hot-standby (LACP only)
       S - Suspended      r - Module-removed
       S - Switched       R - Routed
       U - Up (port-channel)
       M - Not in use. Min-links not met
       F - Configuration failed
-----
--
```

Group	Port-Channel	Type	Protocol	Member Ports	
<hr/>					
--					
1	Po1(SU)	Eth	LACP	Eth1/6(P)	Eth1/8(P)
2	Po2(SU)	Eth	LACP	Eth1/31(P)	Eth1/32(P)
3	Po3(SU)	Eth	NONE	Eth1/38(P)	
4	Po4(SU)	Eth	NONE	Eth1/11(P)	Eth1/12(P)

LEAF101# show fex							
FEX Number	FEX Description	FEX State	FEX Model	Serial			
<hr/>							
<hr/>							
101	FEX0101	Online	N2K-C2248TP-1GE	XXXXX			

When the FEX has been operationalized, access policies are still needed for FEX port connectivity down to CIMC ports. You can navigate to the FEX profile and configure an interface selector for these ports. [Figure 7-28](#) shows connectivity for 24 FEX ports being prestaged using a newly created interface policy group for non-aggregated ports.



**Figure 7-28** Configuring FEX Downlinks to Servers via FEX Interface Profiles

[Example 7-12](#) shows how fabric extenders might be implemented via the APIC CLI.

## **Example 7-12 Configuring a FEX and Downstream Connectivity via the APIC CLI**

[Click here to view code image](#)

```
APIC1# show running-config leaf-interface-profile LEAF101-IntProfile
(...output truncated for brevity...)
leaf-interface-profile LEAF101-IntProfile
  leaf-interface-group Port-Channel-to-FEX101
    interface ethernet 1/11-12
    fex associate 101 template FEX101
    exit
  exit

APIC1# show running-config fex-profile FEX101
fex-profile FEX101
  fex-interface-group Multiplayer-Gaming-CIMC
    interface ethernet 1/1-24
    policy-group Server-CIMC-PolGrp
    exit
  exit
```

### **Note**

Not all ACI leaf switches can function as FEX parents.

## **Configuring Dynamic Breakout Ports**

Cisco sells ACI leaf switches like the Nexus 93180YC-FX that are optimized for 10 Gbps/25 Gbps compute attachment use cases. It also offers switch models like the Nexus 9336C-FX2, whose 36 ports each support speeds of up to 100 Gbps.

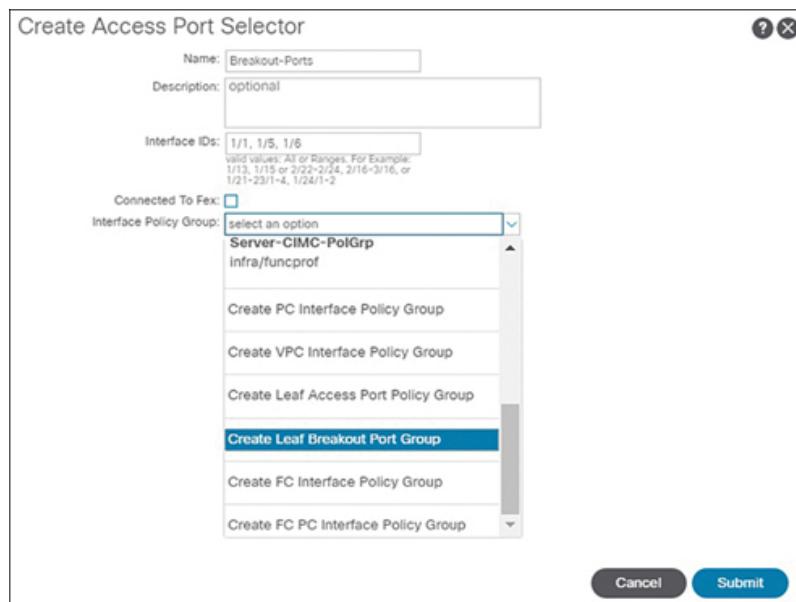
Port speeds on platforms like the Nexus 9336C-FX2 can be lowered by seating a CVR-QSFP-10G adapter into a port, along with a supported 10 Gbps or 1 Gbps transceiver. Purchasing a platform like this and using CVR adapters to lower port speeds, however, could turn out to be an expensive approach when calculating the per-port cost because this approach makes suboptimal use of the forwarding capacity of the switch. This approach may still be deemed economical, however, if only a fraction of ports are “burned” this way.

**Key Topic**

Another approach is to dynamically split ports into multiple lower-speed connections. With **dynamic breakout ports**, a 40 Gbps switch port can be split into four independent and logical 10 Gbps ports. Likewise, a 100 Gbps port can be split into four independent and logical 25 Gbps ports. This does require special breakout cabling, but it allows customers to use a greater amount of the forwarding capacity of high-bandwidth ports.

Let's say you have just initialized two new high-density Nexus switches with node IDs 103 and 104. These two switches both support dynamic breakout ports. Imagine that you have been asked to deploy 12 new servers, and each server needs to be dual-homed to these new switches using 25 Gbps network cards. Since the ports on these particular switches are optimized for 100 Gbps connectivity, implementation of dynamic breakout ports can help. Splitting three 100 Gbps ports on each switch, in this case, yields the desired 12 25 Gbps connections from each leaf to the servers.

To deploy dynamic breakout ports, create a new interface selector on the interface profiles bound to each switch and select Create Leaf Breakout Port Group from the Interface Policy Group drop-down box, as shown in [Figure 7-29](#).



**Figure 7-29** Navigating to the Create Leaf Breakout Port Group Page

On the Create Leaf Breakout Port Group page, select a name for the new interface selector and select an option from the Breakout Map drop-down

box. [Figure 7-30](#) shows the option 25g-4x being selected, which implies that a 100 Gbps port will be broken out into four 25 Gbps ports.



Create Leaf Breakout Port Group

Name:

Description:

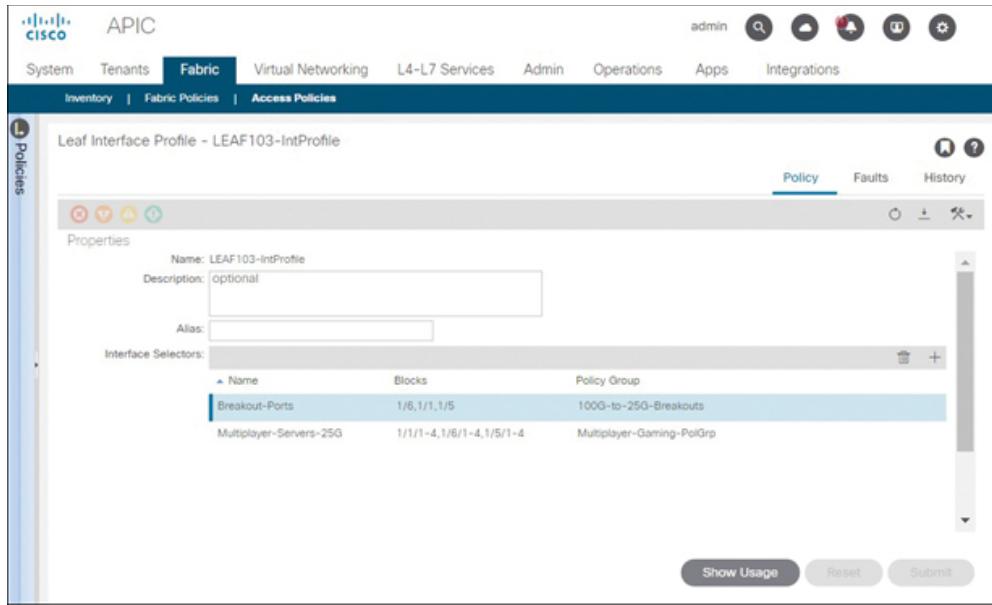
Breakout Map:

**Figure 7-30** Configuring Dynamic Port Breakouts

When implementing the equivalent breakouts for any additional nodes, you may find that you can reuse the interface policy group that references the breakout map.

Once breakouts have been implemented on the desired ports on the switches, you can configure the desired access policies for the resulting subports. These subports resulting from dynamic breakouts need to be referenced using the numbering *module/port/subport*. [Figure 7-31](#) illustrates access policies being applied to subports of interface 1/1—namely, logical ports 1/1/1, 1/1/2, 1/1/3, and 1/1/4.





**Figure 7-31** Implementing Access Policies for Subports Within a Breakout Port

Example 7-13 shows the APIC CLI commands that are equivalent to the GUI-based dynamic breakout port configurations implemented in this section.

**Example 7-13** Implementing Dynamic Breakout Ports via the APIC CLI  
[Click here to view code image](#)

```
leaf-interface-profile LEAF103-IntProfile
  leaf-interface-group Breakout-Ports
    interface ethernet 1/1
    interface ethernet 1/5
    interface ethernet 1/6
    breakout 25g-4x
  exit
  leaf-interface-group Multiplayer-Servers-25G
    interface ethernet 1/1/1-4
    interface ethernet 1/5/1-4
    interface ethernet 1/6/1-4
    policy-group Multiplayer-Gaming-PolGrp
  exit
exit
```

## Configuring Global QoS Class Settings

Quality of service (QoS) allows administrators to classify network traffic and prioritize and police the traffic flow to help avoid congestion in the network.

To gain an understanding of QoS in ACI, the behavior of the platform can be analyzed in four key areas:

- **Traffic classification:** *Traffic classification* refers to the method used for grouping traffic into different categories or classes. ACI classifies traffic into *priority levels*. Current ACI code has six user-configurable priority levels and several reserved QoS groups. ACI allows administrators to classify traffic by trusting ingress packet headers, such as Differentiated Services Code Point (DSCP) or Class of Service (CoS). Administrators can also assign a priority level to traffic via contracts or by manually assigning an EPG to a priority level.
- **Policing:** The term *policing* refers to enforcement of controls on traffic based on classification. Even though there should be no oversubscription concerns in ACI fabrics, there is still a need for policing. Suppose backup traffic has been trunked on the same link to a server as data traffic. In such cases, administrators can police traffic to enforce bandwidth limits on the link for the backup EPG. ACI policing can be enforced on an interface or on an EPG. If traffic exceeds prespecified limits, packets can be either marked or dropped. Policing applies both in the inbound direction and in the outbound direction.
- **Marking:** Once a switch classifies traffic, it can also *mark* traffic by setting certain values in the Layer 3 header (DSCP) or in the Layer 2 header (Class of Service [CoS]) to notify other switches in the traffic path of the desired QoS treatment. Under default ACI settings, marking takes place on ingress leaf switches only.
- **Queuing and scheduling:** Once a platform assigns packets to a QoS group, outbound packets are queued for transmission. Multiple queues can be used based on packet priority. A scheduling algorithm determines which queue's packet should be transmitted next. Scheduling and queuing, therefore, collectively refer to the process of prioritization of network packets and scheduling their transmission outbound on the wire. ACI uses the Deficit Weighted Round Robin (DWRR) scheduling algorithm.

Which aspects of QoS relate to access policies? Global QoS class settings govern priority levels and other fabricwide aspects of QoS applicable to treatment of server and other endpoint traffic and therefore fall under access policies.

To review the global QoS class settings or make changes, navigate to **Fabric > Access Policies > Policies > Global > QoS Class**.

### Key Topic

Let's say that at some point in the future, your company intends to connect its Cisco Unified Computing System (UCS) domains to the ACI fabric in an effort to gradually migrate all workloads into ACI. Default gateways will move into the fabric at a later time, and legacy data center infrastructure is expected to remain in the network for a long time. UCS server converged network adapters (CNA) tag certain critical traffic with CoS values, and the production network currently honors markings from UCS servers. The IT organization wants to ensure that ACI preserves these CoS values and restores them as these packets leave the fabric so that the legacy network can act on these markings. After reviewing the settings in the Global - QoS Class page, you might learn that ACI preserves DSCP markings by default but does not preserve CoS markings. You can enable the Dot1p Preserve setting to address this requirement, as shown in [Figure 7-32](#).

The screenshot shows the Cisco Application Policy Infrastructure Controller (APIC) web interface. The top navigation bar includes links for System, Tenants, Fabric (which is selected), Virtual Networking, L4-L7 Services, Admin, Operations, Apps, and Integrations. The user is logged in as 'admin'. On the left, a sidebar titled 'Policies' lists various policy types: Quick Start, Switches, Modules, Interfaces, Policies, Switch, Interface, Global, Attachable Access, QoS Class, DHCP Relay, MCP Instance Pool, Error Disabled Rule, Monitoring, Troubleshooting, Pools, and Physical and External Devices. Under 'Policies', 'QoS Class' is expanded, and 'Global' is selected. The main content area is titled 'Global - QoS Class' and displays a table of six QoS levels. A 'Properties' section above the table includes a checkbox for 'Preserve COS: Dot1p Preserve', which is checked. The table columns are: Name, Admin State, Priority Flow Control, No-Drop-Cos, MTU, Minimum Buffers, Congestion Algorithm, Congestion Notification, Queue Control, Queue Limit (bytes), Scheduling Algorithm, and Bandwidth Allocation (in %). The rows show the following data:

Name	Admin State	Priority Flow Control	No-Drop-Cos	MTU	Minimum Buffers	Congestion Algorithm	Congestion Notification	Queue Control	Queue Limit (bytes)	Scheduling Algorithm	Bandwidth Allocation (in %)
Level1	Enabled	false	802.1p	9216	0	Tail Drop	Disabled	Dynamic	1522	Weighted round robin	20
Level2	Enabled	false	802.1p	9216	0	Tail Drop	Disabled	Dynamic	1522	Weighted round robin	20
Level3 (Default)	Enabled	false	802.1p	9216	0	Tail Drop	Disabled	Dynamic	1522	Weighted round robin	20
Level4	Enabled	false	802.1p	9216	0	Tail Drop	Disabled	Dynamic	1522	Weighted round robin	0
Level5	Enabled	false	802.1p	9216	0	Tail Drop	Disabled	Dynamic	1522	Weighted round robin	0
Level6	Enabled	false	802.1p	9216	0	Tail Drop	Disabled	Dynamic	1522	Weighted round robin	0

At the bottom right of the table are three buttons: 'Show Usage', 'Reset', and 'Submit'.

**Figure 7-32** Enabling the Dot1p Preserve Checkbox on the Global - QoS Class Page

### Key Topic

Notice in the figure the six user-configurable QoS priority levels in ACI and the bandwidth allocation for each of them. Any traffic that cannot be otherwise classified into a priority level gets assigned to the default class (Level 3).

### Note

Reserved QoS groups in ACI consist of APIC controller traffic, control plane protocol traffic, Switched Port Analyzer (SPAN) traffic, and traceroute traffic. ACI places APIC and control plane protocol traffic in a strict priority queue; SPAN and traceroute traffic are considered best-effort traffic.

## Configuring DHCP Relay

In ACI, if a bridge domain has been configured to allow flooding of traffic and a DHCP server resides within an EPG associated with the bridge domain, any endpoints within the same EPG can communicate with the DHCP server without needing DHCP relay functionality.

When flooding is not enabled on the bridge domain or when the DHCP server resides in a different subnet or EPG than endpoints requesting dynamic IP assignment, DHCP relay functionality is required.

To define a list of DHCP servers to which ACI should relay DHCP traffic, a DHCP relay policy needs to be configured. There are three locations where a DHCP relay policy can be configured in ACI:

- **In the Access Policies view:** When bridge domains are placed in user tenants and one or more DHCP servers are expected to be used across these tenants, DHCP relay policies should be configured in the Access Policies view.
- **In the common tenant:** When bridge domains are placed in the common tenant and EPGs reside in user tenants, DHCP relay policies are best placed in the common tenant.
- **In the infra tenant:** When DHCP functionality is needed for extending ACI fabric services to external entities such as hypervisors and VMkernel interfaces need to be assigned IP addresses from the infra tenant, DHCP relay policies need to be configured in the infra tenant. This option is beyond the scope of the DCACI 300-620 exam and, therefore, this book.

Once DHCP relay policies have been configured, bridge domains can reference these policies.

Let's say that you need to configure a DHCP relay policy referencing all DHCP servers within the enterprise network. In your environment, your team has decided that bridge domains will all be configured in user tenants. For this reason, DHCP relay policies should be configured under **Fabric > Access Policies > Policies > Global > DHCP Relay**. [Figure 7-33](#) shows how you create a new DHCP policy by entering a name and adding providers (DHCP servers) to the policy.



Create DHCP Relay Policy

Name:

Description:

Providers:

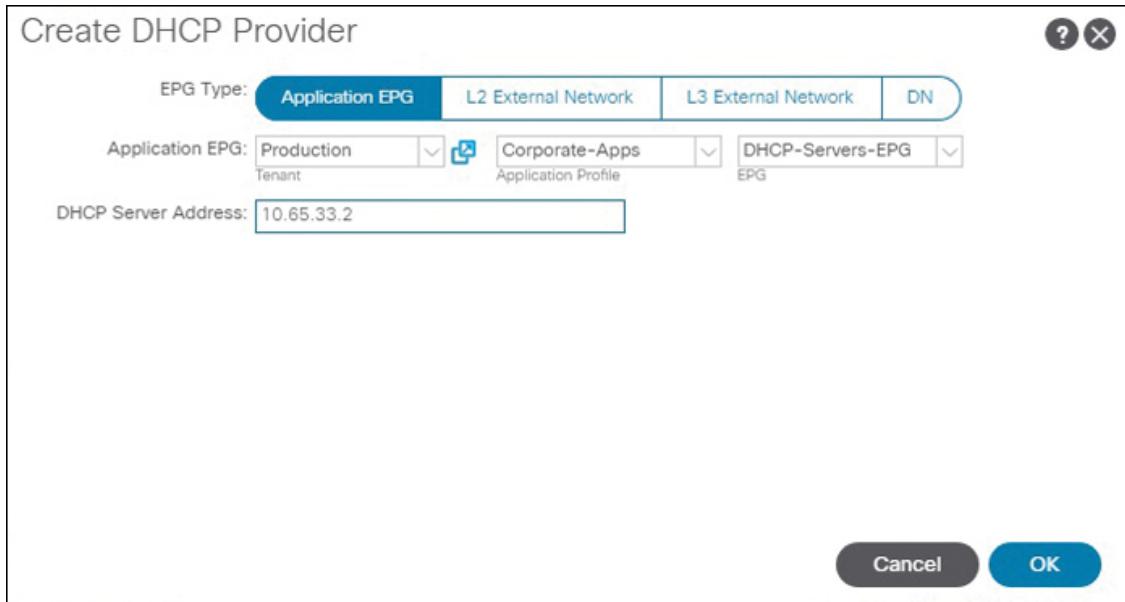
Associated EPG	DHCP Server Address

The screenshot shows a dialog box titled "Create DHCP Relay Policy". It has fields for "Name" (set to "Corporate-DHCP-Servers") and "Description" (set to "optional"). Below these is a "Providers" section with a table. The table has two columns: "Associated EPG" and "DHCP Server Address", both of which are currently empty. There are "Cancel" and "Submit" buttons at the bottom. A "Key Topic" icon is located to the left of the dialog box.

**Figure 7-33** Configuring a New DHCP Relay Policy in the Access Policies View

Define each DHCP server by adding its address and the location where it resides. Where a DHCP server resides within the fabric, select Application EPG and define the tenant, application profile, and EPG in which the server resides and then click OK. Then add any redundant DHCP servers to the policy and click Submit.





**Figure 7-34** Configuring a Provider Within a DHCP Relay Policy

Chapter 8, “[Implementing Tenant Policies](#),” covers assignment of DHCP relay policies to EPGs and DHCP relay caveats in ACI.

## Configuring MCP

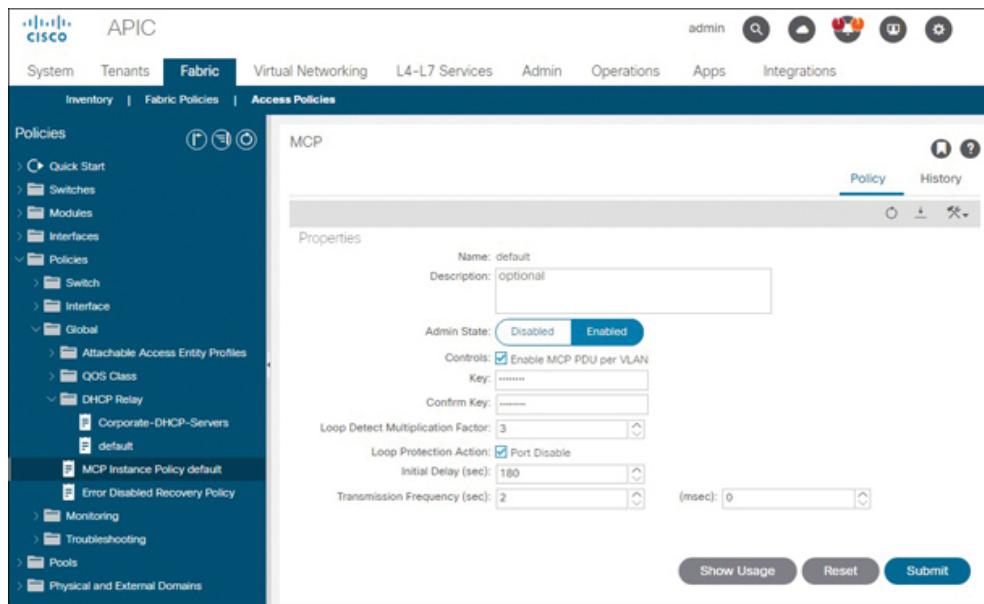
A Layer 2 loop does not impact the stability of an ACI fabric because ACI can broadcast traffic at line rate with little need to process the individual packets. Layer 2 loops, however, can impact the ability of endpoints to process important traffic. For this reason, mechanisms are needed to detect loops resulting from miscabling and misconfiguration. One of the protocols ACI uses to detect such externally generated Layer 2 loops is MisCabling Protocol (MCP).

### Key Topic

MCP is disabled in ACI by default. To enable MCP, you must first enable MCP globally and then ensure that it is also enabled at the interface policy level. As part of the global enablement of MCP, you define a key that ACI includes in MCP packets sent out on access ports. If ACI later receives an MCP packet with the same key on any other port, it knows that there is a Layer 2 loop in the topology. In response, ACI can either attempt to mitigate the loop by disabling the port on which the MCP protocol data unit was received or it can generate a system message to notify administrators of the issue.

**Key Topic**

To enable MCP globally, navigate to **Fabric > Access Policies > Policies > Global > MCP Instance Policy Default**. As shown in [Figure 7-35](#), you can then enter a value in the Key field, toggle Admin State to Enabled, check the Enable MCP PDU per VLAN checkbox, select the desired Loop Prevention Action setting, and click Submit.



**Figure 7-35** Enabling MCP Globally Within a Fabric

[Table 7-4](#) describes these settings.

**Table 7-4** Settings Available in Global MCP Policy

**Key Topic**

**Set Description  
ting**

## Set Description ting

Ad min Stat e	This setting determines whether MCP is globally enabled or disabled. The default setting for this field is Disabled.
Ena ble MCP PDU per VLA N	By default, ACI only sends MCP packets on the native VLAN on a port. This tends to be useless in detecting Layer 2 loops when an EPG has been trunked over a port. To ensure that loops behind tagged ports can also be detected, the Enable MCP PDU per VLAN option needs to be checked. If this option is checked, ACI sends MCP packets on up to 256 VLANs per interface. If more than 256 VLANs have been mapped to EPGs on a port, the first 256 VLAN IDs are chosen.
Key	This is a string that ACI includes in MCP packets to uniquely identify the fabric with the intent to be able to later validate whether it has been the originator of a given MCP packet.
Loo p Det ect Mul tipl icati on Fact or	This is the number of self-originated continuous MCP packets ACI needs to receive before it declares a loop. The default value for this setting is 3. With default settings, it takes ACI approximately 7 seconds to detect a loop.

## Set Description ting

Loop Protection Action	This is the response ACI takes after receiving a number of self-originated MCP packets on a port. If the Port Disable option is checked, ACI disables the port on which the MCP packets have been received and logs the incident. If the Port Disabled checkbox is disabled, ACI just logs the incident, which can be forwarded to a syslog server for administrators to take action.
Initial Delay	This is the delay time, in seconds, before MCP begins taking action. By default, the option is set to 180 seconds, but it can be tuned down.
Transmission Frequency	This is the frequency for transmission of MCP packets, in seconds or milliseconds.

### Key Topic

To enable MCP on a port-by-port basis, create an explicit MCP interface policy by navigating to **Fabric > Access Policies > Policies > Interface**, right-clicking MCP Interface, and selecting Create MisCabling Protocol Interface Policy. Assign a name to the policy and toggle Admin State to Enabled, as shown in [Figure 7-36](#).

Create Mis-cabling Protocol Interface Policy

Name: MCP-ENABLED

Description: optional

Admin State:  Disabled  Enabled

**Figure 7-36** Creating an Interface Policy with MCP Enabled

Then you can apply the policy on relevant interface policy groups, as shown in [Figure 7-37](#).

The screenshot shows the APIC (Cisco Application Policy Infrastructure Controller) interface. The top navigation bar includes System, Tenants, Fabric (selected), Virtual Networking, L4-L7 Services, Admin, Operations, Apps, and Integrations. The user is logged in as admin.

The left sidebar under Policies shows categories like Quick Start, Switches, Modules, Interfaces, Spine Interfaces, Leaf Interfaces (selected), Profiles, and Policy Groups. Under Leaf Interfaces, the Multiplayer-Gaming-PolGrp is selected.

The main content area displays the "Leaf Access Port Policy Group - Multiplayer-Gaming-PolGrp" properties. The MCP Policy dropdown is set to MCP-ENABLED. Other settings include Link Level Policy (Speed-Auto), CDP Policy (CDP-ENABLED), CoPP Policy (select a value), LLDP Policy (LLDP-ENABLED), STP Interface Policy (select a value), Storm Control Interface Policy (select a value), and L2 Interface Policy (select a value). Buttons at the bottom include Show Usage, Reset, and Submit.

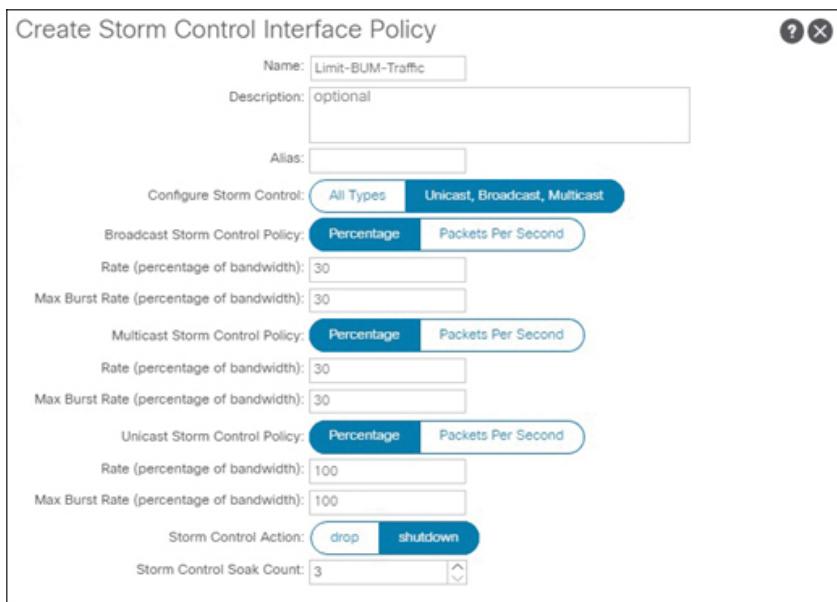
**Figure 7-37** Applying an MCP Interface Policy to an Interface Policy Group

**Key Topic**

# Configuring Storm Control

Storm control is a feature that enables ACI administrators to set thresholds for broadcast, unknown unicast, and multicast (BUM) traffic so that traffic exceeding user-defined thresholds within a 1-second interval can be suppressed. Storm control is disabled in ACI by default.

Say that for the multiplayer gaming business unit, you would like to treat all multiplayer servers with suspicion due to lack of IT visibility beyond the physical interfaces of these servers. Perhaps these servers may someday have malfunctioning network interface cards and might possibly trigger traffic storms. If the servers never need to push more than 30% of the bandwidth available to them in the form of multicast and broadcast traffic, you can enforce a maximum threshold for multicast and broadcast traffic equivalent to 30% of the bandwidth of the server interfaces. [Figure 7-38](#) shows the settings for such a storm control interface policy, configured by navigating to **Fabric > Access Policies > Policies > Interface**, right-clicking Storm Control, and selecting Create Storm Control Interface Policy.



**Figure 7-38** Configuring a Storm Control Interface Policy

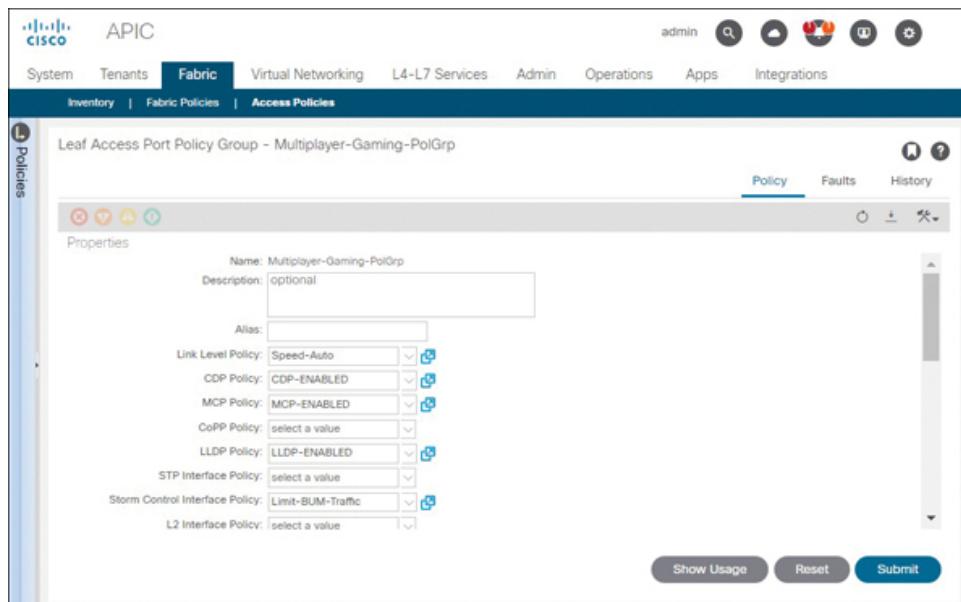
As indicated in [Figure 7-38](#), thresholds can be defined using bandwidth percentages or the number of packets traversing a switch interface (or aggregates of interfaces) per second.

The Rate parameter determines either the percentage of total port bandwidth or number of packets allowed to ingress associated ports during each 1-second interval. The Max Burst Rate, also expressed as a percentage of total port bandwidth or the number of packets entering a

switch port, is the maximum accumulation of rate that is allowed when no traffic passes. When traffic starts, all the traffic up to the accumulated rate is allowed in the first interval. In subsequent intervals, traffic is allowed only up to the configured rate.

The Storm Control Action setting determines the action ACI takes if packets continue to exceed the configured threshold for the number of intervals specified in the Storm Control Soak Count setting. In the configuration shown in [Figure 7-38](#), the Storm Control Soak Count has been kept at its default value of 3, but Storm Control Action has been set to Shutdown. This ensures that any port or port channel configured with the specified interface policy is shut down on the third second it continues to receive BUM traffic exceeding the configured rate. Storm Control Soak Count can be configured to between 3 and 10 seconds.

[Figure 7-39](#) shows that once created, a storm control interface policy needs to be applied to an interface policy group before it can be enforced at the switch interface level.



**Figure 7-39** Applying Storm Control to an Interface Policy Group

### Note

In the configurations presented in [Figure 7-38](#) and [Figure 7-39](#), the assumption is that the L2 Unknown Unicast setting on the bridge domains associated with the servers will be configured using the Hardware Proxy setting, which enables use of spine-proxy addresses for forwarding within a fabric when a destination is unknown to leaf switches. If the L2 Unknown Unicast setting for relevant bridge

domains were configured to Flood, it would be wise to also set a threshold for unicast traffic. This example shows how the storm control threshold for unicast traffic does not really come into play when Hardware Proxy is enabled on pertinent bridge domains.

## Configuring CoPP

Control Plane Policing (CoPP) protects switch control planes by limiting the amount of traffic for each protocol that can reach the control processors. A switch applies CoPP to all traffic destined to the switch itself as well as exception traffic that, for any reason, needs to be handled by control processors. CoPP helps safeguard switches against denial-of-service (DoS) attacks perpetrated either inadvertently or maliciously, thereby ensuring that switches are able to continue to process critical traffic, such as routing updates.



ACI enforces CoPP by default but also allows for tuning of policing parameters both at the switch level and at the interface level. Supported protocols for per-interface CoPP are ARP, ICMP, CDP, LLDP, LACP, BGP, Spanning Tree Protocol, BFD, and OSPF. CoPP interface policies apply to leaf ports only. Switch-level CoPP can be defined for both leaf switches and spine switches and supports a wider number of protocols.

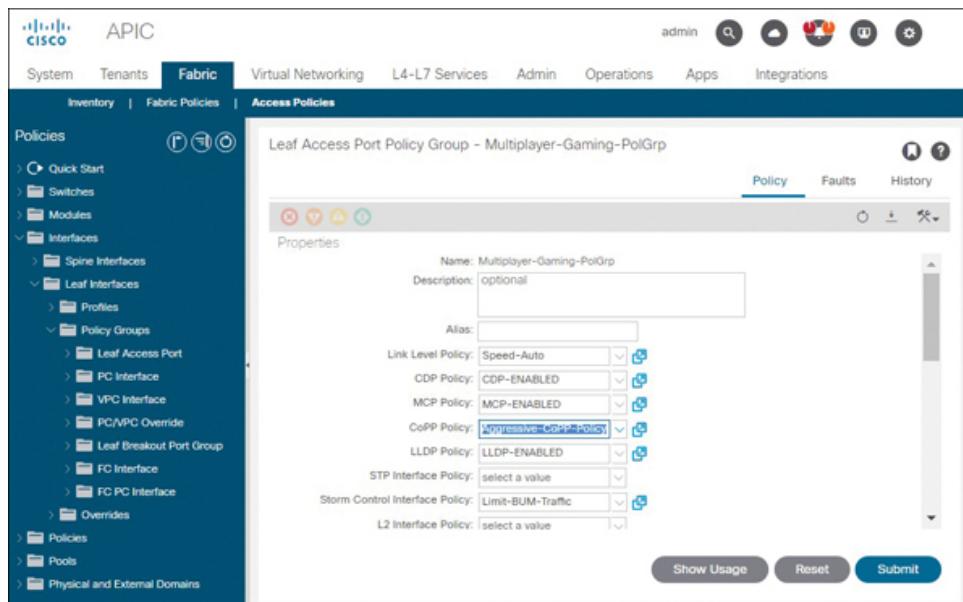
Let's say you need to ensure that multiplayer gaming servers can send only a limited number of ICMP packets to their default gateways. They should also be allowed to send only a limited number of ARP packets. This can be accomplished via a CoPP interface policy. As indicated in [Figure 7-40](#), the relevant interface policy wizard can be accessed by navigating to **Fabric > Access Policies > Interface**, right-clicking CoPP Interface, and selecting Create per Interface per Protocol CoPP Policy.



**Figure 7-40** Configuring a CoPP Interface Policy

The columns Rate and Burst in Figure 7-40 refer to Committed Information Rate (CIR) and Committed Burst (BC), respectively. The Committed Information Rate indicates the desired bandwidth allocation for a protocol, specified as a bit rate or a percentage of the link rate. The Committed Burst is the size of a traffic burst that can exceed the CIR within a given unit of time and not impact scheduling.

For the CoPP interface policy to take effect, it needs to be applied to the interface policy groups of the multiplayer gaming servers, as shown in Figure 7-41.



## Figure 7-41 Applying a CoPP Interface Policy to Interface Policy Groups

What do the default settings for CoPP on leaf switches look like? [Example 7-14](#) displays the result of the command **show copp policy** on a specific leaf switch.

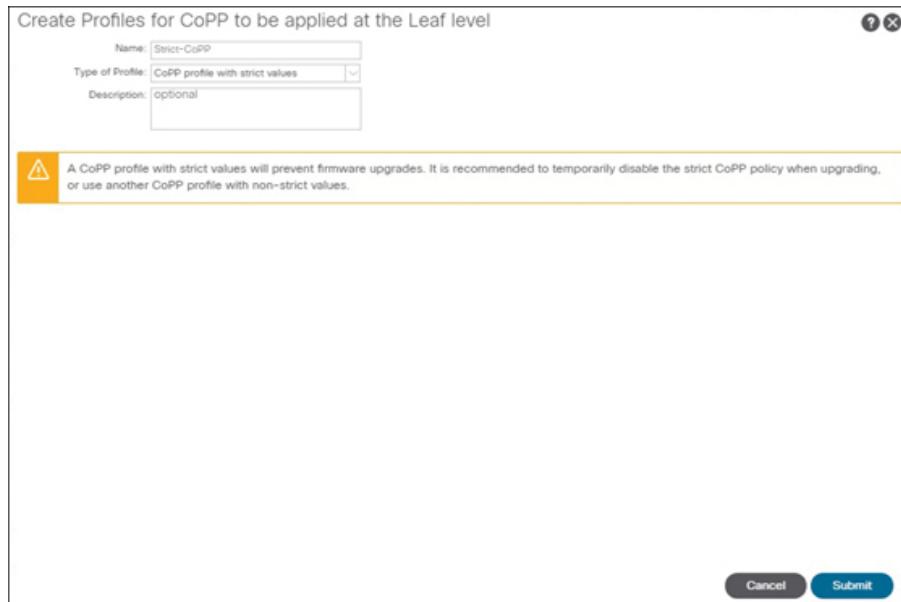
### Example 7-14 Default CoPP Settings on a Leaf Switch

[Click here to view code image](#)

LEAF101# show copp policy	COPP Class	COPP proto	COPP Rate	COPP Burst
	lldp	lldp	1000	1000
	traceroute	traceroute	500	500
	permitlog	permitlog	300	300
	nd	nd	1000	1000
	icmp	icmp	500	500
	isis	isis	1500	5000
	eigrp	eigrp	2000	2000
	arp	arp	1360	340
	cdp	cdp	1000	1000
	ifcspan	ifcspan	2000	2000
	ospf	ospf	2000	2000
	bgp	bgp	5000	5000
	tor-glean	tor-glean	100	100
	acllog	acllog	500	500
	mcp	mcp	1500	1500
	pim	pim	500	500
	igmp	igmp	1500	1500
	ifc	ifc	7000	7000
	coop	coop	5000	5000
	dhcp	dhcp	1360	340
	ifcother	ifcother	332800	5000
	infraarp	infraarp	300	300
	lacp	lacp	1000	1000
	glean	glean	100	100
	stp	stp	1000	1000

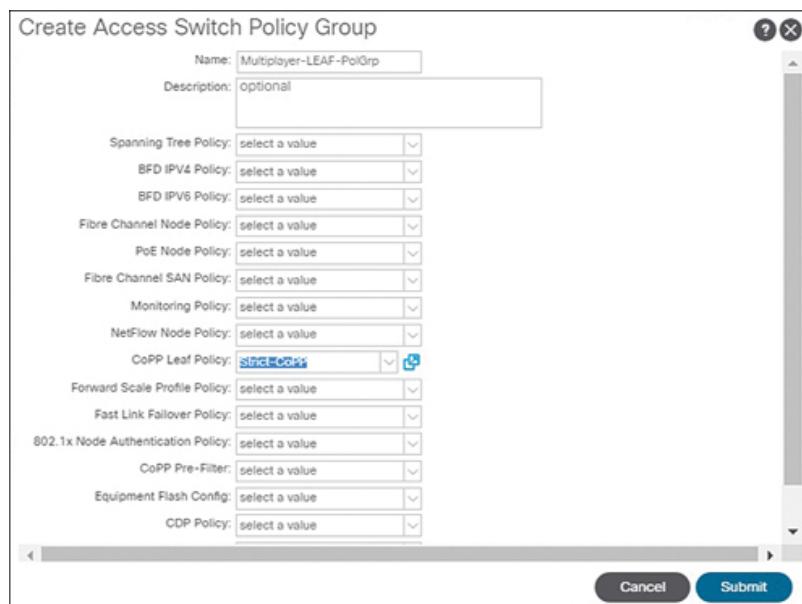
To modify the CoPP settings applied on a leaf, navigate to **Fabric > Access Policies > Policies > Switch**, right-click Leaf CoPP, and select Create Profiles for CoPP to be Applied at the Leaf Level. Notice that there are options to define custom values for each protocol, apply default CoPP values on a per-platform basis, apply permissive CoPP values, enforce strict CoPP values, and apply values between permissive and strict. [Figure](#)

**7-42** shows the selection of strict CoPP settings. Strict values can potentially impact certain operations, such as upgrades.



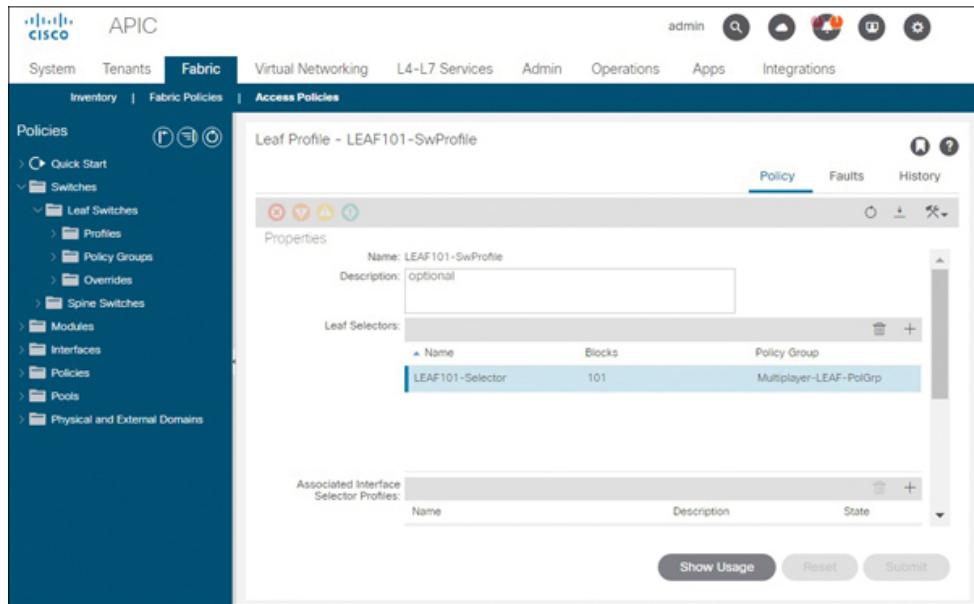
**Figure 7-42** Creating a CoPP Switch Policy That Uses Aggressively Low Values

Switch CoPP policies need to be applied to a switch policy group before they can be associated with switch profiles. **Figure 7-43** shows the creation and application of switch CoPP policies to a new switch policy group.



**Figure 7-43** Applying of a CoPP Switch Policy to a Switch Policy Group

Finally, you can allocate the CoPP switch policy group to leaf selectors referencing the intended switches as shown in [Figure 7-44](#).



**Figure 7-44** Applying a CoPP Switch Policy Group to Desired Leaf Selectors

Verification of the current CoPP settings indicates that the application of the strict CoPP policy has dramatically lowered the CoPP values to those that appear in [Example 7-15](#).

**Example 7-15** Switch CoPP Settings Following Application of Strict CoPP Values

[Click here to view code image](#)

LEAF101# show copp policy			
COPP Class	COPP proto	COPP Rate	
COPP Burst			
lldp	lldp	10	
traceroute	traceroute	10	
permitlog	permitlog	10	
nd	nd	10	
icmp	icmp	10	
isis	isis	10	
eigrp	eigrp	10	
arp	arp	10	
cdp	cdp	10	
ifcspan	ifcspan	10	

ospf	ospf	10	10
bgp	bgp	10	10
tor-glean	tor-glean	10	10
acllog	acllog	10	10
mcp	mcp	10	10
pim	pim	10	10
igmp	igmp	10	10
ifc	ifc	7000	
7000			
coop	coop	10	10
dhcp	dhcp	10	10
ifcother	ifcother	10	10
infraarp	infraarp	10	10
lacp	lacp	10	10
glean	glean	10	10
stp	stp	10	10

### Note

IFC stands for Insieme Fabric Controller. Even strict CoPP policies keep IFC values relatively high. This is important because IFC governs APIC communication with leaf and spine switches.

Another CoPP configuration option in ACI is to implement CoPP leaf and spine prefilters. CoPP prefilter switch policies are used on spine and leaf switches to filter access to authentication services based on specified sources and TCP ports with the intention of protecting against DDoS attacks. When these policies are deployed on a switch, control plane traffic is denied by default, and only the traffic specified by CoPP prefilters is permitted. Misconfiguration of CoPP prefilters, therefore, can impact connectivity within multipod configurations, to remote leaf switches, and in Cisco ACI Multi-Site deployments. For these reasons, CoPP prefilter entries are not commonly modified.

## Modifying BPDU Guard and BPDU Filter Settings

Spanning Tree Protocol bridge protocol data units (BPDUs) are critical to establishing loop-free topologies between switches. However, there is little reason for servers and appliances that do not have legitimate reasons for participating in Spanning Tree Protocol to be sending BPDUs into an ACI fabric or receiving BPDUs from the network. It is therefore best to

implement BPDU Guard and BPDU Filter on all server-facing and appliance-facing ports unless there is a legitimate reason for such devices to be participating in Spanning Tree Protocol. Although ACI does not itself participate in Spanning Tree Protocol, this idea still applies to ACI. When a BPDU arrives on a leaf port, the fabric forwards it on all ports mapped to the same EPG on which the BPDU arrived. This behavior ensures that non-ACI switches connecting to ACI at Layer 2 are able to maintain a loop-free topology.

When applied on a switch port, BPDU Filter prevents Spanning Tree Protocol BPDUs from being sent outbound on the port. BPDU Guard, on the other hand, disables a port if a Spanning Tree Protocol BPDU arrives on the port.

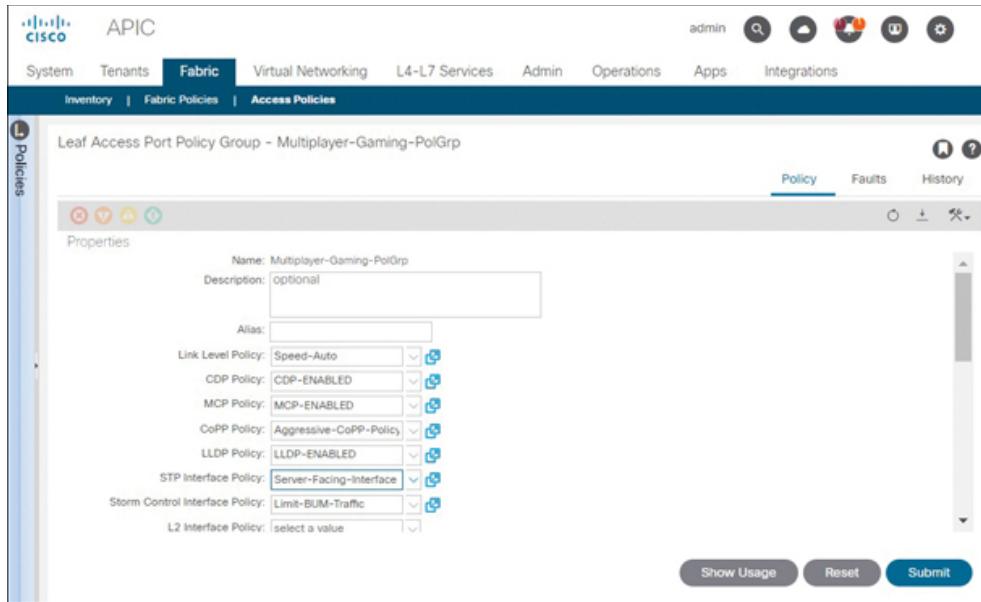
If you were concerned that a group of servers might one day be hacked and used to inject Spanning Tree Protocol BPDUs into the network with the intent of triggering changes in the Spanning Tree Protocol topology outside ACI, it would make a lot of sense to implement BPDU Filter and BPDU Guard on all ACI interfaces facing such servers.

To implement BPDU Filter and BPDU Guard, you first create a Spanning Tree Protocol interface policy with these features enabled (see [Figure 7-45](#)).

The dialog box has a title bar 'Create Spanning Tree Interface Policy' with a question mark icon and a close button. It contains four input fields: 'Name' (filled with 'Server-Facing-Interface'), 'Description' (filled with 'optional'), and 'Alias' (empty). Below these is a section 'Interface controls' with two checked checkboxes: 'BPDU filter enabled' and 'BPDU Guard enabled'. At the bottom are 'Cancel' and 'Submit' buttons.

**Figure 7-45** Creating a Spanning Tree Interface Policy

The policy should then be associated with interface policy groups for the intended servers (see [Figure 7-46](#)).



**Figure 7-46** Applying a Spanning Tree Interface Policy to Interface Policy Groups

Note that FEX ports enable BPDU Guard by default, and this behavior cannot be changed.

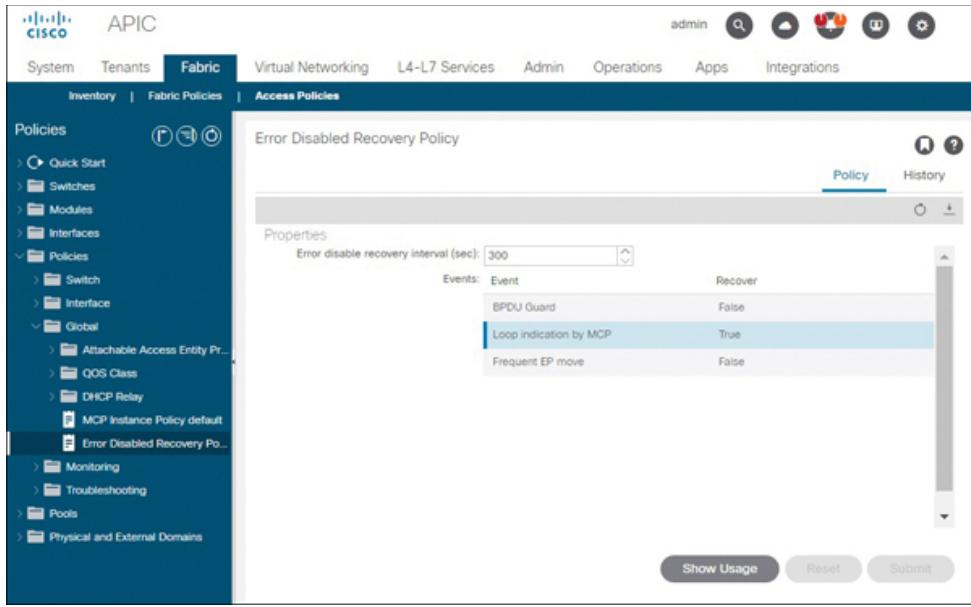
## Modifying the Error Disabled Recovery Policy

When administrators set up features like MCP and BPDU Guard and determine that ports should be error disabled as a result of ACI loop-detection events, the error disabled recovery policy can be used to control whether the fabric automatically reenables such ports after a recovery interval.

ACI can also move a port into an error-disabled state if an endpoint behind the port moves to other ports at a high frequency with low intervals between moves. The reasoning in such cases is that high numbers of endpoint moves can be symptomatic of loops.

To modify the error disabled recovery policy in a fabric, navigate to **Fabric > Access Policies > Policies > Global > Error Disabled Recovery Policy**. Figure 7-47 shows a configuration with automatic recovery of ports that have been disabled by MCP after a 300-second recovery interval.

**Key Topic**



**Figure 7-47** Editing the Error Disabled Recovery Policy

To configure whether ACI should disable ports due to frequent endpoint moves in the first place, navigate to **System > System Settings > Endpoint Controls > Ep Loop Protection**.

## Configuring Leaf Interface Overrides

A **leaf interface override** policy allows interfaces that have interface policy group assignments to apply an alternate interface policy group.

Imagine that a group of ports have been configured on Node 101, using a specific interface policy group. One of the interfaces connects to a firewall, and security policies dictate that LLDP and CDP toward the firewall need to be disabled on all firewall-facing interfaces. It might be impossible to modify the interface policy group associated with the port because it might be part of a port block. In this case, a leaf interface override can be used to assign an alternative interface policy group to the port of interest.

To implement such a leaf interface override, you create a new interface policy group with the desired settings. Then you navigate to **Fabric > Access Policies > Interfaces > Leaf Interfaces**, right-click Overrides, and select Create Leaf Interface Overrides. Set Path Type and Path to identify the desired switch interface and the new policy group that needs to be applied to the interface. [Figure 7-48](#) shows a leaf interface override configuration.

Create Leaf Interface Override

Name: Override-Interface-to-Firewall

Description: optional

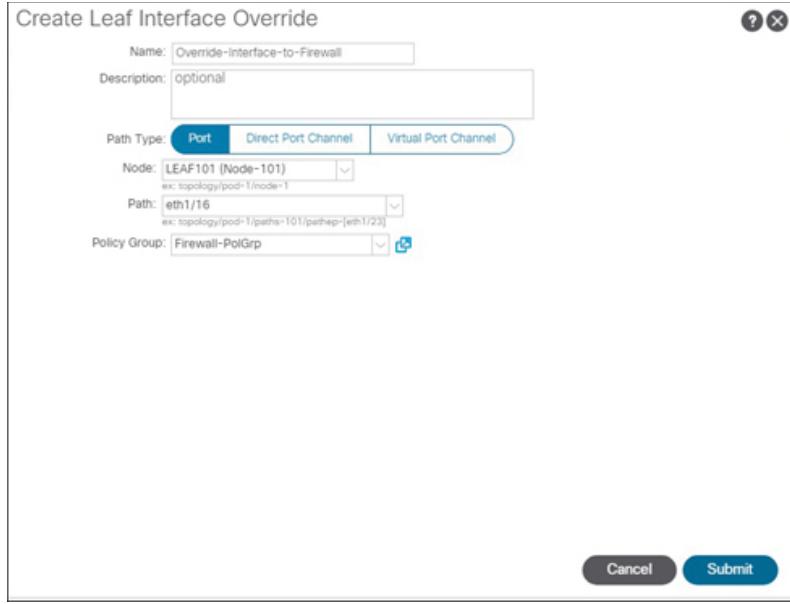
Path Type:  Port  Direct Port Channel  Virtual Port Channel

Node: LEAF101 (Node-101)  
ex: topology/pod-1/node-1

Path: eth1/16  
ex: topology/pod-1/paths-101/pathseg-[eth1/23]

Policy Group: Firewall-PolGrp

Cancel Submit



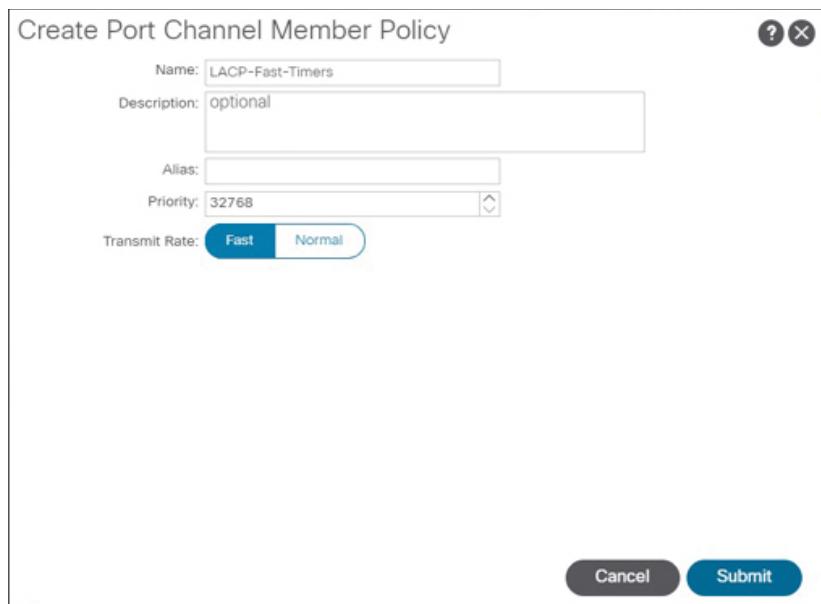
**Figure 7-48** Configuring a Leaf Interface Override

With this configuration, LLDP and CDP have been disabled on firewall-facing interface 1/16.

## Configuring Port Channel Member Overrides

When an override needs to be applied to one or more links that are part of a port channel or vPC but not necessarily the entire port channel or vPC, a **port channel member override** can be used. Examples of port channel member overrides include the implementation of LACP fast timers and the modification of LACP port priorities.

To configure a port channel member override, first configure an interface policy that will be used to override the configuration of one or more member ports. Create a port channel member policy by navigating to **Fabric > Access Policies > Policies > Interface**, right-clicking Port Channel Member, and selecting Create Port Channel Member Policy. [Figure 7-49](#) shows a policy that enables LACP fast timers.



**Figure 7-49** Configuring a Port Channel Member Policy

Note that the port priority setting in this policy has not been modified from its default. The Priority setting can be used to determine which ports should be put in standby mode and which should be active when there is a limitation preventing all compatible ports from aggregating. A higher port priority value means a lower priority for LACP.

[Example 7-16](#) shows the current timer configuration for port 1/32 on Node 101. This port has been configured as part of a port channel along with port 1/31.

**Example 7-16 Ports 1/31 and 1/32 Both Default to Normal LACP Timers**

[Click here to view code image](#)

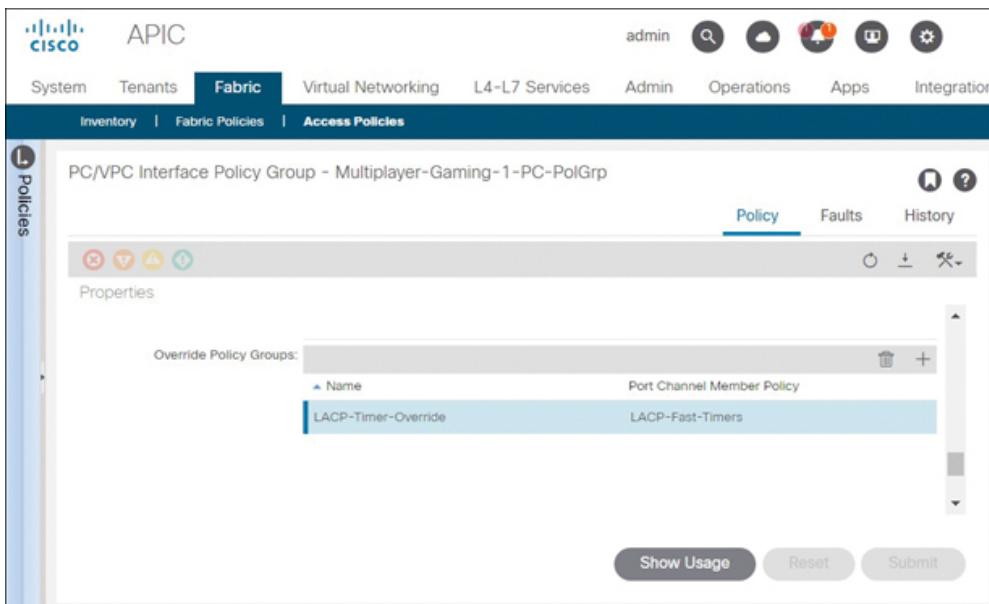
```
LEAF101# show port-channel summary interface port-channel 2
Flags:  D - Down          P - Up in port-channel (members)
        I - Individual    H - Hot-standby (LACP only)
        s - Suspended      r - Module-removed
        S - Switched       R - Routed
        U - Up (port-channel)
        M - Not in use. Min-links not met
        F - Configuration failed
-----
-- 
Group Port-      Type     Protocol Member Ports
      Channel
```

```

-- 
2    Po2(SD)      Eth      LACP      Eth1/31(P)   Eth1/32(P)
LEAF101# show lACP interface ethernet 1/31 | egrep -A8 "Local" | egrep
"Local|LACP"
Local Port: Eth1/31  MAC Address= 00-27-e3-15-bd-e3
  LACP_Activity=active
  LACP_Timeout=Long Timeout (30s)
LEAF101# show lACP interface ethernet 1/32 | egrep -A8 "Local" | egrep
"Local|LACP"
Local Port: Eth1/32  MAC Address= 00-27-e3-15-bd-e3
  LACP_Activity=active
  LACP_Timeout=Long Timeout (30s)

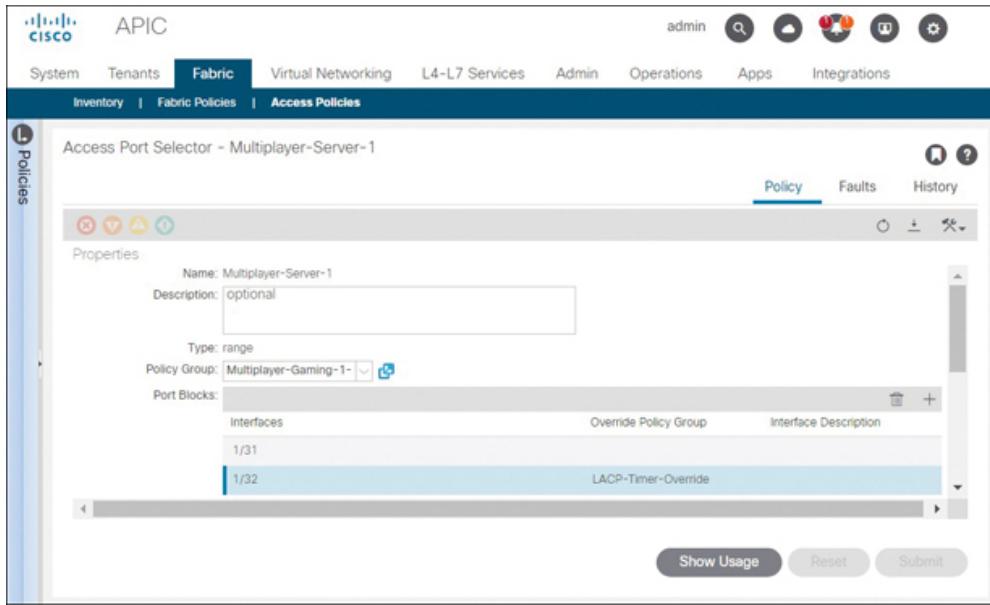
```

To apply the port channel member policy, you first associate the policy to the desired port channel or vPC interface policy group in the form of an override policy group (see [Figure 7-50](#)).



**Figure 7-50** Adding a Port Channel Member Policy to an Interface Policy Group

Next, you determine specifically which ports the override policy applies to. [Figure 7-51](#) shows the application of the policy to port 1/32. After you shut down and reenable the port, it appears to have LACP fast timers implemented. This can be confirmed in the output displayed in [Example 7-17](#).



**Figure 7-51** Applying an Override to a Member of a Port Channel

### Example 7-17 Port 1/32 Overridden Using Fast LACP Timers

[Click here to view code image](#)

```
LEAF101# show lACP interface ethernet 1/31 | egrep -A8 "Local" | egrep "Local|LACP"
Local Port: Eth1/31    MAC Address= 00-27-e3-15-bd-e3
  LACP_Activity=active
  LACP_Timeout=Long Timeout (30s)
LEAF101# show lACP interface ethernet 1/32 | egrep -A8 "Local" | egrep "Local|LACP"
Local Port: Eth1/32    MAC Address= 00-27-e3-15-bd-e3
  LACP_Activity=active
  LACP_Timeout=Short Timeout (1s)
```

## Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: [Chapter 17, “Final Preparation,”](#) and the exam simulation questions in the Pearson Test Prep Software Online.

## Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. **Table 7-5** lists these key topics and the page number on which each is found.



**Table 7-5** Key Topics for Chapter 7

Key Topic Element	Description	Page Number
Figure 7-1	Shows the settings that are available when configuring an LLDP interface policy	189
Figure 7-2	Shows the settings that are available when configuring a link level interface policy	189
Paragraph	Clarifies that the leaf access port policy group needs to be used when configuring non-aggregated ports	191
Figure 7-5	Shows a sample configuration of a leaf access port policy group	191
Figure 7-7	Shows how to map an interface policy group to switch ports via switch access port selectors	192
Figure 7-11	Shows a sample configuration of an interface policy group with LACP enabled	196

Key Topic Element	Description	Page Number
<a href="#">Table 7-3</a>	Details the common control options available for configuration of port channel interface policies	<a href="#">196</a>
<a href="#">Paragraph</a>	Addresses the extent of reusability of port channel and vPC interface policy groups	<a href="#">201</a>
<a href="#">Figure 7-18</a>	Shows how to configure vPC domains in ACI	<a href="#">203</a>
<a href="#">Paragraph</a>	Describes the result of defining a vPC explicit protection group from a forwarding perspective	<a href="#">204</a>
<a href="#">Figure 7-19</a>	Shows the configuration of a port channel interface policy with static port channeling enabled	<a href="#">205</a>
<a href="#">Figure 7-20</a>	Shows the creation of a vPC interface policy group	<a href="#">205</a>
<a href="#">Paragraph</a>	Describes the use case and benefits of AAEP EPGs	<a href="#">209</a>
<a href="#">List</a>	Lists the steps necessary for deploying a new fabric extender in ACI	<a href="#">212</a>

Key Topic Element	Description	Page Number
Paragraph	Explains the function of dynamic breakout ports	<a href="#">215</a>
<a href="#">Figure 7-30</a>	Shows a sample configuration of a leaf port breakout group	<a href="#">216</a>
<a href="#">Figure 7-31</a>	Depicts the implementation of access policies for dynamic breakout subports and the resulting port numbering convention	<a href="#">217</a>
Paragraph	Describes a common use case for implementing the Dot1p Preserve setting	<a href="#">218</a>
<a href="#">Figure 7-32</a>	Shows how to enable the Dot1p Preserve setting	<a href="#">219</a>
<a href="#">Figure 7-33</a>	Shows configuration of a DHCP relay policy in the Access Policies view	<a href="#">220</a>
<a href="#">Figure 7-34</a>	Shows the addition of a DHCP server to a DHCP relay policy in the Access Policies view	<a href="#">220</a>
Paragraph	Provides an understanding of the steps needed to implement MCP	<a href="#">221</a>

Key Topic Element	Description	Page Number
Paragraph	Explains where MCP can be globally enabled	221
<a href="#">Table 7-4</a>	Describes the configuration settings available when implementing MCP globally	222
Paragraph	Reinforces the idea that MCP needs to be enabled both globally and at the interface level	222
Paragraph	Describes the use case for storm control and its default configuration state in ACI	223
Paragraph	Describes CoPP in ACI and how CoPP can be configured	225
<a href="#">Figure 7-47</a>	Shows how to edit the error disabled recovery policy	231

## Complete Tables and Lists from Memory

Print a copy of [Appendix C, “Memory Tables”](#) (found on the companion website), or at least the section for this chapter, and complete the tables and lists from memory. [Appendix D, “Memory Tables Answer Key”](#) (also on the companion website), includes completed tables and lists you can use to check your work.

## Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

link debounce interval  
vPC peer dead interval  
dynamic breakout port  
leaf interface override  
port channel member override

# Chapter 8

## Implementing Tenant Policies

**This chapter covers the following topics:**

**ACI Endpoint Learning:** This section describes the various lookup tables available in ACI and details how ACI learns endpoints attached to the fabric.

**Packet Forwarding in ACI:** This section complements the topic of endpoint learning by examining the four major packet forwarding scenarios in ACI.

**Deploying a Multi-Tier Application:** This section walks through the deployment of tenant policies for a hypothetical application to the point of endpoint learning.

**Whitelisting Intra-VRF Communications via Contracts:** This section covers whitelisting of components of a hypothetical application using contracts.

This chapter covers the following exam topics:

- 1.6 Implement ACI logical constructs
  - 1.6.b application profile

- 1.6.d bridge domain (unicast routing, Layer 2 unknown hardware proxy, ARP flooding)
- 1.6.e endpoint groups (EPG)
- 1.6.f contracts (filter, provider, consumer, reverse port filter, VRF enforced)
- 2.1 Describe endpoint learning
- 2.2 Implement bridge domain configuration knob (unicast routing, Layer 2 unknown hardware proxy, ARP flooding)

Thus far in the book, you have learned how to configure switch ports for connectivity to servers and appliances. Next, you need to learn how to deploy the policies that enable these attached devices to communicate with one another.

The goal of this chapter is not simply to show you what buttons to push when configuring tenant objects. Rather, the objective is to convey some of the logic behind decisions that enable you to deploy applications and associated whitelisting policies more effectively. In the process, you will also learn how to verify endpoint learning and proper traffic forwarding at a basic level.

## **“Do I Know This Already?” Quiz**

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 8-1](#) lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz

questions. You can find the answers in Appendix A, “Answers to the ‘Do I Know This Already?’ Questions.”

**Table 8-1** “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
ACI Endpoint Learning	1–4
Packet Forwarding in ACI	5–7
Deploying a Multi-Tier Application	8
Whitelisting Intra-Tenant Communications via Contracts	9, 10

### Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 
- 1.** Which of the following statements about endpoint learning is correct?

    - a. ACI learns devices behind L3Outs as remote endpoints.
    - b. An ACI leaf prevents the need for flooding by learning all endpoints, including remote endpoints to which no local devices seek to communicate.
    - c. An ACI leaf learns both the MAC address and any IP addresses of any local endpoint.
    - d. An ACI leaf notifies the spine of any remote endpoints it has learned.
  - 2.** An ACI fabric has problems learning a silent host. Which of the following best explains why ACI cannot learn this endpoint?

    - a. ACI has trouble detecting silent hosts.
    - b. The silent host is in an L2 BD, which has been configured for hardware proxy.
    - c. Unicast Routing has been enabled for the BD.
    - d. An SVI has been deployed for the BD.
  - 3.** Which of the following may signal endpoint flapping?

    - a. Non-transient output in ACI suggesting that a MAC address has more than one IP address association
    - b. Non-transient output in ACI suggesting that an IP address has more than one MAC address association
    - c. Transient output in ACI suggesting that a MAC address has more than one IP address association
    - d. Transient output in ACI suggesting that an IP address has more than one MAC address association

- 4.** True or false: An endpoint learned locally by a leaf becomes the single source of truth for the entire fabric.
- a.** True
  - b.** False
- 5.** True or false: With hardware proxy configured, a leaf forwards L2 unknown unicast traffic to the spine. If the spine does not know the destination, it drops the traffic and initiates silent host detection.
- a.** True
  - b.** False
- 6.** True or false: When hardware proxy is enabled, ACI no longer needs to forward ARP traffic.
- a.** True
  - b.** False
- 7.** True or false: An ACI leaf that needs to perform flooding forwards the traffic out all uplinks, and spines, likewise, forward the traffic on all the fabric links as well until the traffic has flowed over all fabric links within the topology.
- a.** True
  - b.** False
- 8.** There appears to be a delay in forwarding when a port with a static binding comes up. Which of the following could potentially explain the reason behind this delay?
- a.** The Deployment Immediacy parameter for the static binding has been set to Immediate.
  - b.** The Port Encap parameter was set incorrectly, and ACI had to dynamically correct the issue.
  - c.** ACI has trouble learning endpoints because it has to wait for ARP packets.

- d. The Deployment Immediacy parameter for the static binding has been set to On Demand.
- 9.** An administrator has created a restrictive filter to allow any source to reach destination port 22. She now wants to create a contract using this single filter. Which setting or settings should she use to enable ACI to automatically generate a rule for return traffic from the server to client?
- a. Apply Both Directions
  - b. Established
  - c. Apply Both Directions and Reverse Filter Ports
  - d. Reverse Filter Ports
- 10.** An engineer wants to log traffic if it matches a particular criterion. How can this be achieved?
- a. Specify Log in the Actions column of the relevant filter within a contract subject.
  - b. Specify Log in the Directives column of the relevant filter within a contract subject.
  - c. Specify Log in the filter itself.
  - d. Specify Log on the Create Contract page.

## Foundation Topics

### ACI Endpoint Learning

Traditional networks rely heavily on control plane mechanisms such as Address Resolution Protocol (ARP), Gratuitous ARP (GARP), and IPv6 Neighbor Discovery (ND) to populate switch and router forwarding tables. Because of

reliance on protocols like these, traffic flooding remains a cornerstone of address resolution in most networks.

ACI takes a different approach to endpoint learning. In ACI, the emphasis is on learning all endpoint information through the data plane and in hardware. It does so through analysis of both the source MAC address and source IP address included in packets it receives. ACI still takes action based on information in address resolution packets (such as ARP requests) because reliance on the data plane alone can sometimes create an unrealistic picture of endpoint locations. But when analyzing address resolution packets, ACI does so in the data plane, without the need to use switch CPU resources.

In addition to learning endpoint information from the data plane, ACI introduces various enhancements to greatly reduce unknown devices. The idea is that if there are ways to detect unknown endpoints in a fabric, the need for flooding can be minimized.



There are three primary benefits to how ACI learns endpoints. First, data plane-focused endpoint learning is less resource intensive and therefore enables greater fabric scalability. Second, ACI fabrics are able to react to endpoint movements and update endpoint information faster than traditional networks because ACI does *not* need to wait for GARP packets. Third, the emphasis on eliminating unknown endpoints enables ACI to optimize traffic forwarding and greatly reduce packet flooding. This last benefit has a direct impact on endpoint performance.

This section describes how ACI optimizes endpoint learning.

## Lookup Tables in ACI

ACI uses endpoint data to forward traffic within the fabric. In ACI, an **endpoint** is defined as one MAC address and zero or more IP addresses associated with the MAC address.

In a traditional network, three forwarding tables are used to track devices. [Table 8-2](#) documents these tables and their purposes.

**Table 8-2** Traditional Switch Lookup Tables and Their Purposes

Table	Purpose
Routing Information Base (RIB)	Stores IPv4 and IPv6 routes to known destinations as well as the next-hop IP address to reach each destination. The RIB in traditional networks may include /32 host routes for certain interfaces, such as loopback interfaces.
MAC address table	Stores MAC addresses of Layer 2-adjacent devices and the local switch interface that needs to be used to reach the destination MAC address.

## Table Purpose

ARP table	Stores MAC-to-IP associations, allowing a switch or router to look up the MAC address it needs to encapsulate in a packet destination MAC field for receipt by a destination or next-hop device.
-----------	--

ACI also uses three tables to maintain network addresses, but these tables have different purposes and store different information compared to traditional networks. [Table 8-3](#) details the lookup tables in ACI and the function each serves.



**Table 8-3** ACI Lookup Tables and Their Purposes

## Table Purpose

Table

entry

## **Ta Purpose**

**bl  
e**

**En** Stores MAC addresses and/or IP addresses (only /32  
**dp** addresses for IPv4 and /128 addresses for IPv6). The  
**oi** endpoint table is the primary lookup table used by ACI  
**nt** leaf switches in determining how to forward traffic to  
**ta** other endpoints within the fabric.

**bl  
e**

**Ro** Stores IPv4 and IPv6 routes to destination subnets  
**ut** beyond an L3Out or within the Layer 3 domain inside the  
**in** fabric. If advertised to ACI, this may also include  
**g** external /32 host routes for IPv4 or /128 host routes for  
**Inf** IPv6. For destinations within the fabric, the RIB stores  
**or** bridge domain subnets. ACI routing tables do not store  
**m** host routes (/32 for IPv4 or /128 for IPv6) pointing to  
**ati** endpoints within the fabric, but they do include host  
**on** routes pointing to anycast subnet default gateways. ACI  
**Ba** leaf switches consult the routing table if a destination  
**se** endpoint is not found to be in the endpoint table.

**(R  
IB  
)**

## Table Purpose

bl  
e

A Stores IP-to-MAC relationships for direct neighbors RP sitting behind L3Out connections. ACI switches do not ta perform ARP table lookups when forwarding traffic to bl endpoints within the fabric.

e

When a leaf switch needs to make a decision on how to forward a packet, it first consults its endpoint table. If the destination is not in the endpoint table, it then consults the routing table. The ARP table is examined only if the destination is outside the fabric and behind an L3Out.

## Local Endpoints and Remote Endpoints

Key Topic

If an ACI leaf learns an endpoint from an access (non-fabric) port, it considers the endpoint a **local endpoint**. With this definition, an ACI leaf that learns an endpoint from a Layer 2 extension to a traditional switch would also consider the new endpoint to be a local endpoint even though the endpoint is not directly attached to the leaf. Some ACI documentation refers to non-fabric ports as front-panel ports.

**Key Topic**

If, on the other hand, a leaf learns an endpoint over a fabric port (that is, over tunnel interfaces), the leaf considers the endpoint a **remote endpoint**.

The distinction between local and remote endpoints is important first and foremost because ACI leafs store different information for endpoints depending on whether the endpoints are local or remote. If an endpoint is local to a leaf, the leaf needs to store as much information about the endpoint as possible. If an endpoint is remote, it is not always necessary for a switch to learn both its MAC address and associated IP addresses. Therefore, if local endpoints on a leaf need only intra-subnet communication with a remote endpoint, the leaf only stores the MAC address of the remote endpoint. If local endpoints need traffic routed to the remote endpoint, the leaf stores the remote endpoint's IP information. If endpoints on a leaf require both intra-subnet and inter-subnet communication with a remote endpoint, the leaf then stores both the MAC address and any IP addresses of the remote endpoint. Having leaf switches store the minimum amount of information they need enables a substantial amount of endpoint table scalability.

Another important difference between local and remote endpoints is the role each plays in the overall learning process. A leaf communicates endpoint information to the spine Council of Oracle Protocol (COOP) database only if the endpoint is local.

Finally, an additional difference between these endpoint types is the amount of time a leaf retains relevant endpoint information. If an endpoint moves to a different leaf switch, the new leaf advertises the endpoint as a local endpoint to the spine COOP database, which triggers an immediate

notification to the previous leaf and the creation of a bounce entry. Remote endpoint information, on the other hand, is more susceptible to becoming stale. Therefore, it is reasonable for leaf switches to retain local endpoint information for a longer period of time compared to remote endpoints.

[Table 8-4](#) summarizes the key differences between local and remote endpoints.



**Table 8-4** Differences Between Local and Remote Endpoints

Feature	Local Endpoint	Remote Endpoint
One endpoint	1 MAC address and $n$ IP addresses	1 MAC address or 1 IP address
Learning scope	Communicated to spine COOP database	Learned on leafs as a cache of the actual endpoint information
Endpoint retention timer	900 seconds (by default)	300 seconds (by default)

# **Understanding Local Endpoint Learning**

An ACI leaf follows a simple process to learn a local endpoint MAC and its associated IP addresses:

- Step 1.** The leaf receives a packet with source MAC address  $X$  and source IP address  $Y$ .
- Step 2.** The leaf learns MAC address  $X$  as a local endpoint.
- Step 3.** The leaf learns IP address  $Y$  and ties it to MAC address  $X$  only if the packet is either an ARP packet or a routed packet.

Once a leaf learns a local endpoint, it communicates the endpoint information to spines via a protocol called ZeroMQ. The spine switches receive endpoint data using COOP.

It should be apparent by now that one major difference between ACI and traditional endpoint learning is that ACI can also learn source IP addresses from the data plane, even if the packets are not ARP packets. In traditional networks, switches trust only specific packet types, such as ARP and GARP, when learning device IP information.

## **Unicast Routing and Its Impact on Endpoint Learning**

The ACI local endpoint learning process requires that a packet entering a switch port be either a routed packet or an ARP packet for ACI to learn the IP address of the transmitting system. However, there are multiple configuration knobs that may prevent ACI from learning IP addresses.

**Key Topic**

One such BD configuration knob is the Unicast Routing setting. If this setting has been disabled for a bridge domain, ACI does not learn IP addresses for endpoints within EPGs associated with the BD. The reason for this is that if ACI is not expected to route traffic for a BD, there should be no need for ACI to analyze ARP packets for their source IP addresses in the first place.

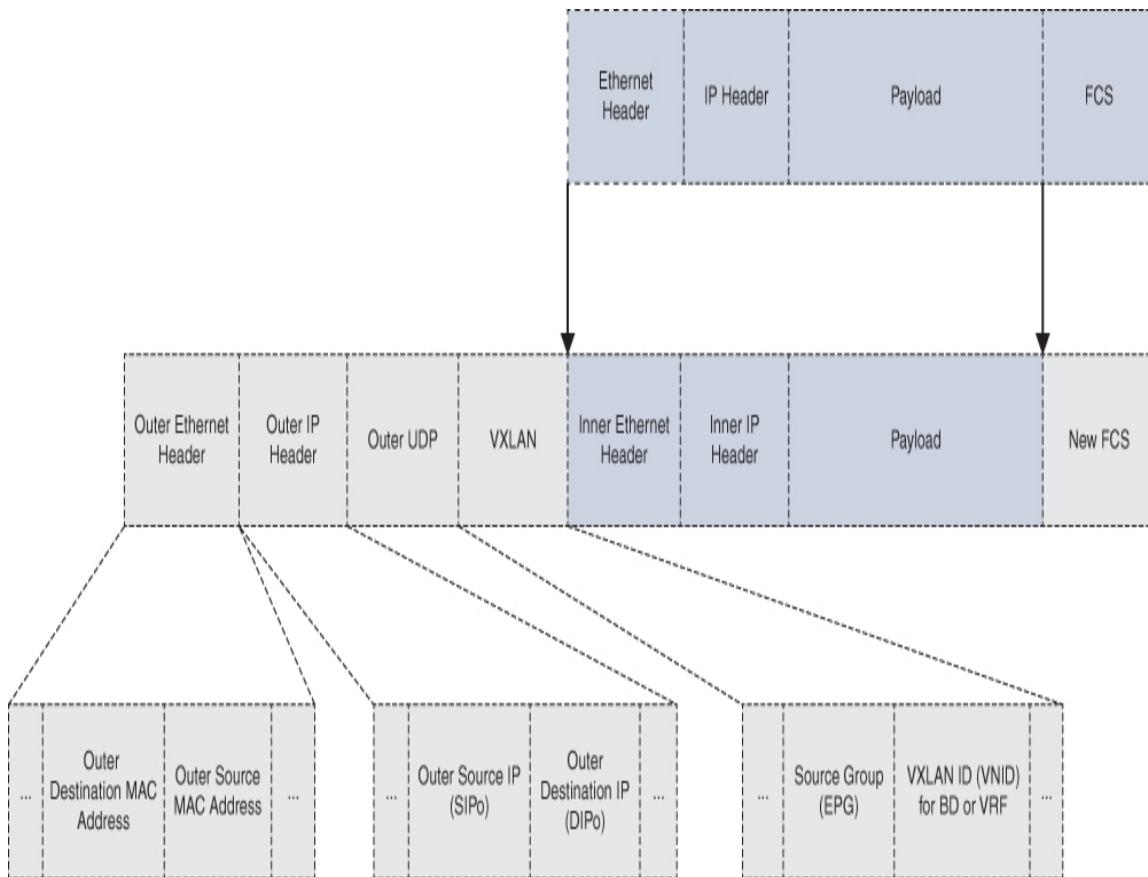
Unicast Routing is also cornerstone to some other endpoint learning optimizations concerning silent hosts that are covered later in this chapter.

Note that Unicast Routing is not the only ACI configuration setting that can prevent IP learning. Other BD configuration knobs that can impact IP address learning include Limit IP Learning to Subnet and Endpoint Data Plane Learning. There are further fabric-level and VRF instance-level configuration options that can also impact endpoint IP learning.

## **Understanding Remote Endpoint Learning**

Like local endpoint learning, remote endpoint learning occurs in the data plane.

When an ACI leaf switch needs to forward traffic to spines or other leafs in the fabric, it encapsulates the packet in VXLAN. [Figure 8-1](#) provides a high-level view of the ACI VXLAN packet format. Only the fields most significant for understanding how ACI forwards traffic are depicted; headers shown are not meant to represent realistic sizes.



**Figure 8-1 ACI VXLAN Packet Format**

The top portion of the figure represents the original Ethernet frame sent by an endpoint. Once the local leaf makes the decision to forward the traffic across the fabric, it adds a VXLAN header, an outer UDP header, an outer IP header, and an outer MAC header to the packet and calculates a new frame check sequence (FCS). The middle portion of the figure reflects the frame after VXLAN encapsulation.

The ACI VXLAN header includes fields for a VXLAN network identifier (VNID) and a source EPG identifier so that a receiving switch can determine the EPG to which a packet belongs. Details about the UDP header are not significant for the purpose of understanding endpoint learning and basic forwarding. The outer IP header includes fields for the source TEP and destination TEP addresses for the traffic.

The outer MAC header contains fields for a source MAC address and a destination MAC address, among other data. (This book places no further emphasis on the MAC header since its analysis would contribute very little to understanding forwarding in a routed fabric.)

### Note

In addition to TEP addresses covered in [Chapter 3](#), “[Initializing an ACI Fabric](#),” this chapter introduces a type of VTEP called vPC VIP addresses. These tunnel endpoint IP addresses also reside in the infra tenant in the overlay-1 VRF instance and are also sometimes placed in the source and destination outer IP address fields for cross-fabric forwarding.

If the leaf sending traffic into the fabric knows which leaf the traffic needs to be sent to, it can forward the traffic directly to the destination leaf by populating the destination leaf PTEP in the outer destination IP header. If the leaf does not know where to forward the traffic, it can either flood the traffic or forward it to one of the anycast spine proxy TEP addresses. When forwarding to the spine proxy, the COOP database determines where the traffic needs to be sent next. The L2 Unknown Unicast setting and a host of other settings on the respective bridge domain determine whether traffic should be flooded or forwarded to the spine proxy TEP.

Once the traffic arrives at the destination leaf, the leaf verifies that it is the intended destination leaf by checking the value in the outer destination IP field. It then decapsulates the extra headers. From the outer source IP field, the destination leaf knows which leaf is local to the source endpoint because it has a record of all tunnel IP addresses in the fabric. From the source EPG field, it knows

to which EPG the source endpoint belongs. It then uses data in the VNID field to determine whether to cache the endpoint MAC address or IP address.

ACI allocates VNIDs to VRF instances, bridge domains, and EPGs. Because each VRF instance represents a Layer 3 domain, inclusion of a VRF instance VNID in the VXLAN header communicates to the destination leaf that the forwarding represents a routing operation. If the VNID included in the VXLAN header is a bridge domain VNID, the destination leaf understands that the forwarding represents a switching operation.

In summary, an ACI leaf follows these steps to learn a remote endpoint's MAC or IP address:



**Step 1.** The leaf receives a packet with source MAC address  $X$  and source IP address  $Y$  on a fabric port.

**Step 2.** If the forwarding represents a switching operation, the leaf learns source MAC address  $X$  as a remote endpoint.

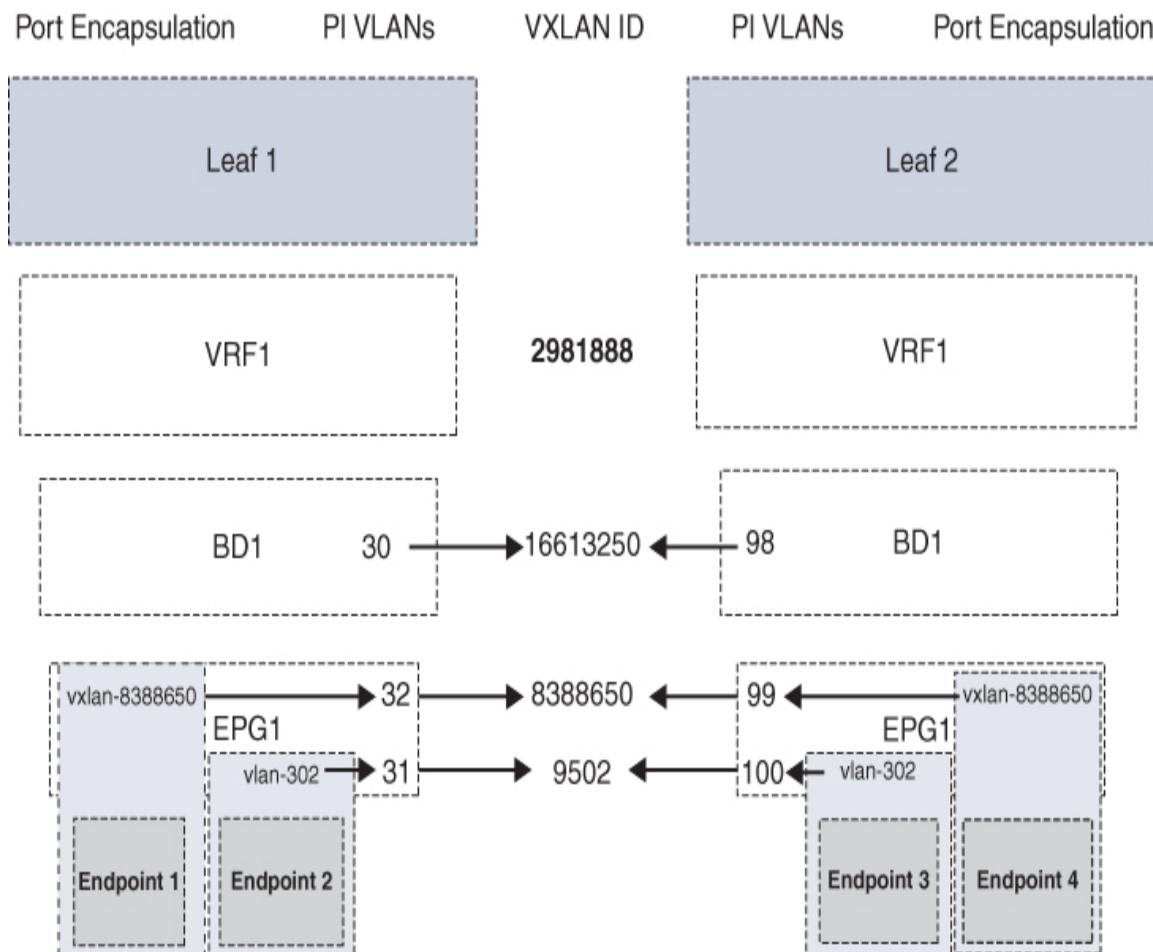
**Step 3.** If the forwarding represents a routing operation, the leaf learns IP address  $Y$  as a remote endpoint.

### Note

One benefit of ACI remote endpoint learning is that once a leaf learns a remote endpoint, it can address packets directly to the destination leaf TEP. This ensures that ACI does not use spine processor resources for forwarding between known endpoints.

# Understanding the Use of VLAN IDs and VNIDs in ACI

Look under the hood in ACI, and you will find a range of VLAN types. Add VXLAN to the mix, and you have a recipe for ultimate confusion. [Figure 8-2](#) addresses some common VLAN types in ACI and how they correlate with VXLAN.



**Figure 8-2 Basic VLAN Types in ACI**



One type of VLAN depicted in this figure is **port encapsulation VLANs**, sometimes called access

encapsulation VLANs. These are the VLAN IDs an administrator uses when mapping an EPG to a switch port. Both static path mapping and AAEP EPG assignments are of this VLAN type. The term *port encapsulation* implies that the VLAN encapsulation used appears on the wire. Of course, ACI does not include the encapsulation on the wire when mapping an EPG to a port untagged.

In truth, the term *port encapsulation* does not only refer to VLANs. ACI can encapsulate VNIDs on the wire over trunk links to certain virtual switching environments to further extend the fabric. This is why VXLAN ID 8388650 also appears as a port encapsulation VLAN in [Figure 8-2](#).

The second type of VLAN called out in the figure is ***platform-independent VLANs (PI VLANs)***. These are VLAN IDs that are locally significant to each leaf switch and represent a bridge domain or EPG for internal operations. Each PI VLAN maps to a VNID, although not all VNIDs map to PI VLANs. Because PI VLANs are locally significant, they cannot be used in the forwarding of traffic.

Finally, *VNIDs*, or VXLAN IDs, are allocated by APICs for VRF instances, BDs, and EPGs. The VNIDs are globally unique within a fabric and are used for forwarding purposes.

[Example 8-1](#) illustrates some of these concepts using output of a slight variation of the **show vlan extended** command. The leftmost column that includes VLAN IDs 30 and 31 lists PI VLANs. The column titled Encap shows port encapsulation VLAN IDs or VNIDs. Each line in the output represents a mapping between internal VLANs and VNIDs. Finally, multiple commands enable verification of VRF instance VNIDs, such as the **show vrf <tenant:vrf> detail extended** command.

**Example 8-1 Leaf Output Showing PI VLANs, Port Encapsulations, and VRF Instance VNIDs**

[Click here to view code image](#)

```
LEAF102# show vlan id 30,31 extended

VLAN      Name                           Encap
Ports

-----
-----  

30        Production:Multiplayer-Servers-BD    vxlan-
16613250   Eth1/38, Po2
31        Production:3rd-Party:Servers-EPG      vlan-302
Eth1/38, Po2

LEAF102# show vrf Production:MP detail extended | grep vxlan
Encap: vxlan-2981888
```

### Note

There are several additional VLAN ID terms you may come across in ACI that may be important for advanced troubleshooting purposes but that are not covered on the DCACI 300-620 exam.

## Endpoint Movements Within an ACI Fabric

Endpoints may move between Cisco ACI leaf switches as a result of a failover event or a virtual machine migration in a hypervisor environment.

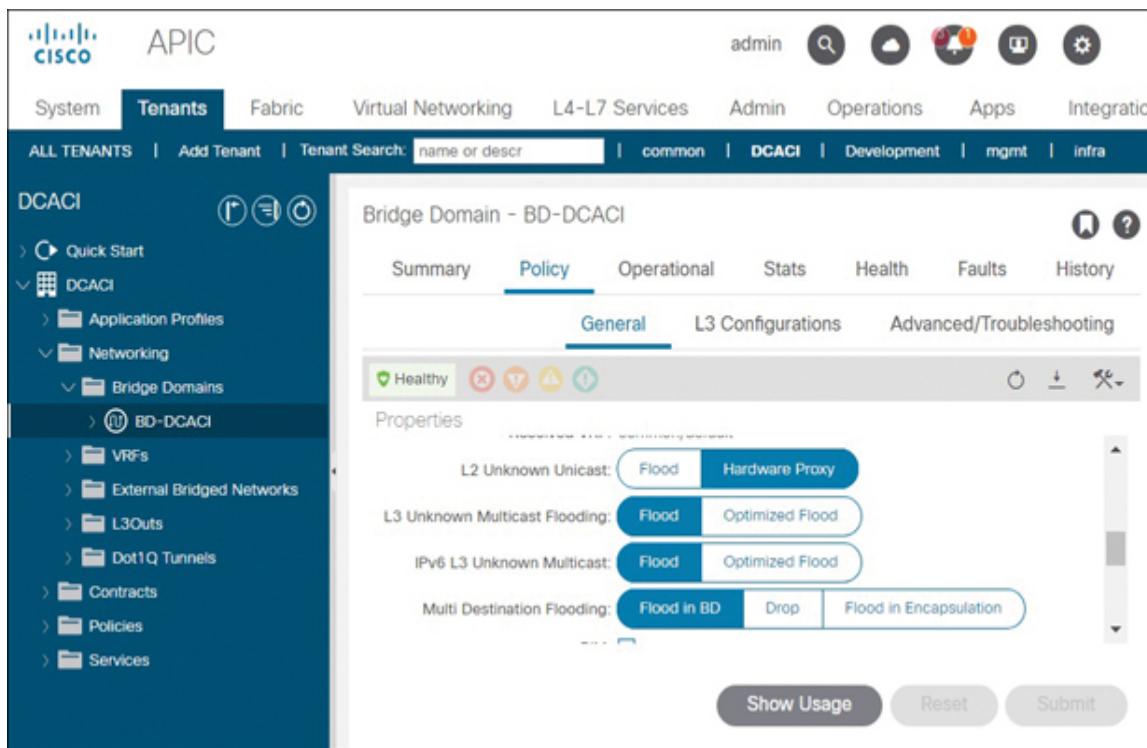
When a new local endpoint is detected on a leaf, the leaf updates the COOP database on spine switches with its new local endpoint information. If the COOP database already learned the same endpoint from another leaf, COOP

recognizes this event as an endpoint move and reports this move to the original leaf that advertised the old endpoint information. The old leaf that receives this notification deletes its old endpoint entry and creates a *bounce entry* pointing to the new leaf. A bounce entry is basically a remote endpoint created by COOP communication instead of data plane learning. With a bounce entry in place, a leaf is able to swap the outer destination IP address of packets destined to the endpoint that moved without having to notify the sending leaf via control plane mechanisms.

The advantage of this approach is scalability. No matter how many leaf switches have learned the endpoint, only three components need to be updated after an endpoint moves: the COOP database, the endpoint table on the new leaf switch to which the endpoint has moved, and the endpoint table on the old leaf switch from which the endpoint has moved. Eventually, all other leaf switches in the fabric update their information about the location of the endpoint through data plane learning.

## **Understanding Hardware Proxy and Spine Proxy**

One of the major benefits of ACI that this book has alluded to is the use of the bridge domain L2 Unknown Unicast Hardware Proxy setting as a means for optimizing forwarding and minimizing flooding within ACI fabrics. This setting can be configured in the General tab under bridge domains, as shown in [Figure 8-3](#).



**Figure 8-3** *Hardware Proxy and Flood as Possible L2 Unknown Unicast Settings*

Under this optimized forwarding scheme, a leaf that receives a packet intended for an unknown destination in the fabric can populate the outer destination IP header with a spine proxy TEP address to allow the spine COOP database to forward the traffic onto its destination. This behavior on the part of leaf switches is sometimes called a *zero-penalty forwarding decision* because the leaf has nothing to lose by sending traffic to the spine, given that the spines are the best candidates for forwarding the traffic onward to its destination anyway.

### Key Topic

When a spine receives a packet destined to its spine proxy address, it knows it needs to perform some action on the

traffic. It checks the destination against its COOP database. If the spine also acknowledges the destination to be unknown, it then drops the traffic.

### Note

Do not let placement of hardware proxy in this section on endpoint learning become a source of confusion. Hardware proxy relates to forwarding, not endpoint learning. However, there are endpoint learning considerations when configuring bridge domains for hardware proxy.

## Endpoint Learning Considerations for Silent Hosts

One problem with hardware proxy logic becomes apparent in the analysis of silent hosts.

A *silent host* is a server or virtual machine that prefers to remain silent until called upon. Because silent hosts, by definition, do not initiate data plane communication, ACI may not be able to learn such endpoints through data plane mechanisms alone. If ACI is to be able to eliminate unknown endpoints, it needs to have methods to detect silent hosts—and that it does.

To detect a silent host, ACI attempts to “tickle” it into sending traffic to then learn it in the data plane. Upon dropping traffic toward an IP address that is not found in the COOP database, spines trigger this tickle effect by prompting leaf nodes that have programmed SVIs for the destination bridge domain to send ARP requests toward the unknown IP address. This process is called **ARP gleanning**, or *silent host detection*.

ARP gleaning works best when there is a BD SVI from which to generate ARP requests. This means that the destination subnet should be defined, and Unicast Routing should be enabled on the destination BD.

If the default gateway for a bridge domain is outside ACI, the bridge domain's L2 Unknown Unicast parameter should not be set to Hardware Proxy. Instead, it should be set to Flood to ensure that the network can learn about potential silent hosts through regular ARP flooding.

## **Where Data Plane IP Learning Breaks Down**

Optimized endpoint learning in ACI works perfectly when all the endpoints in an ACI fabric are servers and do not perform any routing. Problems sometimes occur when devices that route traffic between subnets are placed into an ACI fabric in ways that were not originally intended.

The next few subsections deal with instances in which data plane learning can lead to suboptimal situations.

## **Endpoint Learning on L3Outs**

If ACI were to use data plane IP learning to record each MAC address that lives directly behind an L3Out and associate with these MAC addresses all the external IP addresses that communicate into the fabric, the ACI endpoint table would conceivably grow exponentially to encompass all IP addresses in the Internet. However, this would never actually happen because ACI places a cap on the number of IP addresses that can be associated with a MAC address. Even so, it is likely that the endpoint table would quickly grow beyond the capabilities of the physical switch hardware.

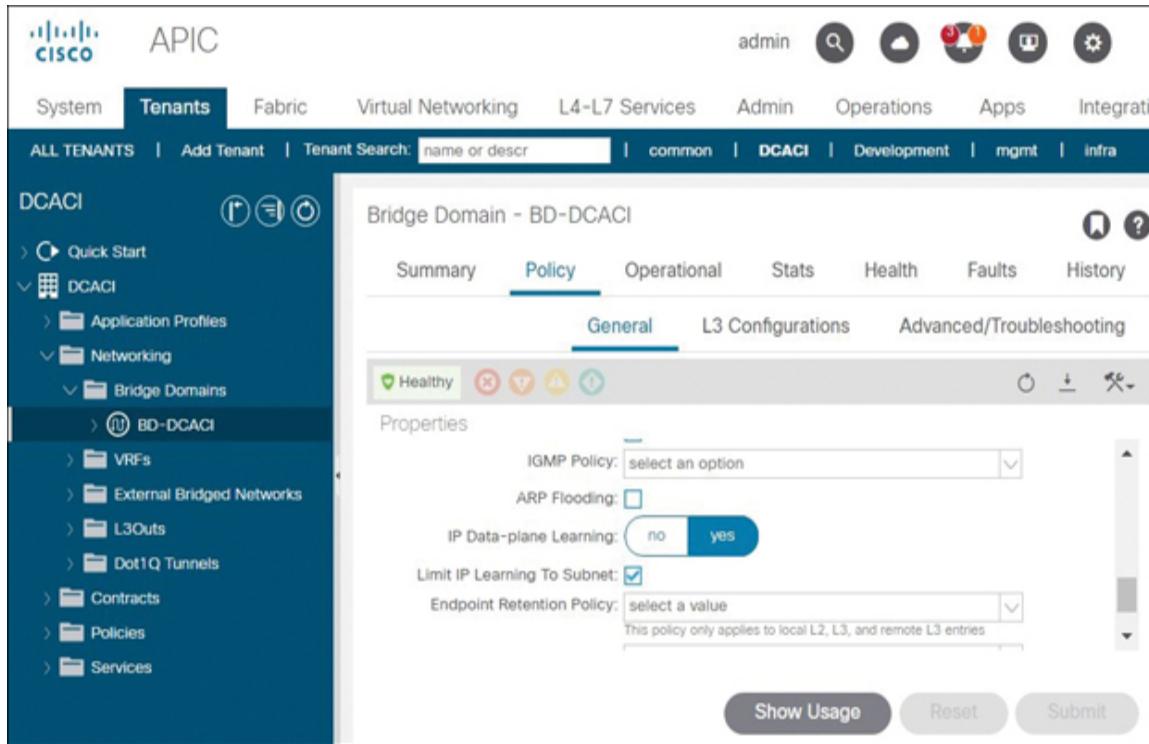
**Key  
Topic**

To optimize the endpoint table size, ACI learns only the source MAC addresses (and not source IP addresses) from data plane packets that arrive on an L3Out. ACI then uses ARP to resolve next-hop IP-to-MAC address information it needs to send traffic out L3Outs.

In summary, information in the ACI routing table and ARP table are all ACI needs to forward traffic out an ACI fabric. There is no technical benefit for ACI to learn /32 IP addresses for all endpoints outside the fabric.

## **Limiting IP Learning to a Subnet**

There might be times when a device is placed in an EPG as a result of a misconfiguration. ACI may then learn the endpoint in the data plane, even if the endpoint has an IP address in a different subnet than that of the EPG and bridge domain in which it has been configured. The Limit IP Learning to Subnet bridge domain setting prevents unnecessary learning of IP addresses when an endpoint IP address does not fall into a subnet that has been defined on the bridge domain (see [Figure 8-4](#)).



**Figure 8-4 Enabling the Limit IP Learning to Subnet Feature**

### Note

Limit IP Learning to Subnet does not prevent the learning of MAC addresses.

## Understanding Enforce Subnet Check

The Limit IP Learning to Subnet BD setting prevents the local learning of endpoint IP addresses if the endpoint is in a subnet other than those configured on a bridge domain, but it does *not* prevent potential erroneous learning of the endpoint MAC address.

The Enforce Subnet Check feature, on the other hand, ensures that an IP address and MAC address are learned as a new local endpoint *only if* the source IP address of the

incoming packet belongs to one of the ingress bridge domain subnets.

### Note

Regardless of the source IP range, the Cisco ACI leaf still learns the MAC address if the forwarding represents a switching operation.

Enforce Subnet Check, which is enabled at the VRF instance level, to an extent supersedes Limit IP Learning to Subnet and enables slightly stronger checks against bridge domain subnets. It is, therefore, a beneficial tool in preventing IP spoofing.

## Disabling Data Plane Endpoint Learning on a Bridge Domain

In general, there is very little downside to ACI data plane learning behavior. There *are* times, however, when it may be valid to turn off data plane learning altogether for a particular bridge domain. There are also times when it makes sense to dumb down ACI a little to accommodate devices with special forwarding or failover needs.

For example, say that you have connected a pair of firewalls to the fabric and have created service graphs. You have created a bridge domain for IP connectivity between the firewalls and the fabric and wish to punt specific traffic to these firewalls. This is the definition of a service graph with policy-based redirect (PBR). But what happens if the firewalls send the traffic back to ACI, and the recipient leaf connecting to the firewalls thinks it has now locally learned the endpoint that generated the traffic?

For this PBR use case, the Endpoint Data Plane Learning setting (shown in [Figure 8-4](#)) can be disabled on the bridge domain connecting the leaf and firewalls together to signal to the leaf not to learn any endpoints in this special-use bridge domain. Note that, as of the time of writing, the PBR use case is the only valid instance for disabling data plane learning at the bridge domain level. This configuration setting prevents both MAC and IP address learning.

## **Disabling IP Data Plane Learning at the VRF Level**

In some environments, there may be challenges with alternative vendor solutions that indirectly impact data plane endpoint learning in ACI. For example, some load-balancer platforms tend to send TCP resets after an active/standby failover has taken place under certain conditions. If these TCP resets are sent by the formerly active node using the source IP address of the then-active node, ACI data plane learning interprets the erroneously sourced TCP reset as an endpoint move. In such cases, this could lead to service disruption.

The best way to approach problems like these is to attach offending devices like load balancers and firewalls to ACI by using service graphs. But if a customer has concerns about service graphs, IP Data Plane Learning can be set to Disabled at the VRF level. With this setting, ACI acts like a traditional network in terms of endpoint learning: ACI continues to learn MAC addresses via the data plane, but it learns IP addresses only via traditional mechanisms, such as ARP, GARP, and ND.

## **Packet Forwarding in ACI**

Even though traffic forwarding is not called out on the Implementing Cisco Application Centric Infrastructure DCACI 300-620 exam blueprint, bridge domain configuration knobs such as L2 Unknown Unicast Flood and Hardware Proxy as well as Unicast Routing and ARP Flooding do appear on the blueprint.

The four major ACI forwarding scenarios in this section help you better understand the difference between these settings and, more generally, better grasp how ACI works.

## **Forwarding Scenario 1: Both Endpoints Attach to the Same Leaf**

If both the source and destination endpoints attempting to communicate with one another attach to the same leaf switch, the leaf considers the two endpoints to be local endpoints.

In this scenario, the forwarding is very similar to how Layer 2 and Layer 3 forwarding takes place in traditional networks. With ACI, the leaf performs a lookup in the endpoint table and forwards the traffic to the destination switch port. This, of course, assumes that the destination endpoint has been learned properly and that the traffic flow has been whitelisted.

Note that an ACI leaf does not need to encapsulate packets in VXLAN if the destination is local. Packets need to be encapsulated in VXLAN only if they need to be forwarded to a VTEP associated with another fabric node.

[Example 8-2](#) shows that two endpoints with IP addresses 10.233.58.20 and 10.233.58.32 have been learned locally on LEAF102. This is evident because the endpoints appear with the letter L in the fourth column. The output also shows that they have different port encapsulation and PI VLANs,

implying that they are likely to be in different EPGs. As long as the necessary contracts have been applied to the respective EPGs, these two endpoints should have no problem communicating with one another.

### **Example 8-2 Verifying Endpoint Learning and the Path Between Locally Attached Endpoints**

[Click here to view code image](#)

```
LEAF102# show endpoint ip 10.233.58.20

Legend:
  s - arp          H - vtep          V - vpc-
  attached      p - peer-aged
  R - peer-attached-rl   B - bounce          S - static
  M - span
  D - bounce-to-proxy    O - peer-attached      a - local-
  aged        m - svc-mgr
  L - local
  service
  +-----+-----+-----+
  -----+-----+
  VLAN/          Encap          MAC Address
  MAC Info/    Interface
  Domain          VLAN          IP Address
  IP Info
  +-----+-----+-----+
  -----+-----+
  76            vlan-3171        0050.56b7.c60a    L
  eth1/46
  Prod:Temp      vlan-3171        10.233.58.20    L
  eth1/46
```

```
LEAF102# show endpoint ip 10.233.58.32
```

Legend:

s - arp attached	H - vtep p - peer-aged	V - vpc-
R - peer-attached-rl	B - bounce	S - static
M - span		
D - bounce-to-proxy	O - peer-attached	a - local-aged
m - svc-mgr		
L - local service		E - shared-
+-----+-----+-----+	+-----+-----+-----+	+-----+-----+
VLAN/ MAC Info/	Encap	MAC Address
Interface		
Domain	VLAN	IP Address
IP Info		
+-----+-----+-----+	+-----+-----+-----+	+-----+-----+
-+-----+-----+	-+-----+-----+	-+-----+-----+
73	vlan-3169	0050.56b7.751d
eth1/45		L
Prod:Temp	vlan-3169	10.233.58.32
eth1/45		L

## Understanding Pervasive Gateways

Sometimes you might find that ACI has not learned a local endpoint or the IP address associated with the endpoint. If this is the case and the default gateway for the relevant bridge domain is in the fabric, it helps to verify that ACI has programmed a pervasive gateway for the endpoint subnet on the intended leaf. When you deploy a static path mapping to a switch port on a leaf, this action should be sufficient for the leaf to deploy a pervasive gateway for the bridge domain subnet unless the EPG domain assignment or underlying access policies for the port have been misconfigured.

**Key Topic**

A **pervasive gateway** is an anycast default gateway that ACI leaf switches install to allow local endpoints to communicate beyond their local subnets. The benefit of this anycast function is that each top-of-rack leaf switch is able to serve as the default gateway for all locally attached endpoints. A pervasive gateway is deployed as a bridge domain SVI and appears in the leaf routing table as a local host route.

Example 8-3 shows pervasive gateways for a particular bridge domain deployed on two switches with hostnames LEAF101 and LEAF102. The distributed nature of pervasive gateways is a key component of how ACI is optimized for east-west traffic flows.

**Example 8-3 Output Reflecting Installation of Pervasive Gateway on Relevant Leafs**

[Click here to view code image](#)

```
LEAF101# show ip int brief vrf Prod:Temp
IP Interface Status for VRF " Prod:Temp"(23)
Interface          Address          Interface Status
vlan72            10.233.58.1/24    protocol-up/link-
                           up/admin-up

LEAF101# show ip route 10.233.58.1 vrf Prod:Temp
10.233.58.1/32, ubest/mbest: 1/0, attached, pervasive
  *via 10.233.58.1, vlan72, [0/0], 05w0ld, local, local

LEAF102# show ip int brief vrf Prod:Temp
IP Interface Status for VRF "Prod:Temp"(25)
Interface          Address          Interface Status
```

```
vlan65          10.233.58.1/24      protocol-up/link-
up/admin-up
```

```
LEAF102# show ip route 10.233.58.1 vrf Prod:Temp
10.233.58.1/32, ubest/mbest: 1/0, attached, pervasive
*via 10.233.58.1, vlan65, [0/0], 05w0ld, local, local
```

In case the reference to VLANs 72 and 65 seems confusing, take a look at [Example 8-4](#), which shows output from both switches. These two VLAN IDs are PI VLANs for a bridge domain deployed on LEAF101 and LEAF102, respectively. You know that both of these PI VLANs reference the same global object because the VNIDs are the same.

#### **Example 8-4 Verifying the Relationship Between PI VLANs and VNIDs**

[Click here to view code image](#)

```
LEAF101# show vlan id 72 extended
```

VLAN	Name	Encap
Ports		
-----		
-----		
72	Prod:BD-Temp	vxlan-15826916
Eth1/45, Eth1/46		

```
LEAF102# show vlan id 65 extended
```

VLAN	Name	Encap
Ports		
-----		
-----		
65	Prod:BD-Temp	vxlan-15826916
Eth1/45, Eth1/46		

It is important to reiterate that a pervasive gateway for a bridge domain can be installed on any number of leaf switches within a fabric as long as relevant EPG mappings exist on the leaf. Scalability dictates that ACI does not deploy policy where it is not needed.

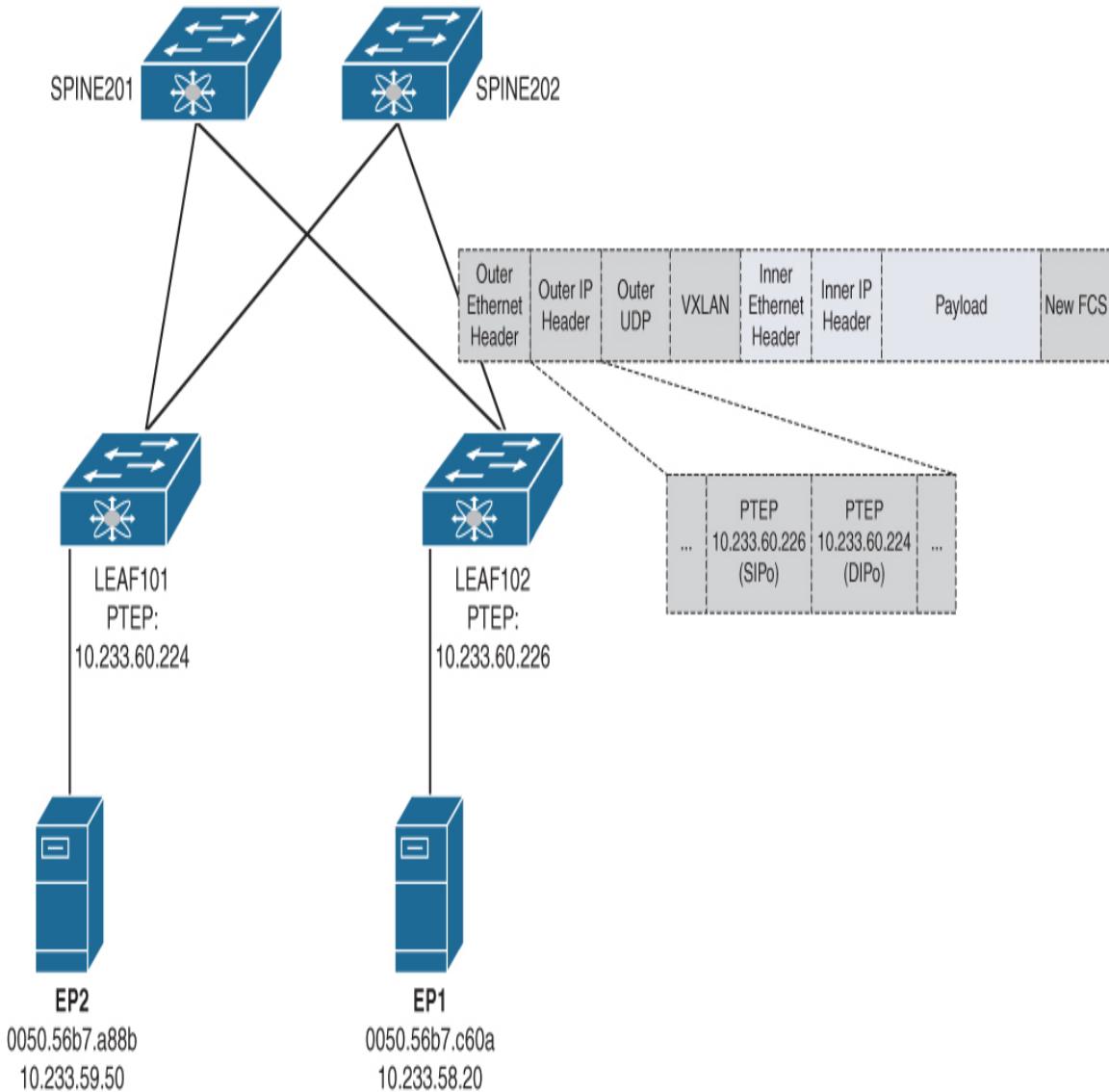
**Note**

Pervasive gateways are never deployed on spines; they are only deployed on leaf switches.

If a bridge domain has more than one subnet assignment, additional subnet gateways appear as secondary IP addresses associated with the same BD SVI.

## **Forwarding Scenario 2: Known Destination Behind Another Leaf**

Suppose endpoint EP1 local to LEAF102 wants to communicate with EP2 on LEAF101, as shown in [Figure 8-5](#). To make this forwarding possible, LEAF102 needs to place a tunnel endpoint IP address corresponding to LEAF101 in the Outer Destination IP field of the VXLAN encapsulation and send it on its way. Because EP1 only connects to LEAF102, the Outer Source IP field would contain the LEAF102 PTEP and not a vPC VIP.



**Figure 8-5** Cross-Leaf Communication Between Known Endpoints

## Verifying the Traffic Path Between Known Endpoints

Let's use the switch CLI to verify the traffic path shown in [Figure 8-5](#). ACI forwarding logic dictates that in determining where to send traffic, the endpoint table is consulted before anything else. [Example 8-5](#) shows that LEAF102 needs to

send the traffic out logical interface tunnel2 toward PTEP address 10.233.60.224 for traffic to reach 10.233.59.50. The fact that a tunnel is shown as the egress interface in the **show endpoint** output is confirmation that LEAF102 considers the endpoint to be a remote endpoint.

### **Example 8-5 CLI Output Showing a Remote Endpoint**

[Click here to view code image](#)

```
LEAF102# show endpoint ip 10.233.59.50
Legend:
  s - arp          H - vtep          V - vpc-attached
  p - peer-aged    R - peer-attached-rl  B - bounce      S - static
  M - span         D - bounce-to-proxy   O - peer-attached  a - local-aged
  m - svc-mgr      L - local          E - shared-
  service
+-----+-----+-----+
-----+
      VLAN/          Encap        MAC Address     MAC
Info/   Interface
      Domain       VLAN        IP Address      IP
Info
+-----+-----+-----+
-----+
  66           vlan-3172      0050.56b7.a88b    0
tunnel2
Prod:Temp      vlan-3172      10.233.59.50    0
tunnel2

LEAF102# show interface tunnel 2
Tunnel2 is up
```

```

MTU 9000 bytes, BW 0 Kbit
Transport protocol is in VRF "overlay-1"
Tunnel protocol/transport is ivxlan
Tunnel source 10.233.60.226/32 (lo0)
Tunnel destination 10.233.60.224
Last clearing of "show interface" counters never
Tx
0 packets output, 1 minute output rate 0 packets/sec
Rx
0 packets input, 1 minute input rate 0 packets/sec

```

What leaf corresponds to 10.233.60.224? As indicated in [Example 8-6](#), 10.233.60.224 is the PTEP for LEAF101. Note that the command **acidiag fnvread** can be run on either APICs or fabric nodes.

### **Example 8-6 Identifying the Node Corresponding to a Tunnel Destination**

[Click here to view code image](#)

APIC1# acidiag fnvread				
ID	Pod ID	Name	Serial Number	IP Address
Role	State	LastUpdMsgId		
101	1	LEAF101	FDOXXXXXA	10.233.60.224/32
leaf	active	0		
102	1	LEAF102	FDOXXXXXB	10.233.60.226/32
leaf	active	0		
103	1	LEAF103	FDOXXXXXC	10.233.60.228/32
leaf	active	0		
104	1	LEAF104	FDOXXXXXD	10.233.60.229/32
leaf	active	0		
201	1	SPINE201	FDOXXXXXE	10.233.60.225/32

spine	active	0		
202	1	SPINE202	FDOXXXXF	10.233.60.227/32
spine	active	0		
Total 4 nodes				

When LEAF101 receives the encapsulated packet, it sees that the outer destination IP address is its PTEP. It therefore decapsulates the packet, sees that it is destined to one of its local endpoints, and forwards it on to 10.233.59.50.

At this point, if LEAF101 does not have entries for remote endpoint EP1, it adds the necessary remote endpoint entry to its endpoint table. In this case, it only needs to learn the IP address because the two communicating endpoints are in different subnets. It populates its local tunnel interface corresponding with PTEP 10.233.60.226 as the egress interface toward this remote endpoint.

Before we move on to the next topic, there is one more point to reflect on. The output in [Example 8-5](#) shows that LEAF102 learned both the MAC address and IP address 10.233.59.50. While a leaf may learn both the MAC address and any IP addresses for a remote endpoint if the switch is performing both switching and routing operations to a destination endpoint, this is not the case in this example. In fact, EP2 is the only endpoint in this particular BD and EPG at the moment; therefore, no Layer 2 operation with endpoints behind LEAF102 could have possibly occurred. Clues to why LEAF102 learned both the MAC address and IP address in this instance can be found by reviewing the type of endpoint registered. The endpoint table logs 10.233.59.50 with the letter O, implying that LEAF102 and LEAF101 have a vPC peering with one another. Even though EP2 is not itself behind a virtual port channel, leaf switches in a vPC domain do synchronize entries. This

synchronization includes both MAC addresses and IP addresses.

Behind the scenes, let's break the vPC peering by deleting the explicit vPC protection group. [Example 8-7](#) shows that, as expected, LEAF102 now only learns the IP address of EP2 since all of its local endpoints only have inter-subnet communication with this particular remote endpoint.

**Example 8-7 No Intra-Subnet Communication Means the Remote Endpoint MAC Address Is Not Learned**

[Click here to view code image](#)

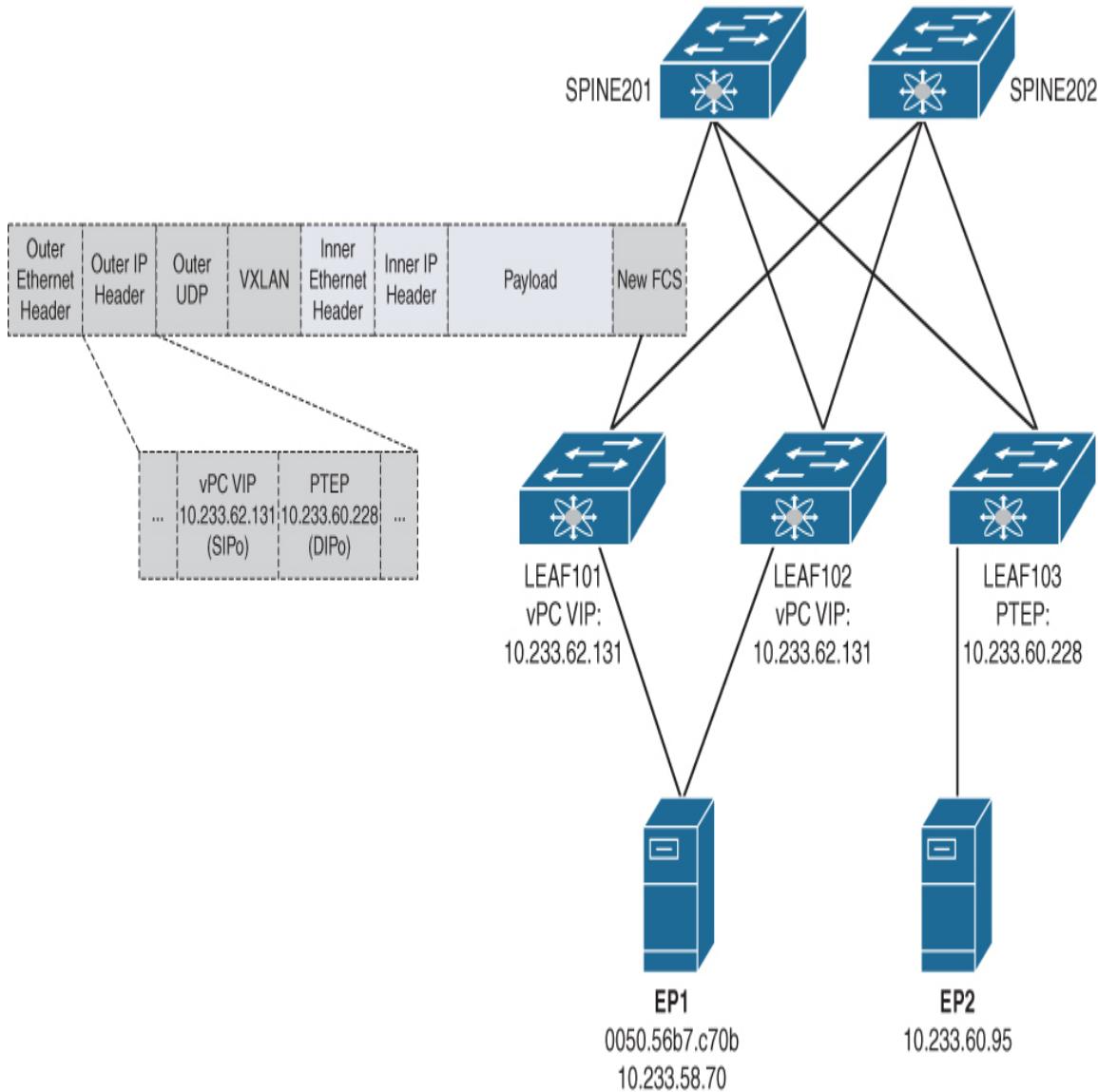
```
LEAF102# show system internal epm endpoint ip 10.233.59.50

MAC : 0000.0000.0000 :: Num IPs : 1
IP# 0 : 10.233.59.50 :: IP# 0 flags :   :: l3-sw-hit: No
Vlan id : 0 :: Vlan vnid : 0 :: VRF name : Prod:Temp
BD vnid : 0 :: VRF vnid : 2228225
Phy If : 0 :: Tunnel If : 0x18010002
Interface : Tunnel2
Flags : 0x80004400 :: sclass : 49161 :: Ref count : 3
EP Create Timestamp : 06/02/2020 06:07:58.418861
EP Update Timestamp : 06/02/2020 06:07:58.418861
EP Flags : IP|sclass|timer|
```

## Understanding Learning and Forwarding for vPCs

Sometimes a tunnel may correspond to a logical switch representing two vPC peers. This is the case when an endpoint is behind a vPC. [Figure 8-6](#) shows EP1 vPC attached to LEAF101 and LEAF102. If EP1 were to send traffic to known remote endpoint EP2, the recipient leaf

(either LEAF101 or LEAF102) would encapsulate the traffic in VXLAN to send it to LEAF103. But the PTEP addresses would not be used to source this traffic. Instead, these vPC peers would place a special TEP address called a *vPC VIP* or (*virtual IP*) in the Outer Source IP Address field.



**Figure 8-6** Communication Sourced from an Endpoint Behind a vPC

This behavior ensures that LEAF103 learns the endpoint from the vPC VIP and not LEAF101 and LEAF102 PTEP

addresses. If ACI did not employ this approach, remote switches would see endpoint entries for EP1 bouncing between LEAF101 and LEAF102. Use of vPC VIPs, therefore, ensures that endpoint learning always remains stable and that Layer 3 equal-cost multipathing can be used for forwarding to and from vPC-attached endpoints.

[Example 8-8](#) shows how you can identify the vPC VIP assigned to a leaf switch that is part of a vPC domain. This command shows only the vPC VIP for the local vPC domain.

### **Example 8-8 Identifying the VTEP Assigned to a vPC Domain**

[Click here to view code image](#)

```
LEAF101# show system internal epm vpc
(...output truncated for brevity...)
Local TEP IP : 10.233.60.224
Peer TEP IP : 10.233.60.226
vPC configured : Yes
vPC VIP : 10.233.62.131
MCT link status : Up
Local vPC version bitmap : 0x7
Peer vPC version bitmap : 0x7
Negotiated vPC version : 3
Peer advertisement received : Yes
Tunnel to vPC peer : Up
```

An alternative way to identify all the vPC VIPs in a fabric is to log in to the APICs instead and run the command **show vpc map**. Similar output can be obtained from the GUI in the Virtual IP column displayed under **Fabric > Access Policies > Policies > Switch > Virtual Port Channel Default**.

In light of how ACI learns remote endpoints behind vPCs, it is important for switches to be paired into vPC domains before deploying vPC interfaces.

## Forwarding Scenario 3: Spine Proxy to Unknown Destination

Imagine that an ACI leaf has performed a lookup in its endpoint table and finds that it has not learned the destination endpoint for a particular traffic flow, as is the case in [Example 8-9](#). What does the leaf do next?

### Example 8-9 Endpoint Not Found in Leaf Endpoint Table

[Click here to view code image](#)

LEAF102# show endpoint ip 10.233.59.100			
Legend:			
s - arp attached	H - vtep peer-aged	V - vpc-	
R - peer-attached-rl	B - bounce	S - static	
M - span			
D - bounce-to-proxy	O - peer-attached	a - local-aged	
m - svc-mgr			
L - local	E - shared-service		
+-----+-----+-----+			
VLAN/ MAC Info/	Encap Interface	MAC Address	
Domain Info	VLAN	IP Address	IP
+-----+-----+-----+			

Once destination endpoint 10.233.59.100 has been confirmed to *not* be in the endpoint table, LEAF102 consults its routing table. [Example 8-10](#) shows what the specific route lookup on LEAF102 might yield.

### **Example 8-10** Pervasive Route for a Bridge Domain Subnet

[Click here to view code image](#)

```
LEAF102# show ip route 10.233.59.100 vrf Prod:Temp
IP Route Table for VRF "Prod:Temp"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

10.233.59.0/24, ubest/mbest: 1/0, attached, direct, pervasive
    *via 10.233.62.130%overlay-1, [1/0], 06:50:59, static,
        tag 4294967294
            recursive next hop: 10.233.62.130/32%overlay-1
```

As indicated by the keyword **pervasive** in [Example 8-10](#), the lookup yields a special route called a *pervasive route*. Pervasive routes and pervasive gateways are two different things with different purposes. To better understand the function of a pervasive route, it helps to identify its next hop. For this purpose, examine [Example 8-11](#).

### **Example 8-11** Anycast TEP Address Used for Forwarding to the Spine Proxy

[Click here to view code image](#)

```
LEAF102# show isis dteps vrf overlay-1
IS-IS Dynamic Tunnel End Point (DTEP) database:
DTEP-Address      Role      Encapsulation     Type
10.233.60.224    LEAF      N/A                  PHYSICAL
```

10.233.60.225	SPINE	N/A	PHYSICAL
10.233.62.130	SPINE	N/A	PHYSICAL, PROXY-
ACAST-V4			
10.233.62.129	SPINE	N/A	PHYSICAL, PROXY-
ACAST-MAC			
10.233.62.128	SPINE	N/A	PHYSICAL, PROXY-
ACAST-V6			
10.233.60.227	SPINE	N/A	PHYSICAL
10.233.62.131	LEAF	N/A	PHYSICAL

It should be clear from the output in [Example 8-11](#) that the pervasive route references the anycast spine proxy address for IPv4 forwarding as the next hop. This is because traffic toward unknown destinations can potentially be sent to one of the spines to allow the spine COOP database to then determine how to forward the traffic onward to its destination.



To summarize, a **pervasive route** is a route to a BD subnet that points to the spine proxy TEP as its next-hop IP address. Because each leaf consults its endpoint table first, a pervasive route does not come into play unless an endpoint is deemed to be unknown. The function of a pervasive route, therefore, is to ensure that a leaf switch knows that a particular destination is expected to be inside the fabric. If a pervasive route were not deployed for a BD subnet, a leaf might think that a default route learned via an L3Out is the best way to get the traffic to its destination. If so, it could decide to either drop the traffic or forward it to the border leaf TEP addresses instead of the spine proxy address. Pervasive routes help prevent this suboptimal forwarding scenario.

## Note

A pervasive route is installed on a leaf if at least one EPG on the leaf is allowed through contracts or another whitelisting mechanism to communicate with at least one EPG associated with the destination bridge domain. This is in line with ACI deploying policies only where they are needed.

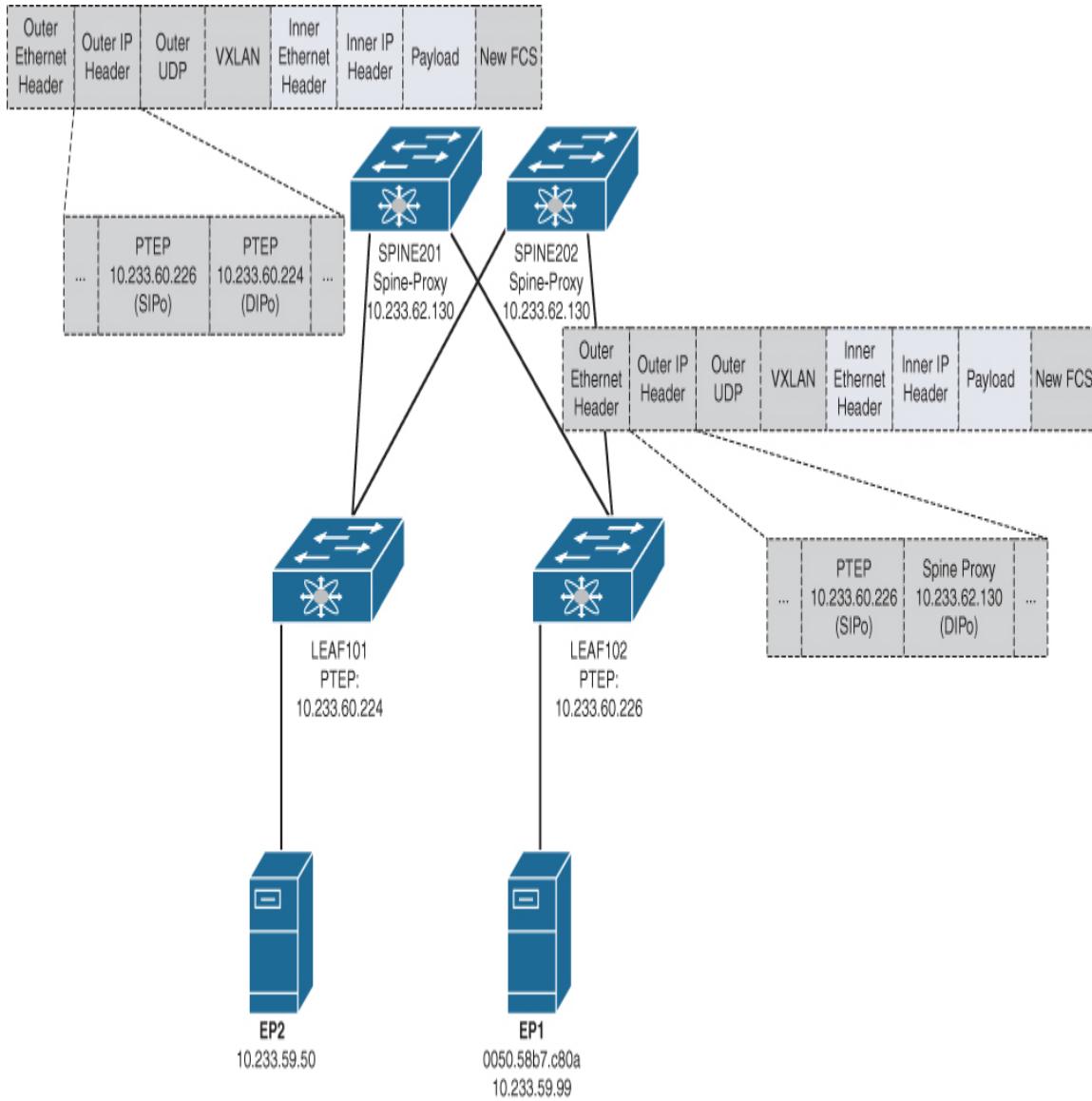


Just because a pervasive route exists and the spine proxy function can potentially be used for forwarding to an unknown destination does not mean that the leaf will choose to forward traffic to the spine proxy. Bridge domain settings dictate what happens next. If, for example, the traffic is L2 unknown unicast traffic and hardware proxy has been configured for the bridge domain, the leaf forwards the traffic destined to an unknown endpoint to the spine proxy function.

When a leaf decides to forward traffic to a spine proxy TEP, it does so by placing its own PTEP or vPC VIP as the outer source IP address in the VXLAN encapsulation. The intended spine proxy TEP gets populated as the outer destination IP address.

[Figure 8-7](#) recaps how spine proxy forwarding works. In this case, EP1 needs to get traffic to EP2. Both endpoints are in the same bridge domain, and the default gateway (10.233.59.1/24) is in the fabric. Let's suppose that EP1 has never communicated with EP2. If that were the case, EP1 would first send an ARP request out to EP2 because it is in the same subnet as EP2. LEAF102 would consult its

endpoint table and not find an entry. It would then do a routing table lookup and see a pervasive route. Regardless of the pervasive route, it knows the default gateway is in the fabric. This is because it, too, has deployed a pervasive gateway for the bridge domain. It then needs to decide whether to use flooding or hardware proxy. It decides to send the traffic to the spine proxy because the L2 Unknown Unicast setting on the bridge domain was set to Hardware Proxy, and ARP flooding has been disabled. Next, it needs to decide which spine proxy TEP to place in the Outer Destination IP Header field. It selects the spine proxy PROXY-ACAST-V4 IP address. Because EP1 is single-homed, LEAF102 places its PTEP in the Outer Source IP Address field. In this example, LEAF101 and LEAF102 are not vPC peers anyway.



**Figure 8-7** LEAF102 Using Spine Proxy to Send EP1 Traffic to an Unknown Destination



Once a spine receives a frame with the spine proxy TEP as the outer destination IP address, it knows it needs to perform a COOP lookup on the destination. If it *does* know the destination, it updates the outer destination IP address

to that of the recipient leaf PTEP or vPC VIP and sends the traffic onward. This DIPo update at the spine is demonstrated in [Figure 8-7](#). If the spine *does not* know the destination and the destination is in a routed bridge domain, it drops the traffic and then kicks off the ARP gleaning process, with the assumption that the destination may be a silent host. If the destination *is* a silent host, it is likely to be learned in time for future spine proxy operations to succeed.

After the spine updates the outer destination IP address and the traffic eventually reaches its destination, the response from the destination endpoint leads to the source leaf learning the remote destination. Future traffic is then sent directly between the leaf switches without spine proxy involvement.

Note that in this entire process, there is no need to flood traffic across the fabric. By minimizing flooding, the hardware proxy forwarding behavior reduces the impact of endpoint learning on switch resources and safeguards the stability of the network.

## Note

The discussion around [Figure 8-7](#) makes reference to the PROXY-ACAST-V4 address. Recall from [Example 8-11](#) that ACI assigns three separate IP addresses to spine proxy functions. PROXY-ACAST-V4 concerns spine proxy forwarding for IPv4 traffic, which also includes ARP requests when ARP flooding has been disabled. The PROXY-ACAST-MAC address is used for L2 unknown unicast traffic when the L2 Unknown Unicast parameter is set to Hardware Proxy. Finally, PROXY-ACAST-V6 is used when spine proxy forwarding IPv6 traffic to an unknown destination.

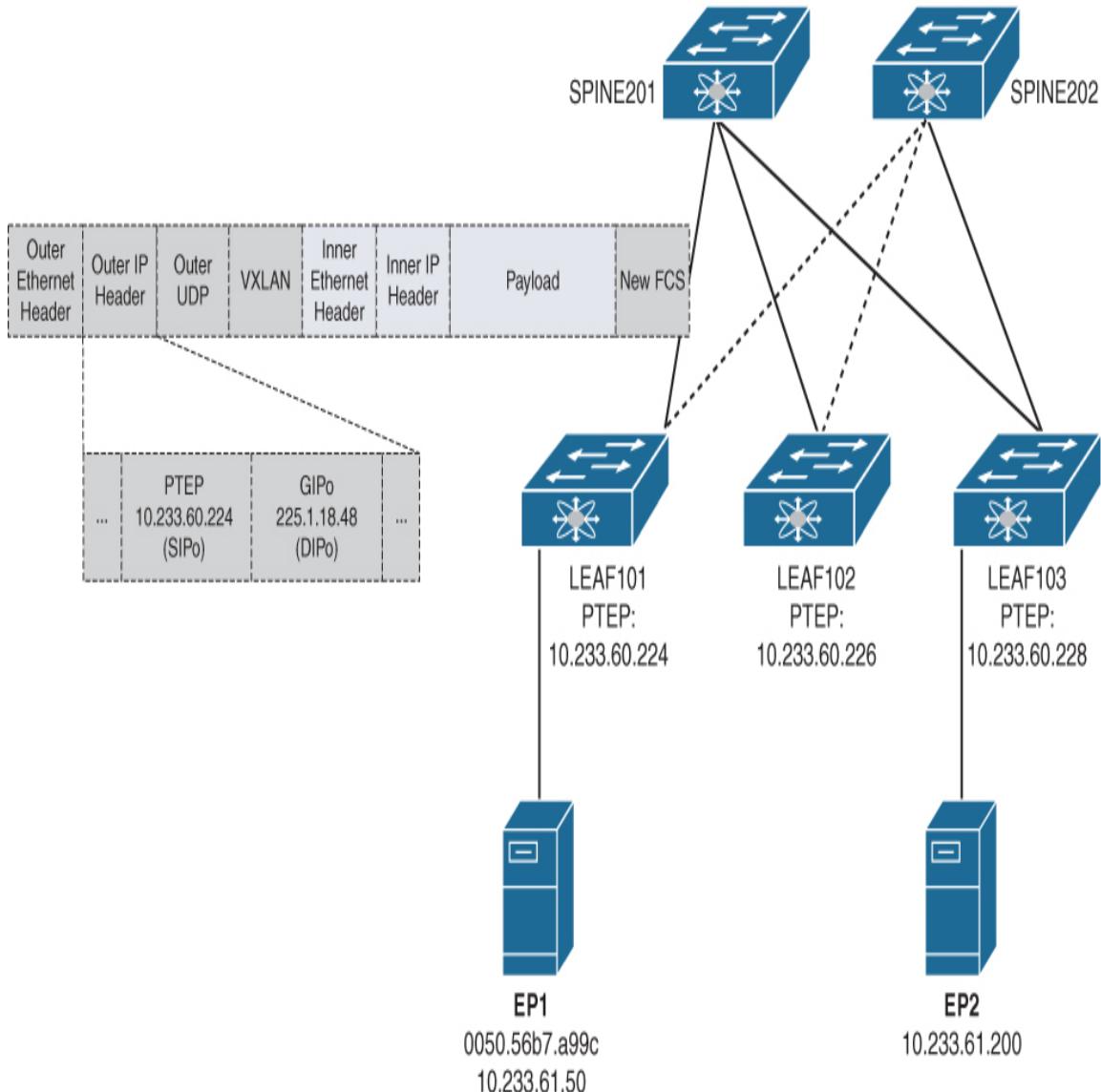
# Forwarding Scenario 4: Flooding to Unknown Destination

There are multiple forwarding scenarios for which ACI may need to flood traffic instead of using the spine proxy. For example, if there are silent hosts in an L2 bridge domain, ACI silent host detection would not work, and spine proxy forwarding might black-hole traffic. Also, when extending a VLAN into an ACI fabric and the default gateway remains outside ACI, flooding can help prevent certain traffic black holes. [Chapter 10, “Extending Layer 2 Outside ACI,”](#) discusses Layer 2 extension in detail and addresses the logic behind flooding during network migrations. For now, let’s dive into the details of how flooding and multi-destination forwarding work in ACI.

When a leaf switch receives traffic that needs to be flooded in a bridge domain, it takes the BD multicast address, selects one of several predefined loop-free topologies, and adds bits corresponding with the selected topology to the BD multicast address. The resulting IP address becomes the outer multicast Group IP outer (GIPo) address, which the source leaf places in the Outer Destination IP Address field when encapsulating the original payload in VXLAN.

The predefined topologies based on which ACI forwards multi-destination traffic are called ***forwarding tag (FTag) trees***. Each FTag tree does not necessarily use all fabric uplinks. That is why ACI creates multiple FTag trees and load balances multi-destination traffic across them. All switches in a fabric understand based on the FTag bits in the GIPo address how to forward the traffic they receive further along the specified FTag tree. Four bits are used to identify FTag IDs; ACI fabrics support up to 12 FTag trees.

For example, [Figure 8-8](#) depicts intra-subnet communication between EP1 and EP2 at a time when LEAF101 has not learned EP2. Say that LEAF101 has either not yet learned EP2 and L2 Unknown Unicast has been set to Flood, or the ARP Flooding setting has been enabled on the respective bridge domain. Either of these two settings would force ARP requests to be flooded. In this case, when LEAF101 determines through an endpoint table lookup that the endpoint is unknown, it needs to decide whether to send the ARP request to the spine proxy or to flood the ARP request. Because the ARP Flooding setting has been enabled, it floods the ARP request by adding the GIPo address as the destination outer IP address and adding its PTEP as the source outer IP address. Let's say that it selected FTag ID 0 when determining the GIPo address. That is why the GIPo address depicted appears to be a /28 subnet ID. In this case, the BD multicast address is the same as the GIPo address. This is not always the case.



**Figure 8-8 Flooding Traffic to an Unknown Destination over an FTag Topology**

When determining where to forward the ARP request, LEAF101 follows the FTag 0 tree and sends the packet to SPINE201. As root of the FTag 0 tree, this spine then forwards the packet to all other leaf switches. Finally, LEAF103 forwards the packet to SPINE202. This last forwarding operation to SPINE202 is not really required for single-pod fabrics because spines do not themselves house endpoints. However, ACI ensures that flooded traffic reaches

all leaf and spine switches anyway. This approach addresses specific use cases like multi-destination forwarding in multipod environments.

Note that even though both of these last two forwarding scenarios involved ARP requests, it is not just ARP traffic that may be flooded or spine proxied in ACI.

## Understanding ARP Flooding

By now, it should be clear that ACI has ways to learn *almost all endpoints*, regardless of the L2 Unknown Unicast setting chosen.



If the Hardware Proxy setting is enabled, the ARP Flooding setting is disabled, and COOP knows of the endpoint, the fabric unicasts ARP requests to the intended destination. If the spine COOP database does not know the destination endpoint, the spines drop the ARP traffic and trigger ARP gleaning.



If, on the other hand, the ARP Flooding setting is enabled, the leaf switch receiving ARP traffic floods the traffic based on an FTag tree. The source leaf then learns the destination endpoint in response packets if the destination endpoint actually exists.

A tangible difference between enabling and disabling ARP flooding occurs with silent host movements. Suppose that hardware proxy has been enabled on a bridge domain, ARP

flooding has been disabled, and ACI has already learned a silent host in the BD through ARP gleaning. If the silent host moves from one location to another without notifying the new ACI leaf via GARP or some other mechanism, ACI switches continue to forward traffic intended for the silent IP address to the previous location until retention timers clear the endpoint from COOP. Until that point, if an endpoint sends ARP requests toward this silent host, ARP gleaning is not triggered because COOP considers the destination endpoint to be known. On the other hand, with ARP flooding enabled on the BD, ARP requests are flooded, and the silent host responds at its new location, enabling the new local leaf to learn the silent host and update COOP.

Just because ARP flooding can help in endpoint learning in situations like these does *not* mean ARP flooding should be enabled on all bridge domains. But in environments in which silent hosts with low-latency communication requirements can move between leaf switches, enabling ARP flooding on the bridge domains housing such endpoints can minimize the potential for traffic disruption.



Note a caveat related to ARP flooding: If Unicast Routing has been disabled on a bridge domain, ARP traffic is always flooded, even if the ARP Flooding setting is not enabled.

## Deploying a Multi-Tier Application

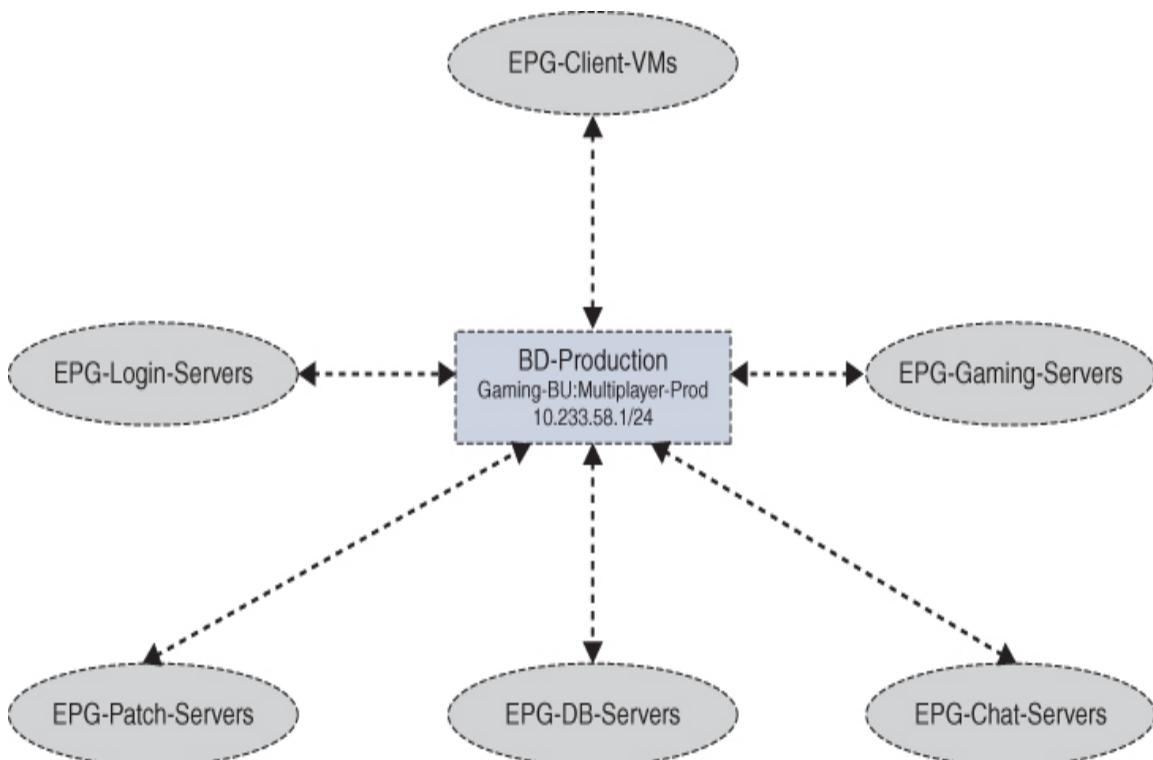
In this section, we put into practice the information provided so far in this chapter. Suppose the business unit involved in gaming products from [Chapter 7, “Implementing Access Policies,”](#) wants to deploy an application consisting of multiple component tiers. One of the first things engineers

typically consider is the number of bridge domains and EPGs needed.

For the sake of argument, say that the engineers involved in the project know that none of the servers that are part of the solution will be silent hosts. All default gateways will reside inside ACI since the desire is to implement granular whitelisting for the application and leverage pervasive gateways for east-west traffic optimization.

With the data already available, the ACI engineers know that flooding for this application can be kept at an absolute minimum. For this reason, they decide to deploy only a single bridge domain; each tier of this application will be a separate EPG.

[Figure 8-9](#) shows a hypothetical BD/EPG design for such an application that is simple yet effective. If the application ever scales beyond a single /24 subnet, the engineers know they can easily add additional subnets to the BD.

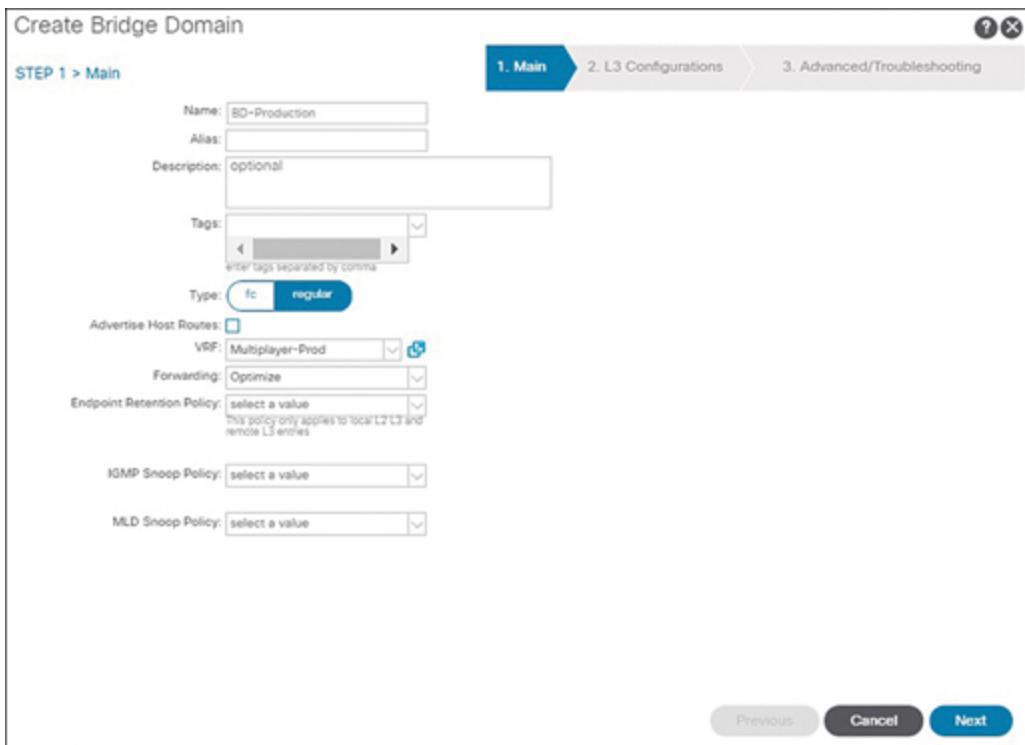


**Figure 8-9** BD and EPG Relationships for a Hypothetical Multi-Tier Application

## Configuring Application Profiles, BDs, and EPGs

Because bridge domains and EPGs for the gaming business unit application are known, it is time to implement the required objects.

To create the bridge domain in line with the diagram shown in [Figure 8-9](#), navigate to **Tenants > Gaming-BU > Networking**, right-click Bridge Domains, and select Create Bridge Domain. [Figure 8-10](#) shows optimal settings for the first page of the Create Bridge Domain wizard.



**Figure 8-10** Configuring a Bridge Domain Using Optimize Forwarding Parameters

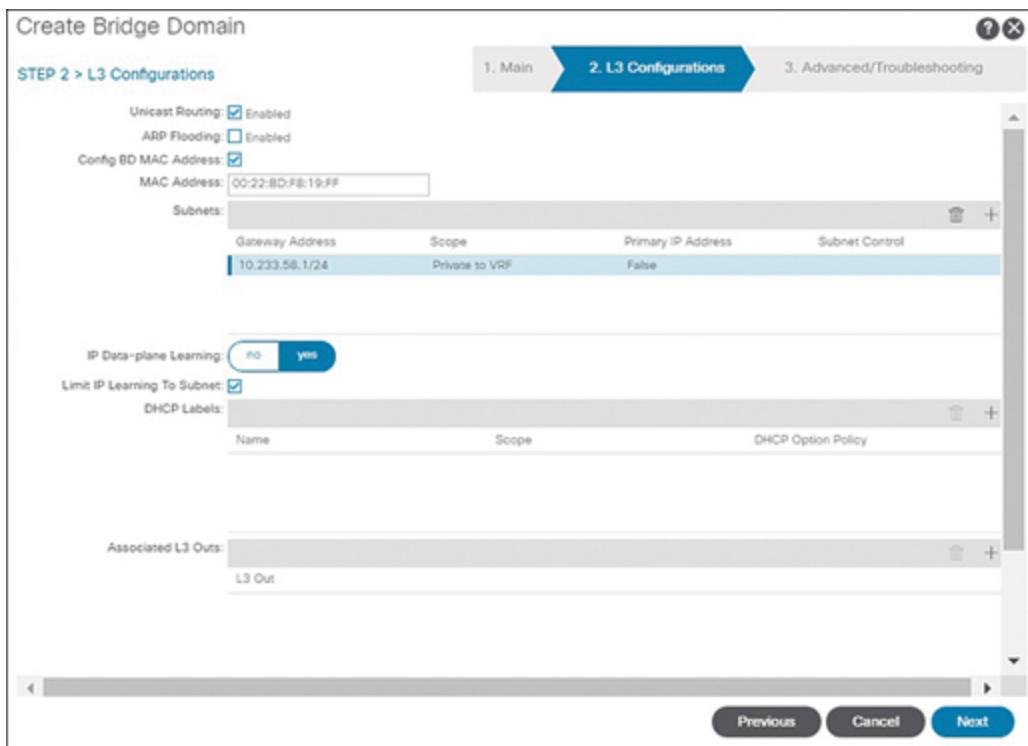
The default Optimize forwarding settings depicted in the Create Bridge Domain wizard are as follows:



- **L2 Unknown Unicast:** Hardware Proxy
- **L3 Unknown Multicast Flooding:** Flood
- **Multi-destination Flooding:** Flood in BD

If any of these settings need to be modified, you can select the Custom option from the Forwarding drop-down box to view these three settings.

Click Next to move on to the second page of the Create Bridge Domain wizard, which allows you to adjust Layer 3 settings. Leave Unicast Routing enabled because ACI will be performing routing for the BD and will be expected to learn endpoint IP addresses in the BD. ARP Flooding is enabled by default in some ACI code versions. However, it can be disabled for this BD because the subnet default gateway will reside in the fabric, and none of the endpoints are anticipated to be silent hosts. Finally, under the Subnets view, click the + sign and add the specified BD subnet. When defining a BD subnet, always enter the default gateway IP address followed by the subnet CIDR notation and *not* the subnet ID. The Private to VRF Scope setting shown in [Figure 8-11](#) suggests that subnet 10.233.58.0/24 will not be advertised out the fabric for the time being. Click Next on the L3 Configurations page and then click Finish on the final page of the Create Bridge Domain wizard to create the BD.



**Figure 8-11** Configuring Layer 3 Settings for a Bridge Domain

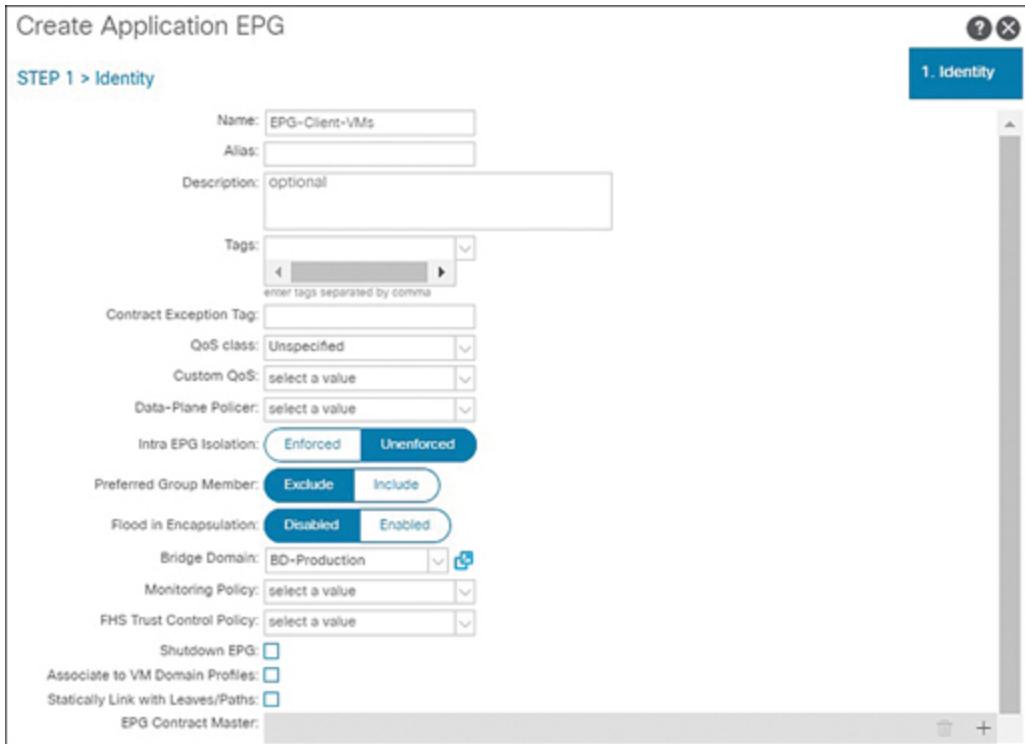
Note that once the BD is created, the APIC assigns a BD multicast address to the bridge domain, as shown in [Figure 8-12](#). This is the address used as the basis for the GIPo address for traffic flooding.

Name	Type	Segment	VRF	Multicast Address	Custom MAC Address	L2 Unknown Unicast	ARP Flooding	Unicast Routing	Subnet
BD-Production	regular	16285613	Multiplayer-Prod	225.0.34.208	00:22:BD:F8:19:FF	Hardware Proxy	False	True	10.233.58.1/24

**Figure 8-12** *Multicast Address Assigned to a Bridge Domain*

With the BD created, it is now time to create the required EPGs. Before creating EPGs, an application profile needs to be created to house the EPGs. Anticipating that further instances of the application may be needed in the future, the ACI engineers decide to name this application container Multiplayer-App1. To create this EPG, you navigate to **Tenants > Gaming-BU > Application Profiles > Multiplayer-App1**, right-click Application EPGs, and select Create Application EPG.

Figure 8-13 shows that the most important settings for an EPG at this point are the name and bridge domain association. All other settings shown here have been left at their defaults. Note that if traffic from endpoints in this EPG should be assigned to a specific QoS level, this can be accomplished via the QoS Class drop-down box. Click Finish to create the EPG.



**Figure 8-13** The Create Application EPG Wizard

By navigating to the Application EPGs subfolder of an application profile, you can view the list of all EPGs in the application profile. [Figure 8-14](#) lists all the EPGs under the Multiplayer-App1 application profile. Note that the APICs have assigned a Class ID to each EPG. This parameter is used in the application of contracts between EPGs.

The screenshot shows the Cisco Application Policy Infrastructure Controller (APIC) web interface. At the top, there's a navigation bar with tabs for System, Tenants (which is selected), Fabric, Virtual Networking, L4-L7 Services, Admin, Operations, Apps, and Integrations. Below the navigation bar is a search bar labeled "Tenant Search: name or descr" with dropdown filters for common, Gaming-BU, infra, Production, and mgmt. The main content area is titled "Application EPGs" and displays a table of EPGs under the "Gaming-BU" tenant. The table columns are Name, Class ID, Preferred Group Member, Flood in Encapsulation, Bridge Domain, QoS class, Intra EPG Isolation, and In Shutdown. The table rows list various EPGs like EPG-Client-VMs, EPG-Login-Servers, etc., with their respective details.

Name	Class ID	Preferred Group Member	Flood in Encapsulation	Bridge Domain	QoS class	Intra EPG Isolation	In Shutdown
EPG-Client-VMs	32780	Exclude	Disabled	BD-Production	Unspecified	Unenforced	No
EPG-Login-Servers	16389	Exclude	Disabled	BD-Production	Unspecified	Unenforced	No
EPG-Gaming-Servers	49162	Exclude	Disabled	BD-Production	Unspecified	Unenforced	No
EPG-Patch-Servers	49163	Exclude	Disabled	BD-Production	Unspecified	Unenforced	No
EPG-DB-Servers	16390	Exclude	Disabled	BD-Production	Unspecified	Unenforced	No
EPG-Chat-Servers	49164	Exclude	Disabled	BD-Production	Unspecified	Unenforced	No

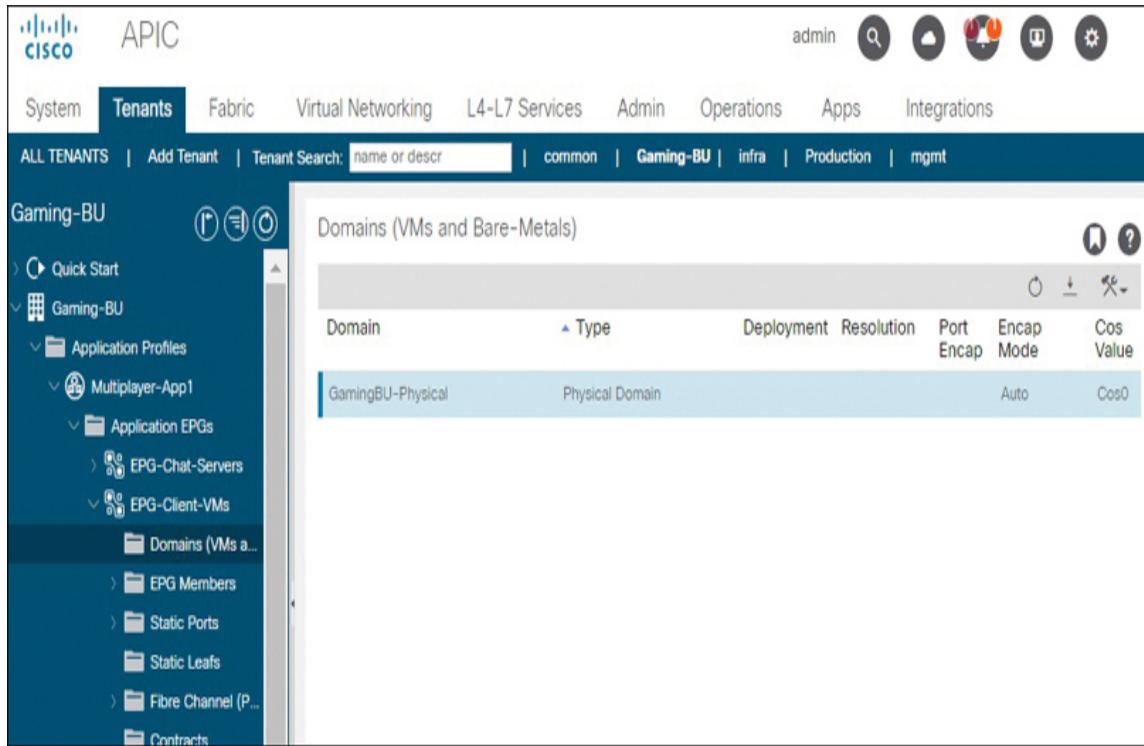
**Figure 8-14** Verifying EPGs Under a Specific Application Profile

In [Figure 8-14](#), notice the In Shutdown column. Shutting down an EPG is an easy way to isolate a set of endpoints without having to make any changes to policy if at any point they are compromised.

## Assigning Domains to EPGs

Before you can map an EPG to physical switch ports, you need to assign a domain to the EPG. [Figure 8-15](#) shows a physical domain named GamingBU-Physical associated with EPG-Client-VMs.





**Figure 8-15** Assigning a Domain to an EPG

Recall that assigning a domain to an EPG is a tenant-side confirmation that an EPG has endpoints of a particular type. It also links the Tenants view with the Access Policies view by enabling certain ACI users to deploy an EPG to certain switch ports using encapsulations specified in the VLAN pool referenced by the added domain.

## Policy Deployment Following BD and EPG Setup

It would be natural but incorrect to assume that the configurations performed so far in this section would lead to the deployment of pervasive gateways, pervasive routes, or even PI VLANs on any leaf switches in the fabric. This goes back to the concept discussed earlier of ACI not deploying policy that is not needed on the switches. In this case, no

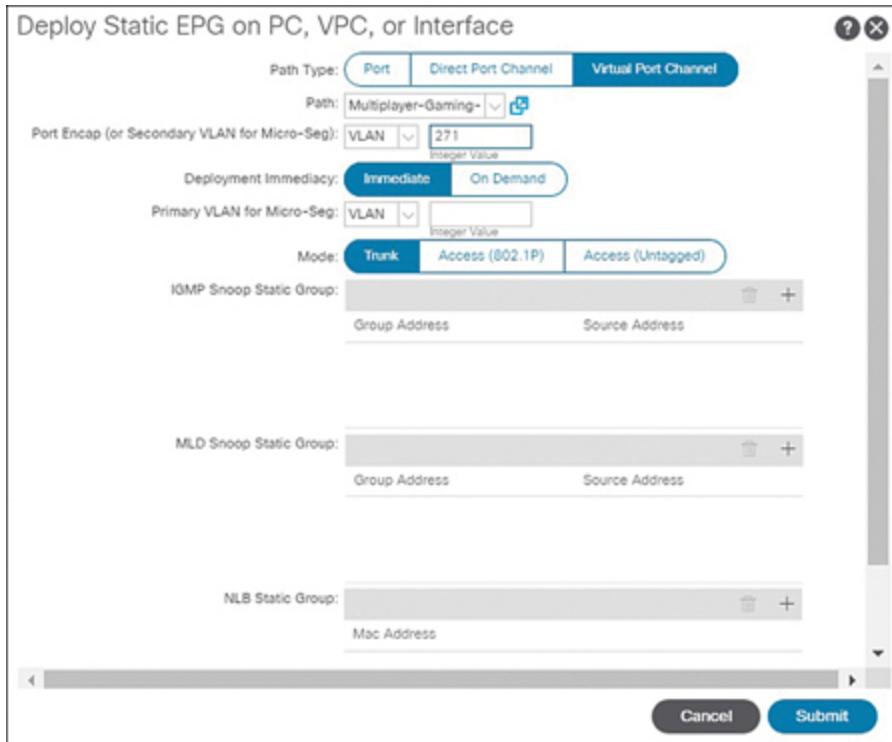
policy is needed on switches because no form of EPG-to-port mapping has yet been implemented.

## Mapping EPGs to Ports Using Static Bindings

[Chapter 7](#) covers the creation of a number of access policies. One of the access policies created there is a vPC named Multiplayer-Server-3, which is assigned to Nodes 101 and 102 on port 1/38. This section shows how to assign one of the newly created EPGs to this vPC and how to get traffic flowing to the Multiplayer-Server-3 server. To do so, navigate to the EPG in question and expose its subfolders, right-click the Static Ports folder, and select Deploy Static EPG on PC, vPC, or Interface.

[Figure 8-16](#) shows the subsequent mapping of the EPG to the access policy Multiplayer-Server-3 using the VLAN Port Encap setting 271. Static binding path types include ports, port channels, and vPCs. Note that two of the critical settings shown require further explanation: Mode and Deployment Immediacy.





**Figure 8-16** *Statically Mapping an EPG to an Encapsulation on a vPC*

The static binding Mode is concerned with how ACI deploys an EPG on a given port. Three port binding modes are available in ACI:

**Key Topic**

- **Trunk:** Traffic for the EPG is sourced by the leaf switch with the specified VLAN tag. The leaf switch also expects to receive traffic tagged with that VLAN to be able to associate it with the EPG. Traffic received untagged is discarded.
- **Access (Untagged):** Traffic for the EPG is sourced by the leaf as untagged. Traffic received by the leaf switch as untagged or with the tag specified during the static binding configuration is associated with the EPG.

- **Access (802.1P):** When only a single EPG is bound to an interface using this setting, the behavior is identical as that of the untagged case. If additional EPGs are associated with the same interface, traffic for the EPG is sourced with an IEEE 802.1Q tag using VLAN 0 (IEEE 802.1P tag) or is sourced as untagged in the case of EX switches.

In general, the Deployment Immediacy setting governs when policy CAM resources are allocated for EPG-to-EPG communications. (Think whitelisting policies.) The Deployment Immediacy parameter can be set to either of the following:



- **On Demand:** Specifies that the policy should be programmed in hardware only when the first packet is received through the data path. This setting helps optimize the hardware space.
- **Immediate:** Specifies that the policy should be programmed in hardware as soon as the policy is downloaded in the leaf software.

For static bindings, a leaf policy download occurs at the time the binding is correctly configured. [Chapter 13, “Implementing Management,”](#) addresses policy download in further detail.

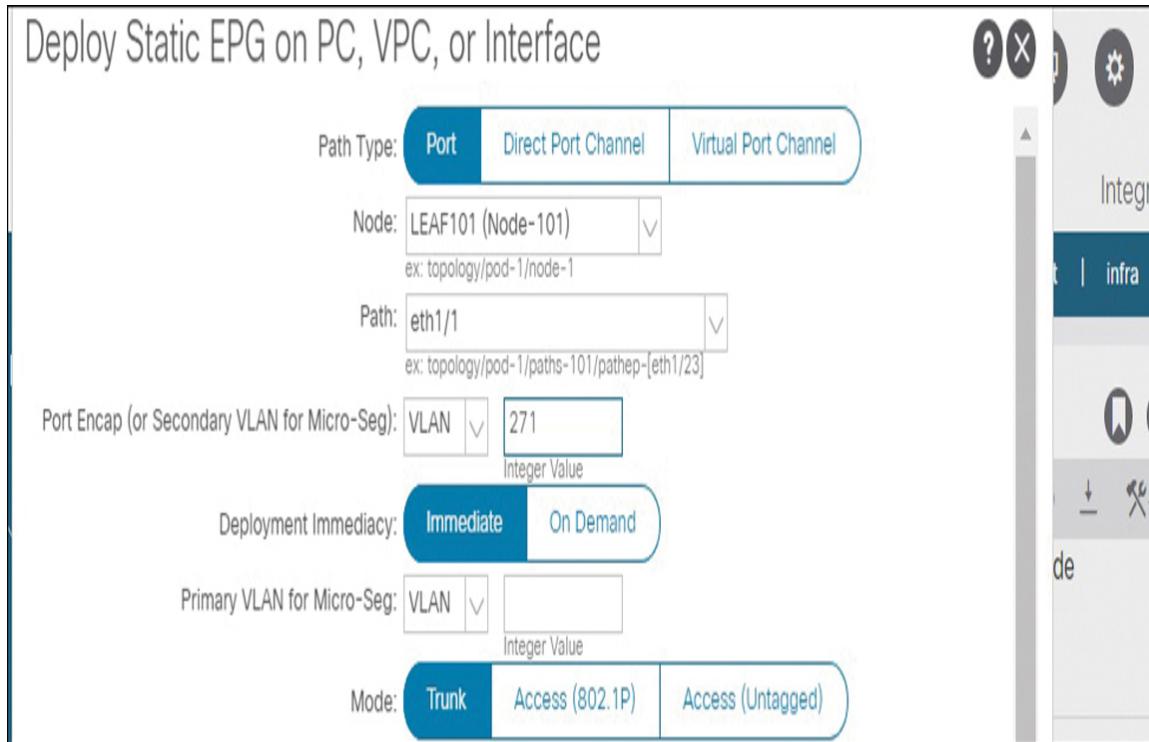
### Note

In small and medium-sized fabrics that have limited any-to-any contracts and where there is little fear of policy CAM exhaustion, implementing static path

bindings with Deployment Immediacy set to Immediate is most common. [Chapters 9, “L3Outs,” and 10, “Extending Layer 2 Outside ACI,”](#) address any-to-any contracts and policy CAM utilization in more detail.

There is a key configuration difference between mapping an EPG to port channels and vPCs and mapping it to individual ports. When mapping an EPG to a link aggregation, you select the desired vPC or port channel interface policy group in the Path drop-down box and are not necessarily concerned with physical port assignments. This is clear in [Figure 8-16](#). When mapping an EPG to an individual port, however, the desired leaf and physical port associated with the mapping need to be selected using the Node and Path drop-downs, respectively. Compare [Figure 8-17](#) with [Figure 8-16](#) for context.

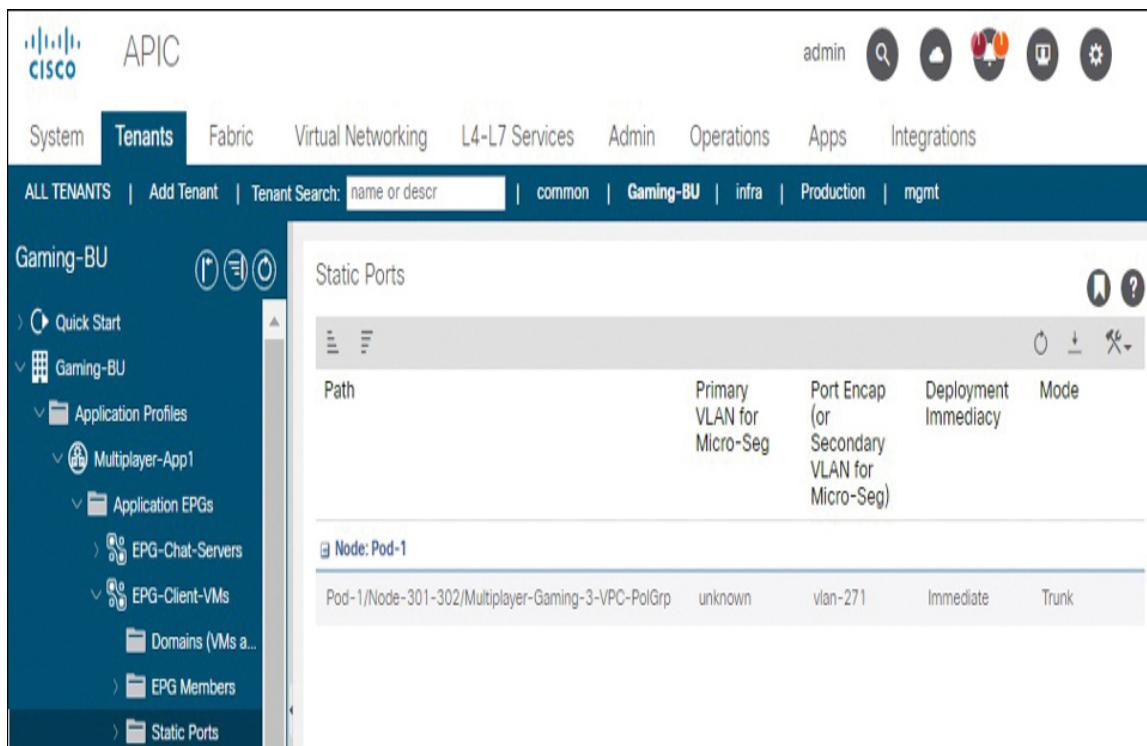




**Figure 8-17** *Statically Mapping an EPG to an Encapsulation on an Individual Port*

## Verifying EPG-to-Port Assignments

Static path bindings for an EPG can be verified under the Static Ports subfolder of the EPG. Although [Figure 8-18](#) shows that static path bindings have been configured, it does not verify whether the configuration has taken effect. It helps to check faults to rule out the possibility of underlying configuration problems preventing the deployment of the policy.



**Figure 8-18** Verifying a Static Path Binding for an EPG

### Key Topic

A more valid way to verify an EPG-to-port mapping is to check the status of its deployment on the switch itself. [Example 8-12](#) shows that EPG-Client-VMs has indeed been mapped to port Eth1/38 using the desired encapsulation. Note that Eth1/38 is a member of Po3, and it shows up in the output because the static binding being verified is for a vPC.

### **Example 8-12** CLI Output Indicating EPG-to-Port Mapping Deployed to a Leaf Switch

[Click here to view code image](#)

```
LEAF101# show vlan extended
(...output truncated for brevity...)
```

VLAN Name		Encap
Ports		
-----	-----	-----
-----	-----	-----
69	Gaming-BU:BD-Production	vxlan-
16285613	Eth1/38, Po3	
70	Gaming-BU:Multiplayer-App1:EPG-Client-VMs	vlan-271
	Eth1/38, Po3	

## Policy Deployment Following EPG-to-Port Assignment

When a leaf deploys EPG-to-port bindings, it also enables any other associated policies, such as BD subnet pervasive gateways and pervasive routes. [Example 8-13](#) verifies the deployment of additional policies on LEAF101.

### **Example 8-13** Verifying Deployment of a Pervasive Gateway and a Pervasive Route

[Click here to view code image](#)

```
LEAF101# show ip int brief vrf Gaming-BU:Multiplayer-Prod
IP Interface Status for VRF "Gaming-BU:Multiplayer-Prod" (23)
Interface          Address          Interface Status
vlan69            10.233.58.1/24    protocol-up/link-
                           up/admin-up

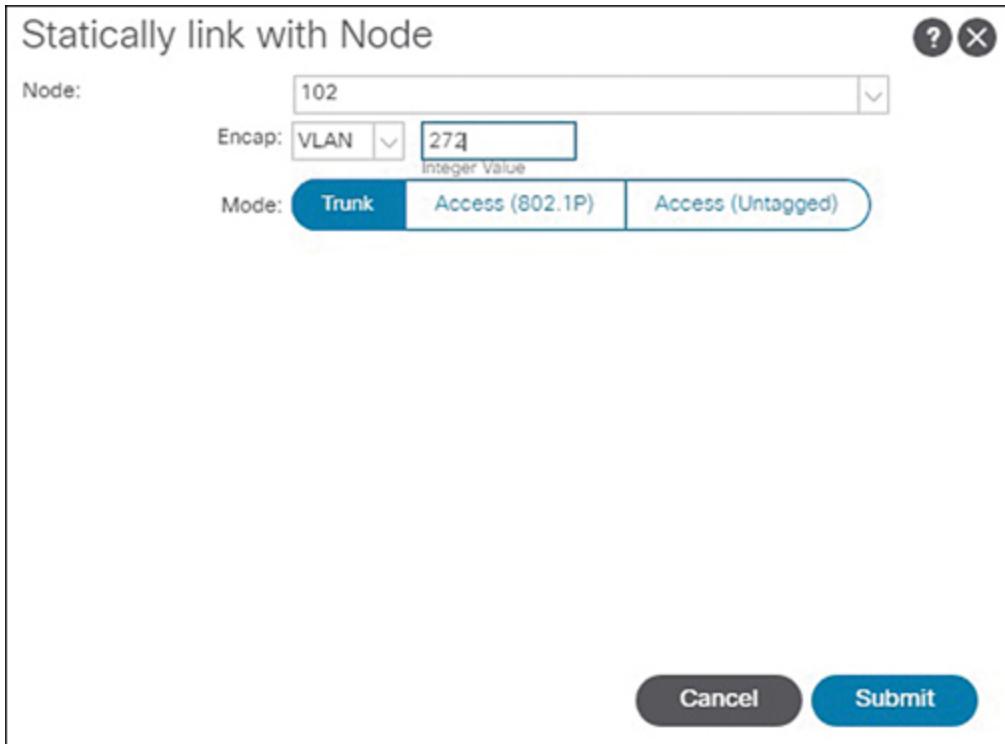
LEAF101# show ip route vrf Gaming-BU:Multiplayer-Prod
IP Route Table for VRF "Gaming-BU:Multiplayer-Prod"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>
```

```
10.233.58.0/24, ubest/mbest: 1/0, attached, direct, pervasive
  *via 10.233.62.130%overlay-1, [1/0], 09:16:13, static,
tag 4294967294
10.233.58.1/32, ubest/mbest: 1/0, attached, pervasive
  *via 10.233.58.1, vlan69, [0/0], 09:16:13, local, local
```

## Mapping an EPG to All Ports on a Leaf

Sometimes, a company procures a leaf for a particular function and requires that a handful of EPGs be deployed to all non-fabric ports on the leaf. This might be the case when attaching CIMC connections to a dedicated low-bandwidth copper leaf, for instance. In cases like these, use of the Static Leafs feature can reduce the number of clicks necessary to deploy EPG-to-port mappings.

To map an EPG to all non-fabric ports on a leaf, double-click the desired EPG to expose its subfolders. Then right-click Static Leafs and select Statically Link with Node. [Figure 8-19](#) shows the binding of an EPG to all non-fabric ports on a leaf with node ID 102 using the Port Encap setting 272.



**Figure 8-19** Mapping an EPG to All Non-Fabric Ports on a Leaf Switch

## Enabling DHCP Relay for a Bridge Domain

[Chapter 7](#) addresses the creation of DHCP relay policies under the Access Policies menu. These policies can then be consumed by bridge domains in any tenant to allow the BD to relay DHCP traffic to servers in different subnets.

To launch the Create DHCP Relay Label wizard, double-click the bridge domain in question to expose its subfolders, right-click DHCP Relay Labels, and select Create DHCP Relay Labels.

[Figure 8-20](#) illustrates the deployment of DHCP relay functionality for BD-Production using a DHCP relay policy created in [Chapter 7](#). The Scope setting shown in this figure

refers to whether the DHCP relay policy that will be consumed resides within a tenant (tenant) or under the Access Policies menu (infra). You can select a DHCP relay policy under the Name drop-down or create a new one and then click Submit.

The screenshot shows a configuration dialog titled "Create DHCP Relay Label". At the top right are a help icon (?) and a close button (X). Below the title, there is a "Scope" section with two tabs: "infra" (which is selected) and "tenant". Underneath is a "Name" field containing "Corporate-DHCP-Servers" with a dropdown arrow and a refresh/copy icon. A "DHCP Option Policy" field below it contains "select a value" with a dropdown arrow. At the bottom are "Cancel" and "Submit" buttons.

**Figure 8-20** Enabling DHCP Relay Functionality for a Bridge Domain

You can configure a DHCP option policy to provide DHCP clients with configuration parameters such as the domain, name servers, subnet, and IP addresses. Once the policy is deployed, you can verify the DHCP relay configuration on leaf switches via the **show ip dhcp relay** command.

## Whitelisting Intra-VRF Communications via Contracts

ACI being able to learn endpoints is not the same as endpoints being able to communicate with one another. Administrators still need to define valid traffic flows and whitelist them.

[Chapter 5](#), “Tenant Building Blocks,” provides a primer on contract theory. The remainder of this chapter puts that theory into practice by covering the implementation of contracts for a hypothetical multitier application.

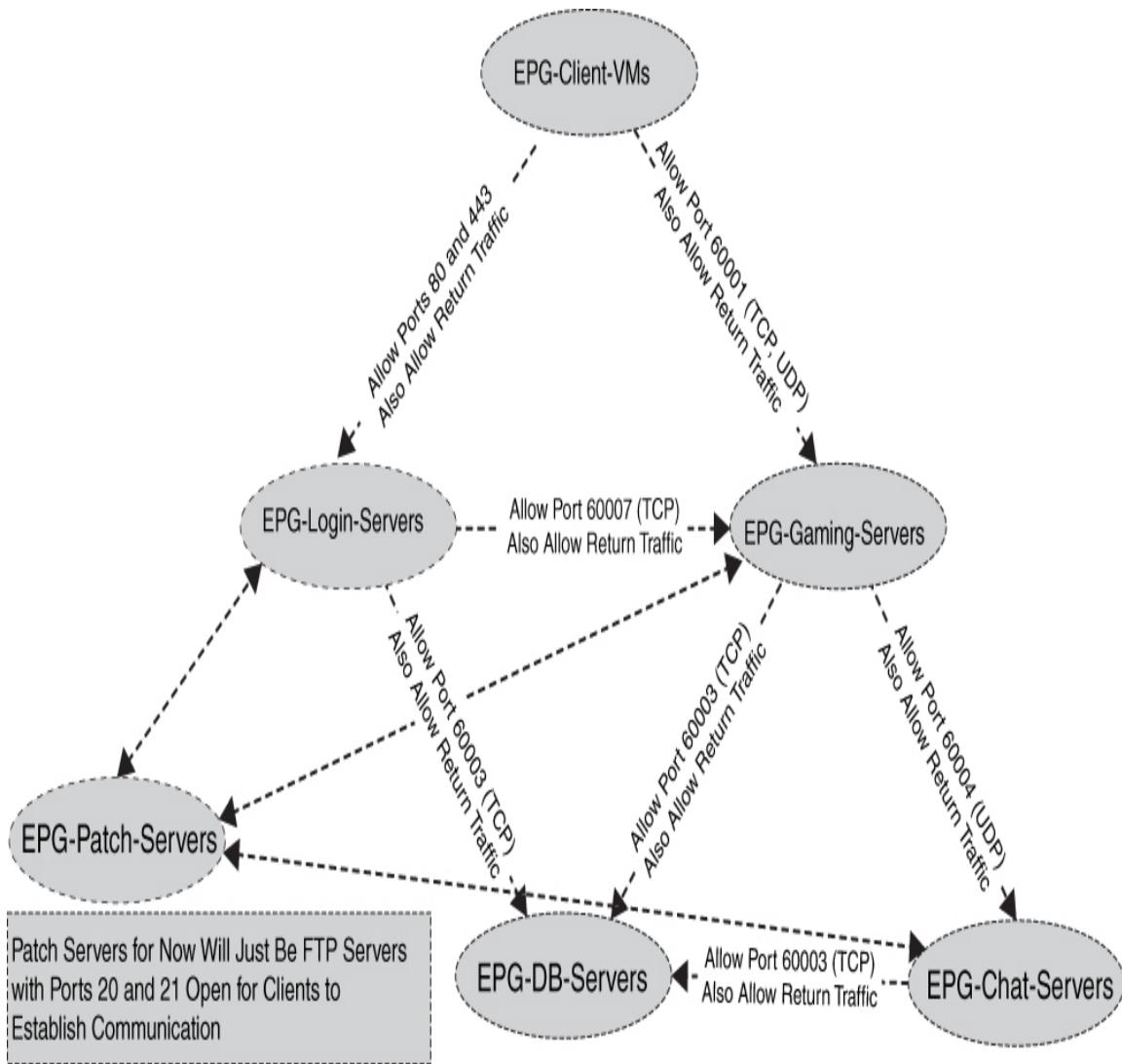
### Note

While Cisco calls out contracts and filters as DCACI 300-620 exam topics, it does not cover route leaking (which is instead covered on the DCACIA 300-630 exam). This implies that DCACI candidates should not be tested on whitelisting inter-VRF communications. Therefore, this section limits its focus to enabling intra-VRF communication among EPGs.

## Planning Contract Enforcement

Before enforcing contracts between endpoints, it is important to compile some form of data outlining valid and approved traffic flows. Solutions such as Cisco Tetration can help visualize all traffic flows in a data center. After you list the traffic flows that should be allowed, contracts can be put in place.

On the other hand, deploying a new application does not have to be very complex. Imagine that [Figure 8-21](#) represents traffic flows that an applications team says should be allowed between EPGs that will together form a very basic in-house application.



**Figure 8-21** Valid Traffic Flows for a Hypothetical Multi-Tier Application

### Note

Figure 8-21 is not intended to represent a real-world application, nor is it meant to represent the network architecture of an actual game. It is meant only to provide a conceptual understanding of contracts. That is why the majority of ports shown are in the dynamic port range.

Say that the team deploying this application wants to keep the environment isolated for the time being. All application tiers reside in a VRF instance that has no Layer 3 connectivity with the outside world. For this reason, the deployment team has decided to stage a set of client virtual machines that can connect to the frontend components of the overall solution and mimic customer systems.

The first set of requirements, therefore, relates to client virtual machine connectivity into the environment. EPG-Client-VMs will serve as the EPG for such endpoints and should be able to communicate with EPG-Login-Servers on ports 80 and 443 and also with EPG-Gaming-Servers on port 60001. Both TCP and UDP traffic from EPG-Client-VMs to EPG-Gaming-Servers should be allowed.

The second set of requirements relates to bidirectional communications between servers that are part of the gaming infrastructure. Endpoints in EPG-Gaming-Servers, for example, need to be able to communicate with EPG-Chat-Servers via UDP port 60004. Likewise, EPG-Login-Servers, EPG-Gaming-Servers, and EPG-Chat-Servers need to be able to communicate with a number of database servers in EPG-DB-Servers via port 60003. Only TCP communication with database servers should be allowed. Finally, endpoints in EPG-Login-Servers should be able to connect to those in EPG-Gaming-Servers bidirectionally via TCP port 60007.

The final set of requirements relates to patch management. EPG-Login-Servers, EPG-Gaming-Servers, and EPG-Chat-Servers should all be allowed to initiate bidirectional communication with EPG-Patch-Servers via TCP on port 60006. UDP port 60005 to EPG-Patch-Servers and UDP port 60010 from EPG-Patch-Servers to the three noted endpoint groups also need to be opened.

# Configuring Filters for Bidirectional Application

The first set of requirements listed is quite basic. To create a filter for HTTP and HTTPS traffic, navigate to the tenant in question, double-click the Contracts menu, right-click Filters, and select Create Filter.

## Note

Filters and contracts can also be created in the common tenant for reuse across all tenants, but using the common tenant and associated caveats are beyond the scope of the DCACI 300-620 exam.

Figure 8-22 shows the configuration of a single filter with two filter entries: one for HTTP traffic and the other for HTTPS. Recall that filters are only for matching traffic. They do *not* determine what to do with traffic.

The screenshot shows the Cisco Application Policy Infrastructure Controller (APIC) web interface. The top navigation bar includes tabs for System, Tenants (which is selected), Fabric, Virtual Networking, L4-L7 Services, Admin, Operations, Apps, and Integrations. On the far right, there are icons for admin, search, and other system functions. Below the navigation is a 'Create Filter' dialog box. The 'Name' field contains 'HTTP-and-HTTPS'. The 'Entries' section displays two rows of filter entries:

Name	Alias	EtherType	ARP Flag	IP Protocol	Match Only Fragments	Stateful	Source Port / Range	Destination Port / Range	TCP Session Rules
HTTP	IP			tcp	False	False	unspecified	unspecified	http
HTTP	IP			tcp	False	False	unspecified	unspecified	https

At the bottom right of the dialog are 'Cancel' and 'Submit' buttons.



## **Figure 8-22 A Unidirectional Filter Whose Return Traffic Will Be Allowed via Subject**

Various columns can be populated for a filter entry. When you do not populate a column, ACI does not verify the associated part or parts of the packet. Some of the most important columns are as follows:



- **Name:** This is the (mandatory) name for the filter entry.
- **EtherType:** This field allows you to filter on EtherType. Available EtherType options include IP, IPv4, IPv6, ARP, FCoE, and Unspecified. For example, you might want to classify FCoE traffic in a filter to be able to drop such traffic to a server but at the same time allow ARP and IP traffic.
- **ARP Flag:** This field allows matching of ARP traffic, using the options ARP Reply, ARP Request, or Unspecified.
- **IP Protocol:** Available protocol options include TCP, UDP, ICMP, EIGRP, PIM, and Unspecified.
- **Match Only Fragments:** This field allows you to match only packet fragments. When it is enabled, the rule applies to any IP fragment, except the first.
- **Stateful:** This setting takes effect only when ACI is extended into hypervisors using Cisco AVE or Cisco AVS. By itself, ACI hardware performs stateless filtering.

(Stateful filtering is described in more detail later in the chapter.)

- **Source Port/Range and Destination Port/Range:** These fields allow you to define a single port by specifying the same value in the From and To fields, or you can define a range of ports from 0 to 65535 by specifying different values in the From and To fields.
- **TCP Session Rules:** This field allows you to specify that ACI should match the traffic only if certain TCP flags are present in the packet. The available options for matching are Synchronize, Established, Acknowledgment, Unspecified, Reset, and Finish.



If you select the most generic EtherType option, Unspecified, all other fields are grayed out, and the resulting filter matches all traffic. To be able to match on specific TCP or UDP ports, it is crucial to first set EtherType to IP, IPv4, or IPv6. Otherwise, the IP Protocol and Match Only Fragments fields are grayed out because they do not apply to the other EtherType options. The same concept holds for ARP traffic. It is only when EtherType is set to ARP that a user can specify an ARP flag on which to match traffic. Likewise, the Stateful checkbox and the TCP Session Rules field appear grayed out until TCP is selected in the IP Protocol field.



To understand Source Port/Range and Destination Port/Range, you need to put these fields into the context of traffic flow. For instance, when looking at traffic from a

consumer (client) to a provider (server), the Source Port/Range field refers to the port or range of ports on the client side that should be matched to allow the clients to talk to servers. Because client-side port selection is almost always dynamic, selection of the option Unspecified for this field makes the most sense for filters applied in the consumer-to-provider direction. With this same traffic direction in mind, the Destination Port/Range field refers to ports that need to remain open on the provider side. Selection of separate entries for HTTP and HTTPS in [Figure 8-22](#) is based on the fact that ports 80 and 443 are not subsequent ports and therefore do not fall into a range.

Thinking about that logic, it may be clear that the filter depicted is just one side of the equation. What happens to return traffic from the server? The answer is that ACI *is* able to create an equivalent rule to also allow traffic in the reverse direction. This is why filters created for bidirectional application to contracts often only specify destination ports.

## **Configuring Subjects for Bidirectional Application of Filters**

Subjects are bound to contracts and are not reusable, even though the contracts to which they are bound are reusable. You create subjects as part of the contract creation process. To create a contract, navigate to the tenant in question, double-click Contracts, right-click Standard, and select Create Contract.

[Figure 8-23](#) shows the Create Contract page. On this page, you select a name for the contract, select a scope, and click the + sign next to Subjects to create a subject.

## Create Contract

Name:

Alias:

Scope:

QoS Class:

Target DSCP:

Description:

Tags:

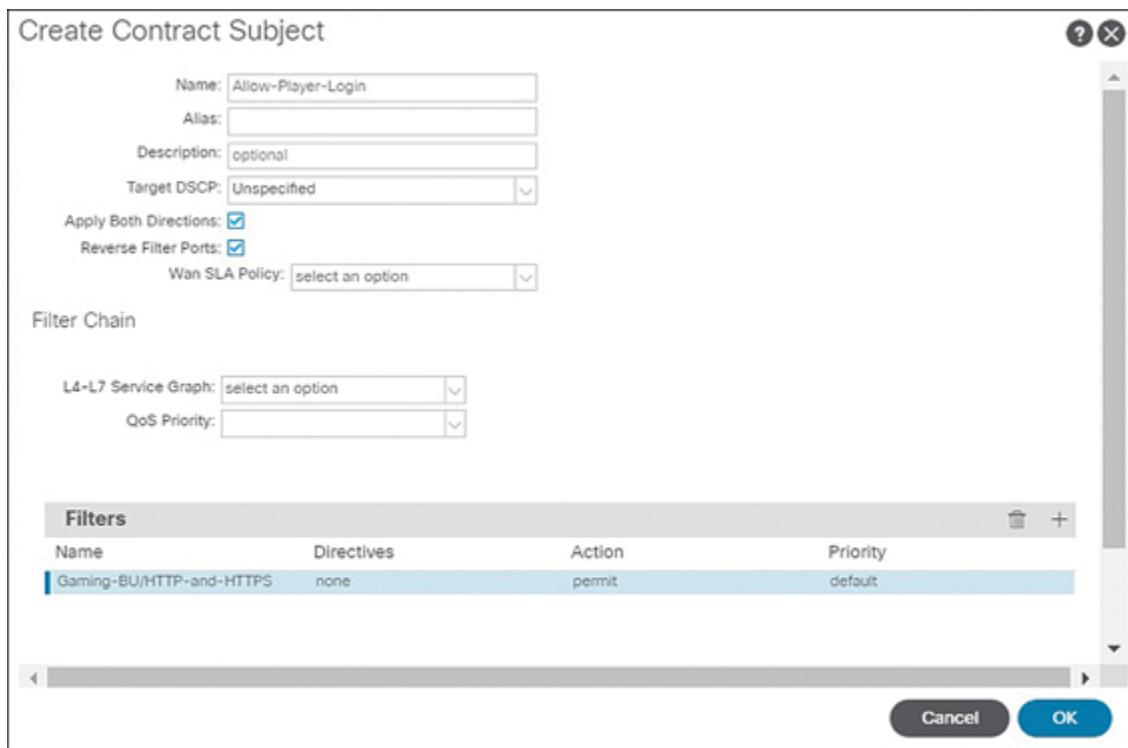
Subjects:

Name	Description

Cancel
Submit

## **Figure 8-23** The Create Contract Page

[Figure 8-24](#) shows the Create Contract Subject page. Here, you add all relevant filters in the Filters section. When you add multiple filters to a subject, you are telling ACI to take the same actions on traffic matched by any of the filters.



**Figure 8-24** Creating a Subject for a Contract

The following four columns appear in the Filters view at the bottom of the Create Contract Subject page:



- **Name:** This is the name of the filter added to the subject.
- **Directives:** As of ACI Release 4.2, available options for this column are Log, Enable Policy Compression, and None. The Log directive enables rate-limited logging of traffic that matches the filter. This is only supported on Generation 2 and later switches. The Enable Policy Compression directive potentially reduces leaf policy CAM utilization by allowing identical filter rules to share a single TCAM entry even if applied to multiple different pairs of provider and consumer EPGs. This comes at the

expense of logging data granularity and is supported only on Nexus 9300 series FX and later switches. If you do not select any of the noted directives, the filter shows up with the default setting None.

- **Action:** Available options are Permit and Deny. The Permit option allows matched traffic through. The Deny option drops traffic matched by the filter.
- **Priority:** When a Deny action has been selected for a filter, the Priority field defines the level of the precedence of the specific filter action. The Priority field for a filter is grayed out when the action is Permit.

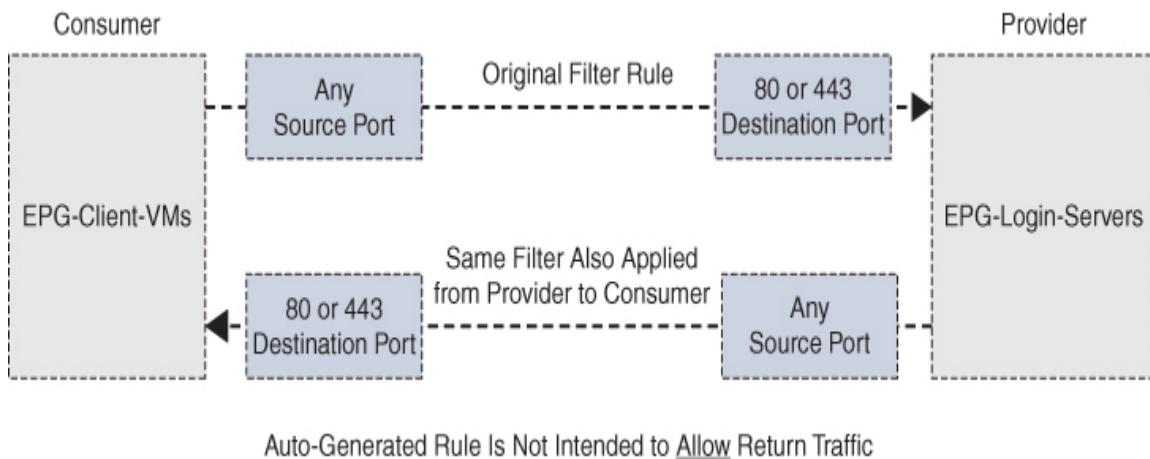
Aside from the Filters section, you may have noticed two critical checkboxes in [Figure 8-24](#): Apply Both Directions and Reverse Filter Ports. These two checkboxes determine whether the selected filter actions are applied bidirectionally.

## **Understanding Apply Both Directions and Reverse Filter Ports**

There have been many instances of engineers disabling the Reverse Filter Ports checkbox and inadvertently breaking connectivity between EPGs. The reason usually turns out to be misinterpretation. Undeniably, it is easy to misinterpret the text Apply Both Directions to mean that communication in the return direction should also be allowed. However, this does *not* align with how ACI applies contracts.

Consider the filter entries created in [Figure 8-22](#), earlier in the chapter. These filter entries allow traffic sourced from any port to reach port 80 or 443 on the destination side in one direction. If this same filter were to be applied in both the consumer-to-provider direction as well as the provider-to-consumer direction on two EPGs, both EPGs could

communicate with one another on destination ports 80 and 443 unidirectionally. This is what happens when you keep the Apply Both Directions checkbox enabled but disable the Reverse Filter Ports checkbox. [Figure 8-25](#) illustrates the resulting communication.

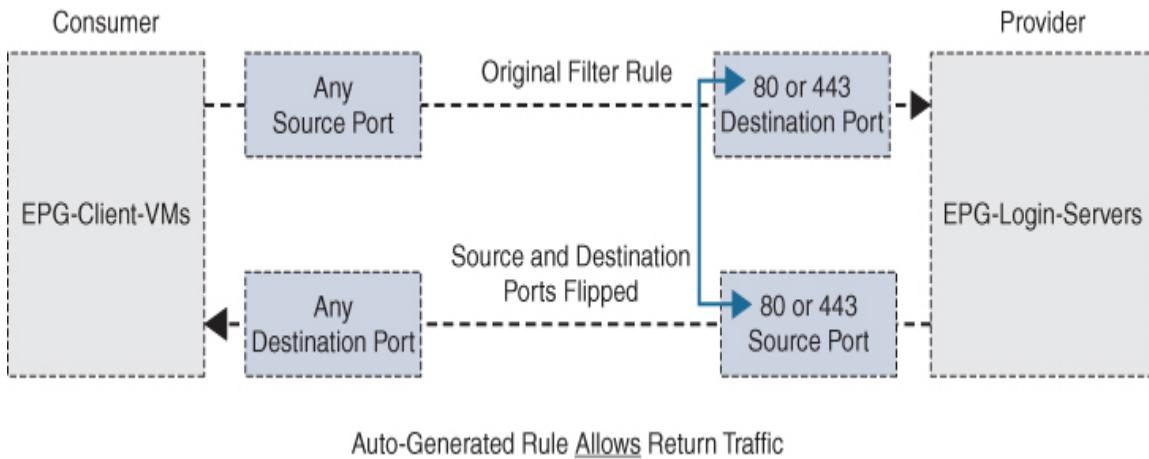


**Figure 8-25** *Apply Both Directions Enabled and Reverse Filter Ports Disabled*

Although some applications benefit from unidirectional flows, there are not many realistic use cases for applying a single unidirectional filter toward a destination port bidirectionally across multiple EPGs.

Reverse Filter Ports complements the Apply Both Directions feature by swapping the ports in the Source Port/Range and Destination Port/Range fields with one another in the return direction, thus truly enabling return traffic to flow. [Figure 8-26](#) illustrates what ACI does when Apply Both Directions and Reverse Filter Ports are both enabled.

## Key Topic



**Figure 8-26** Apply Both Directions and Reverse Filter Ports Enabled

The bottom line is that you should exercise caution when disabling Reverse Filter Ports.

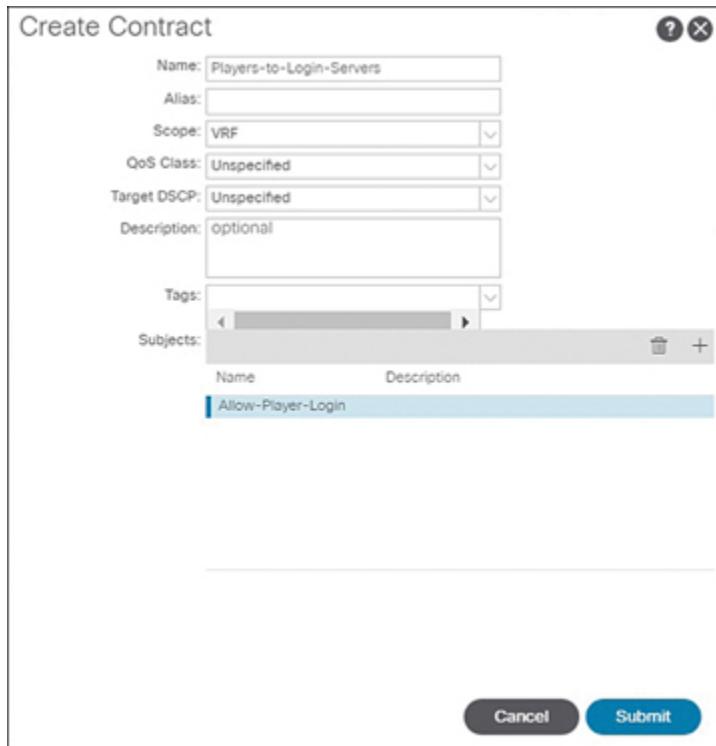
### Note

If you are struggling to put these concepts into words, it helps to interpret Apply Both Directions as “apply these filters as is both in the consumer-to-provider and provider-to-consumer directions.” Translate Reverse Filter Ports as “reverse the filter source and destination ports for return traffic.”

Because there is no reason for Reverse Filter Ports to be used without first applying a filter bidirectionally, this option is grayed out when Apply Both Directions is disabled.

## Verifying Subject Allocation to a Contract

After you define one or more contract subjects, these subjects should appear in the Subjects section of the contract. [Figure 8-27](#) shows that the subject Allow-Player-Login from [Figure 8-24](#) has been added to a contract called Players-to-Login-Servers.



**Figure 8-27** Confirming That a Subject Has Been Created and Added to a Contract

This illustration should reinforce the idea that multiple subjects can be applied to a contract. Some subjects may permit forwarding of particular traffic, and others may deny forwarding. Yet other subjects may punt traffic via PBR to stateful services devices such as firewalls, change QoS markings, or perform some other function on the traffic.

After adding the desired subjects and verifying the contract scope, click Submit to execute creation of the contract.

## Assigning Contracts to EPGs

After you create a contract, you need to assign the contract to relevant EPGs. Based on the example in [Figure 8-21](#), the contract Players-to-Login-Servers should be applied to EPG-Login-Servers as a provided contract and to EPG-Client-VMs as a consumed contract.

[Figure 8-28](#) shows the contract being added as a provided contract to EPG-Login-Servers. To allocate a contract to an EPG in the provider/provided direction, double-click on the EPG to expose its subfolders, right-click Contracts, and select Add Provided Contract. Then, select the contract and click Submit.



### Add Provided Contract

Contract:  Type at least 4 characters to select contracts

QoS:

Contract Label:

Subject Label:

**Figure 8-28** Adding a Contract to an EPG in the Provided Direction

The contract then needs to be consumed by one or more EPGs. [Figure 8-29](#) shows the contract being consumed by EPG-Client-VMs. To accomplish this, you double-click the EPG, right-click on the Contracts subfolder, select Add Consumed Contract, select the desired contract, and click Submit.



The screenshot shows the Cisco Application Policy Infrastructure Controller (APIC) web interface. The top navigation bar includes tabs for System, Tenants (which is selected), Fabric, Virtual Networking, L4-L7 Services, Admin, Operations, Apps, and Integrations. The top right shows the user is logged in as 'admin'. Below the navigation is a search bar and filter buttons for common, Gaming-BU, infra, Production, and mgmt. The main left sidebar shows the 'Gaming-BU' tenant structure, including 'Application Profiles' (Multiplayer-App1, Application EPGs, EPG-Chat-Servers, EPG-Client-VMs), 'Domains (VMs and Bare...)', 'EPG Members', 'Static Ports', 'Static Leafs', 'Fibre Channel (Paths)', 'Contracts' (selected), 'Static Endpoint', 'Subnets', 'L4-L7 Virtual IPs', 'L4-L7 IP Address Pool', and 'EPG-DB-Servers'. The right panel is titled 'Contracts' and displays a table of contracts. The table has columns for Tenant Name, Contract Name, Contract Type, Provided / Consumed, QoS Class, and State. A single row is visible: 'Gaming-BU' with 'Players-to-Login-Servers' as the contract name, 'Contract' as the type, 'Consumed' as the status, 'Unspecified' as the QoS class, and 'formed' as the state.

**Figure 8-29** An EPG Functioning as Consumer of a Contract

## Understanding the TCP Established Session Rule

The contract just applied does whitelist client to server communication to destination ports 80 and 443. However, some engineers may find that the provider will also be able

to initiate communication back to the consumer if it sources its communication from port 80 or 443. This could be considered an issue if the system is ever compromised because ACI is a stateless firewall.



To ensure that a provider is not able to initiate TCP communication toward a consumer by sourcing the session from a port intended to be opened as a destination port, one thing you can do is to allow return traffic only on the condition that the session has already been established or is in the process of being established. This can be done using a filter that has **Established** configured in the TCP Session Rules column.



The **Established** keyword adds the extra condition of matching traffic based on the control bits in the TCP header. **Established** matches TCP control bits ACK and RST.

## **Creating Filters for Unidirectional Application**

Let's take a more in-depth look at applying a contract to specific EPGs using the **Established** TCP Session Rules using the requirements for EPG-Client-VMs communication with EPG-Login-Servers. It is clear that the filter for consumer-to-provider communication would be the same as what is shown in [Figure 8-22](#).

[Figure 8-30](#) shows the complementary filter to that shown in [Figure 8-22](#), matching the desired return traffic from EPG-Login-Servers.

Create Filter

Name:	HTTP-and-HTTPS-Established																																	
Alias:																																		
Description:	optional																																	
Tags:	<input type="text"/> enter tags separated by comma																																	
Entries: <table border="1"> <thead> <tr> <th>Name</th> <th>EtherType</th> <th>IP Protocol</th> <th>Match Only Fragments</th> <th>Stateful</th> <th>Source Port / Range</th> <th>Destination Port / Range</th> <th>TCP Session Rules</th> </tr> </thead> <tbody> <tr> <td>HTTP-EST</td> <td>IP</td> <td>tcp</td> <td>False</td> <td>False</td> <td>80</td> <td>80</td> <td>unspecified</td> </tr> <tr> <td>HTTPS-EST</td> <td>IP</td> <td>tcp</td> <td>False</td> <td>False</td> <td>443</td> <td>443</td> <td>unspecified</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Established</td> </tr> </tbody> </table>			Name	EtherType	IP Protocol	Match Only Fragments	Stateful	Source Port / Range	Destination Port / Range	TCP Session Rules	HTTP-EST	IP	tcp	False	False	80	80	unspecified	HTTPS-EST	IP	tcp	False	False	443	443	unspecified								Established
Name	EtherType	IP Protocol	Match Only Fragments	Stateful	Source Port / Range	Destination Port / Range	TCP Session Rules																											
HTTP-EST	IP	tcp	False	False	80	80	unspecified																											
HTTPS-EST	IP	tcp	False	False	443	443	unspecified																											
							Established																											
<input type="button" value="Cancel"/> <input type="button" value="Submit"/>																																		

**Figure 8-30** Filter for Established Return Traffic from the EPG-Login-Servers



Notice that the source and destination ports for each entry in [Figure 8-30](#) have been reversed. This is a very important point when applying a filter unidirectionally as ACI does not reverse source and destination ports.

## Configuring Subjects for Unidirectional Application of Filters



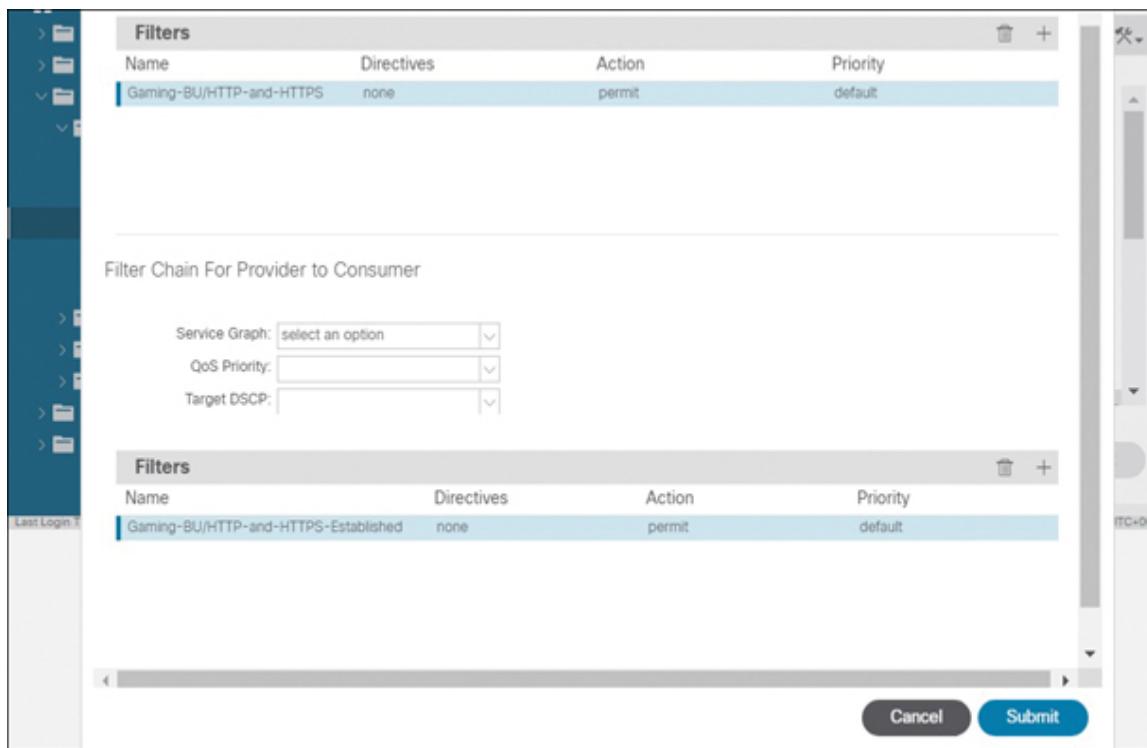
Use of the new filter created in the previous section requires a new contract but with Apply Both Directions and Reverse Filter Ports disabled. [Figure 8-31](#) shows that once these two checkboxes are disabled, separate filters can be applied in the consumer-to-provider direction versus the provider-to-consumer direction.

The screenshot shows the 'Create Contract Subject' dialog box. At the top, there are fields for Name (Allow-Player-Login), Alias, Description (optional), and Target DSCP (Unspecified). Below these are checkboxes for 'Apply Both Directions' (unchecked) and 'Reverse Filter Ports' (unchecked). A 'Wan SLA Policy' dropdown is set to 'select an option'. The next section, 'Filter Chain For Consumer to Provider', contains fields for Service Graph (select an option), QoS Priority (select an option), and Target DSCP (select an option). At the bottom, a 'Filters' table is shown with one entry:

Name	Directives	Action	Priority
Gaming-BU/HTTP-and-HTTPS	none	permit	default

**Figure 8-31** New Contract with Filter in the Consumer-to-Provider Direction

Whereas [Figure 8-31](#) shows the filter that ACI applies in the consumer-to-provider direction, [Figure 8-32](#) shows the filter with the Established keyword applied in the reverse direction.



**Figure 8-32 A Separate Filter Applied in the Provider-to-Consumer Direction**

Once applied as a provided contract to EPG-Login-Servers and as a consumed contract to EPG-Client-VMs, this contract prevents servers in EPG-Login-Servers from initiating any form of communication with the outside world. This, of course, does not hold if the system is someday compromised by somebody who is able to craft packets with either the TCP ACK or RST flags set and knows to source them from port 80 or 443.

## Additional Whitelisting Examples

Let's take a look at the remaining requirements from [Figure 8-21](#), shown earlier in this chapter.

To complete the set of requirements dealing with communication from and to EPG-Client-VMs, this book needs to address how you can allow both TCP and UDP traffic from

EPG-Client-VMs to EPG-Gaming-Servers on destination port 60001 as well as relevant return traffic. There is nothing unique about this requirement. Let's say, however, that you want to use a bidirectional contract using Apply Both Directions and Reverse Filter Ports to achieve this. It is important to understand that just because ACI stateless filtering would allow return traffic for a port as a result of a TCP filter that doesn't have the Established TCP Sessions Rules parameter set doesn't mean that the filter would also allow UDP traffic over the same port. Hence, [Figure 8-33](#) shows that both TCP and UDP filter entries would be needed if both are desired.

The screenshot shows the Cisco Application Policy Infrastructure Controller (APIC) web interface. The top navigation bar includes tabs for System, Tenants, Fabric, Virtual Networking, L4-L7 Services, Admin, Operations, Apps, and Integrations. The 'Tenants' tab is selected, and the URL bar shows the tenant name 'Gaming-BU'. The left sidebar displays a tree view of tenant resources under 'Gaming-BU', including Application Profiles, Application EPGs (EPG-Chat-Servers, EPG-Client-VMs), Domains, EPG Members, Static Ports, Static Leafs, Fibre Channel Paths, Contracts, Static Endpoint, Subnets, L4-L7 Virtual IPs, L4-L7 IP Address Pool, and EPG-DB-Servers. The main content area is titled 'Contracts' and lists a single entry:

Tenant Name	Contract Name	Contract Type	Provided / Consumed	QoS Class	State
Gaming-BU	Players-to-Login-Servers	Contract	Consumed	Unspecified	formed

**Figure 8-33** Both TCP and UDP Filter Entries Needed If EPG Talks on Both for a Port

The second set of requirements for communication between EPG-Login-Servers, EPG-Gaming-Servers, and EPG-DB-Servers is very basic. However, one point to make here is that, in practice, engineers often enable this type of TCP

communication by creating a filter addressing the consumer-to-provider traffic flow and then apply it to the relevant EPGs via a unidirectional subject. TCP return traffic for all EPGs within the associated VRF instance is then enabled using a separate filter with the Established keyword. To do this, you can use a construct called `vzAny`, which is discussed in [Chapter 10](#).

The next requirement relates to UDP communication from EPG-Gaming-Servers to EPG-Chat-Servers on port 60004. This would most likely be implemented using a bidirectional contract. It is important to note that provider EPG-Chat-Servers would still be able to initiate communication with endpoints in EPG-Client-VMs as a result of the required bidirectional contract. But there is nothing that can be done with the Established parameter since UDP is connectionless.

Finally, let's look at the glorified patch server that needs to be able to communicate on ports 20 and 21. Without going into the details of how FTP works, it is important to understand that some types of communication greatly benefit from stateful firewalling. If a stateful firewall were placed between ACI and these FTP servers, the stateful firewall could snoop the FTP control connection (TCP port 21) and get the data connection details. It could then prevent any connection to port 20 unless it is an authentic data connection. Either way, the filter to make this communication possible would look something like the one in [Figure 8-34](#).

Create Filter

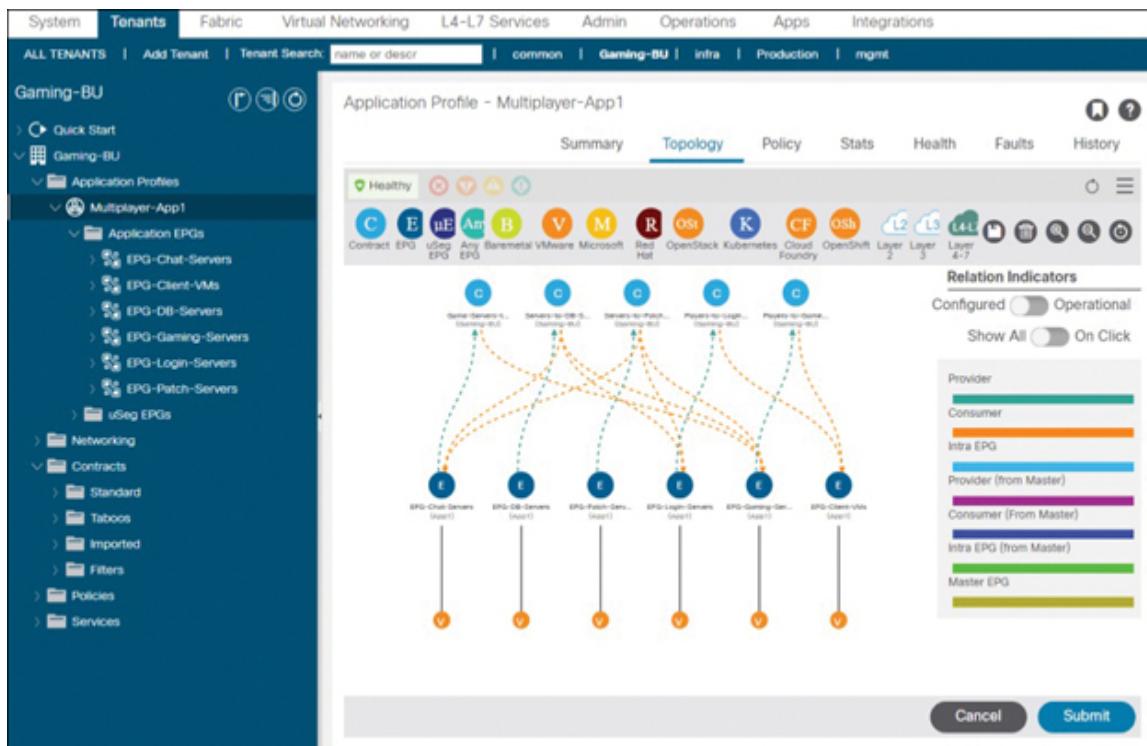
Name:	FTP	?		X																						
Alias:																										
Description:	optional																									
Tags:	enter tags separated by comma																									
Entries: <span style="float: right;">Delete +</span> <table border="1"> <thead> <tr> <th>Name</th> <th>Alias</th> <th>EtherType</th> <th>ARP Flag</th> <th>IP Protocol</th> <th>Match Only Fragments</th> <th>Stateful</th> <th>Source Port / Range</th> <th>Destination Port / Range</th> <th>TCP Session Rules</th> </tr> </thead> <tbody> <tr> <td>FTP</td> <td>IP</td> <td></td> <td></td> <td>tcp</td> <td>False</td> <td>False</td> <td>unspecified</td> <td>unspecified</td> <td>20</td> <td>21</td> <td>Unspecified</td> </tr> </tbody> </table>					Name	Alias	EtherType	ARP Flag	IP Protocol	Match Only Fragments	Stateful	Source Port / Range	Destination Port / Range	TCP Session Rules	FTP	IP			tcp	False	False	unspecified	unspecified	20	21	Unspecified
Name	Alias	EtherType	ARP Flag	IP Protocol	Match Only Fragments	Stateful	Source Port / Range	Destination Port / Range	TCP Session Rules																	
FTP	IP			tcp	False	False	unspecified	unspecified	20	21	Unspecified															
<span>Cancel</span> <span>Submit</span>																										

**Figure 8-34** Filter Matching on TCP Ports 20 and 21

If ACI were to place a firewall in front of the traffic, it would be in the form of PBR. This would be possible using a service graph associated with the subject.

## Verifying Contract Enforcement

Through the GUI, you can verify contract allocation to EPGs within an application profile by simply navigating to the Topology menu, as shown in [Figure 8-35](#).



**Figure 8-35** Viewing Contracts Applied to EPGs Within an Application Profile

Unfortunately, this view does not actually indicate whether contracts are applied in hardware on a particular leaf switch. The switch CLI **show zoning-rule** command does, however. Example 8-14 shows a variation of this command. (Several relatively insignificant columns have been removed from the output in order to fit it on a printed page.)

#### Example 8-14 Verifying Hardware Enforcement of a Contract on a Leaf Switch

[Click here to view code image](#)

```
LEAF101# show zoning-rule contract Players-to-Login-Servers
+-----+-----+-----+-----+-----+
| SrcEPG | DstEPG | Dir     | operSt | Scope   |      Name
+-----+-----+-----+-----+-----+
| Action |
```

-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+
16389   32780   uni-dir   enabled   2228225   Players-to-
Login-Servers   permit
32780   16389   uni-dir   enabled   2228225   Players-to-
Login-Servers   permit
-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+

In this output, SrcEPG and DstEPG represent the sclass or pcTag of the source and destination EPGs, respectively. The scope is the VRF instance VNID. In this case, the administrator could have verified application of all contracts within the VRF instance by using the **show zoning-rule scope 2228225** command.

## Understanding the Stateful Checkbox in Filter Entries

Just as it is important to understand what ACI is, it is also important to understand what it is not. ACI, by itself, is *not* a stateful firewall. At best, you can consider it semi-stateful if you associate TCP control bit checking with connection state. Most often, though, it is termed *stateless*. This should not be a surprise because ACI does not track TCP sequence numbers and other aspects of TCP sessions.



So, what is the Stateful checkbox that can be enabled when creating filters? This feature takes effect only in conjunction with Cisco AVS and AVE. When ACI is extended into one of these stateful firewalling hypervisor solutions, AVS and AVE can use server resources to enable a connection-tracking

capability. Not only is the overall solution then able to track TCP handshake state and sequence numbers, it is also able to inspect traffic and dynamically open negotiated TCP ports. In the case of FTP traffic, for example, this would enable port 20 to be usable for only valid data communication between the server and client.

Another point about ACI is that it is not intended to do application layer inspection. If you whitelist port 80 to a provider EPG, ACI would not care to drop traffic destined to the provider just because the consumer EPG was found to be sending a payload other than HTTP.

At the end of the day, ACI is here to do line-rate packet filtering. Its EPG architecture is a basic zone-based firewall of sorts. It can be configured to forward flows that require inspection to next-generation firewalls. It can also integrate with other solutions, such as AVS and AVE, to improve the overall data center security posture. But it is not here to replace firewalls.

## **Contract Scopes in Review**

Let's revisit contract scopes in light of the implementation examples covered in this chapter. The contracts shown in this chapter were all created using the VRF scope. But what if the business unit deploying this new application actually intends to deploy multiple instances of this same application? Maybe it intends to deploy one for development, one for customer testing, and one or more for production. Would the company be able to reuse the EPGs and contracts, or would it need to do all this work over again?

Practically, customers that want to deploy an application multiple times could automate the policy deployment. There are very easy ways to generate a script based on an

application profile and all the EPGs under it. The script could then deploy the same objects with different application profile names. (Yes, EPGs under different application profiles can have the same name.)

But what about contract reuse? The answer lies in the contract scope. When you change the contract scope to Application Profile, the contracts are only enforced between EPGs that are in a common application profile. This approach helps ensure that security rules only need to be put in place once and can be updated (for example, by adding a new subject to a contract) across all relevant application profiles simultaneously. This approach underscores why contract scopes are so important.

## Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: [Chapter 17, “Final Preparation,”](#) and the exam simulation questions in the Pearson Test Prep Software Online.

## Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. [Table 8-5](#) lists these key topics and the page number on which each is found.



**Table 8-5** Key Topics for [Chapter 8](#)

---

Key Topic Element	Description	Page Number
Paragraph	Describes the key benefits of ACI endpoint learning	241
Paragraph	Provides the official definition of an endpoint in ACI	241
Table 8-3	Lists and describes the forwarding tables used in ACI	242
Paragraph	Defines a local endpoint	242
Paragraph	Defines a remote endpoint	242

Key Description Topic Element	Page Number
<a href="#">Table 8-4</a>	Summarizes the differences between local endpoints and remote endpoints
<a href="#">Paragraph</a>	Explains why Unicast Routing must be enabled for ACI to learn IP addresses of relevant endpoints
<a href="#">List</a>	Details the logic an ACI leaf uses to figure out whether to learn a remote endpoint MAC address or IP address
<a href="#">Paragraph</a>	Defines port encapsulation VLANs
<a href="#">Paragraph</a>	Describes hardware proxy logic from the perspective of spines

Key Topic Element	Description	Page Number
Paragraph	Describes why ACI uses the ARP table in conjunction with the routing table for learning devices outside ACI behind L3Outs	249
Paragraph	Describes pervasive gateways and how they are deployed	252
Paragraph	Describes pervasive routes and their significance in spine proxy forwarding	259
Paragraph	Explains how bridge domain settings such as L2 Unknown Unicast determine whether spine proxy forwarding can be used for a given communication	259
Paragraph	Details the spine proxy destination lookup process and what happens to traffic next	260

Key Description Topic Element	Page Number
Paragraph	Describes ACI ARP forwarding behavior with the Hardware Proxy parameter enabled and the ARP Flooding parameter disabled
Paragraph	Describes ACI ARP forwarding behavior with the ARP Flooding parameter enabled
Paragraph	Describes ACI ARP forwarding behavior when unicast routing has been disabled on a bridge domain
List	Lists the settings impacted when Optimize is selected in a bridge domain's Forwarding drop-down
Figure 8-15	Shows how to verify domain assignment to an EPG

Key Description Topic Element	Page Number
<a href="#">Figure 8-16</a>	Shows how to map an encapsulation to an EPG by using static paths
List	Lists and describes the static binding mode options for EPGs
List	Lists and describes the Deployment Immediacy options
<a href="#">Figure 8-17</a>	Shows how to statically map an EPG to an encapsulation on an individual port
<a href="#">Figure 8-18</a>	Shows how to verify the static path bindings for an EPG

Key Topic Element	Description	Page Number
<a href="#">Figure 8-22</a>	Shows how to create a unidirectional filter when return traffic will be allowed via subject settings	<a href="#">274</a>
List	Describes the columns available in the filter configuration view	<a href="#">274</a>
Paragraph	Describes how the filter configuration view disables selection of certain settings based on selected options	<a href="#">274</a>
Paragraph	Describes how to fill out the Source Port/Range and Destination Port/Range columns during filter configuration	<a href="#">275</a>
List	Describes the various columns that appear in the Create Contract Subject page	<a href="#">276</a>

Key Description Topic Element		Page Number
<a href="#">Figure 8-25</a>	Illustrates ACI behavior with Apply Both Directions enabled and Reverse Filter Ports disabled for a subject	<a href="#">277</a>
<a href="#">Figure 8-26</a>	Illustrates ACI behavior with both Apply Both Directions and Reverse Filter Ports enabled for a subject	<a href="#">277</a>
<a href="#">Figure 8-28</a>	Shows how to associate a contract with an EPG in the provider direction	<a href="#">279</a>
<a href="#">Figure 8-29</a>	Shows a contract associated with an EPG in the consumer direction	<a href="#">279</a>
<a href="#">Paragraph</a>	Explains how to further lock down communication on the provider end of communication by using the established TCP session rule	<a href="#">280</a>

Key Description	Page Number	
Topic Element		
Paragraph	Explains what the Established keyword does	280
Paragraph	Reiterates that ports in the source and destination columns need to be reversed when creating an additional filter for return traffic	280
Paragraph	Reiterates that Apply Both Directions and Reverse Filter Ports need to be disabled when applying two different filters for consumer-to-provider versus provider-to-consumer traffic	280
Paragraph	Explains the function of the Stateful checkbox within filters	284

## Complete Tables and Lists from Memory

There are no memory tables or lists in this chapter.

## Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

endpoint

local endpoint

remote endpoint

port encapsulation VLAN

platform-independent VLAN (PI VLAN)

ARP gleaning

pervasive gateway

pervasive route

FTag tree

# **Part III: External Connectivity**

# Chapter 9

## L3Outs

**This chapter covers the following topics:**

**L3Out Fundamentals:** This section covers concepts related to L3Outs, the subobjects that make up L3Outs, interface types, and BGP route reflection.

**Deploying L3Outs:** This section details the process of establishing routing to the outside world via user tenant L3Outs.

**Implementing Route Control:** This section covers the basics of route profiles and some use cases for route profiles in ACI.

This chapter covers the following exam topic:

- 3.2 Implement Layer 3 Out

Previous chapters address how ACI access policies control the configuration of switch downlink ports and the level of tenant access to such ports. This chapter expands on that information by addressing the implementation of L3Outs on switch downlinks to communicate subnet reachability into and out of ACI user VRF instances.

This chapter tackles a number of issues related to L3Outs, such as classifying external endpoints for contract

enforcement, implementing BGP route reflection, and basic route filtering and route manipulation.

## “Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 9-1](#) lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in [Appendix A, “Answers to the ‘Do I Know This Already?’ Questions.”](#)

**Table 9-1** “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
L3Out Fundamentals	1-4
Deploying L3Outs	5-9
Implementing Route Control	10

### Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** An ACI administrator wants to deploy an L3Out to ASR 1000 Series routers in a user VRF while ensuring that future L3Outs in other tenants can also reuse the same physical connectivity to the outside routers. Which interface type should be used?
  - a.** Routed subinterfaces
  - b.** Routed interfaces
  - c.** SVIs
  - d.** Floating SVIs
- 2.** An L3Out has been created in a user VRF and border leaf switches have learned external subnet 10.0.0.0/8 via dynamic routing and added it to their routing tables. Which of the following explains why a compute leaf that has deployed the same user VRF instance has not received that route?
  - a.** A user has configured a route profile for interleak and applied it to the L3Out.
  - b.** A default import route profile has been added to the L3Out.
  - c.** Routing protocol adjacencies between ACI and external routers never formed.

- d. BGP route reflection has not been configured for the fabric.
- 3. Which interface type is ideal for L3Out deployment when a physical firewall appliance needs to be dual-homed to a pair of leaf switches and establish routing protocol adjacencies directly with an ACI fabric?
  - a. Routed subinterfaces
  - b. Routed interfaces
  - c. SVIs
  - d. Floating SVIs
- 4. When configuring BGP route reflection in ACI, what are the two critical parameters that administrators need to define?
  - a. BGP ASN and cluster ID
  - b. Border leaf switches and L3Outs from which ACI should import routes
  - c. BGP ASN and the spines in each pod that should function as route reflectors
  - d. BGP ASN and border leafs in each pod that should function as a route reflector client
- 5. An administrator learns that when she modifies a bridge domain subnet scope to Advertised Externally and adds the bridge domain for advertisement to an OSPF L3Out, the subnet also gets advertised out an EIGRP L3Out. Which of the following options would allow advertisement of the BD subnet out a single L3Out? (Choose all that apply.)
  - a. Remove the bridge domain from the EIGRP L3Out.
  - b. Move one of the L3Outs to a different switch or set of switches.

- c. Switch the bridge domain back to Private to VRF.
  - d. Use BGP for a dedicated route map per L3Out.
- 6. True or false: The same infra MP-BGP ASN used for route reflectors is also used to establish connectivity out of BGP L3Outs unless ACI uses a *local-as* configuration.
  - a. True
  - b. False
- 7. Regarding configuration of BGP L3Outs, which of the following statements are true? (Choose all that apply.)
  - a. When establishing BGP connectivity via loopbacks, BGP peer connectivity profiles should be configured under the node profile.
  - b. ACI allows EIGRP to be configured on a BGP L3Out for BGP peer reachability in multihop scenarios.
  - c. ACI tries to initiate BGP sessions with all IP addresses in a subnet as a result of the dynamic neighbor establishment feature involving prefix peers.
  - d. ACI implements BGP subnet advertisement to outside as a redistribution.
- 8. Which statements about OSPF support and configuration in ACI are correct? (Choose all that apply.)
  - a. The administrative distance for OSPF can be modified at the node profile level.
  - b. OSPF authentication is supported in ACI and can be configured under an OSPF interface profile.
  - c. Border leaf L3Outs support VRF-lite connectivity to external routers.
  - d. ACI does not support OSPFv3 for IPv6.

- 9.** Which statements are correct regarding ACI support for BFD? (Choose all that apply.)
- a. BFD is supported for EIGRP, OSPF, and BGP in ACI.
  - b. BFD is supported on L3Out loopback interfaces.
  - c. BFD is supported for BGP prefix peers (dynamic neighbors).
  - d. BFD is supported on routed interfaces, routed subinterfaces, and SVIs.
- 10.** True or false: Route profiles are a little different from route maps on NX-OS and IOS switches and routers because they may merge configurations between implicit route maps and explicitly configured route profile match statements.
- a. True
  - b. False

## Foundation Topics

### L3Out Fundamentals

An ACI Layer 3 Out (L3Out) is the set of configurations that defines connectivity into and out of an ACI fabric via routing. The following sections discuss the key functions other than routing that an L3Out provides and the objects that comprise an L3Out.

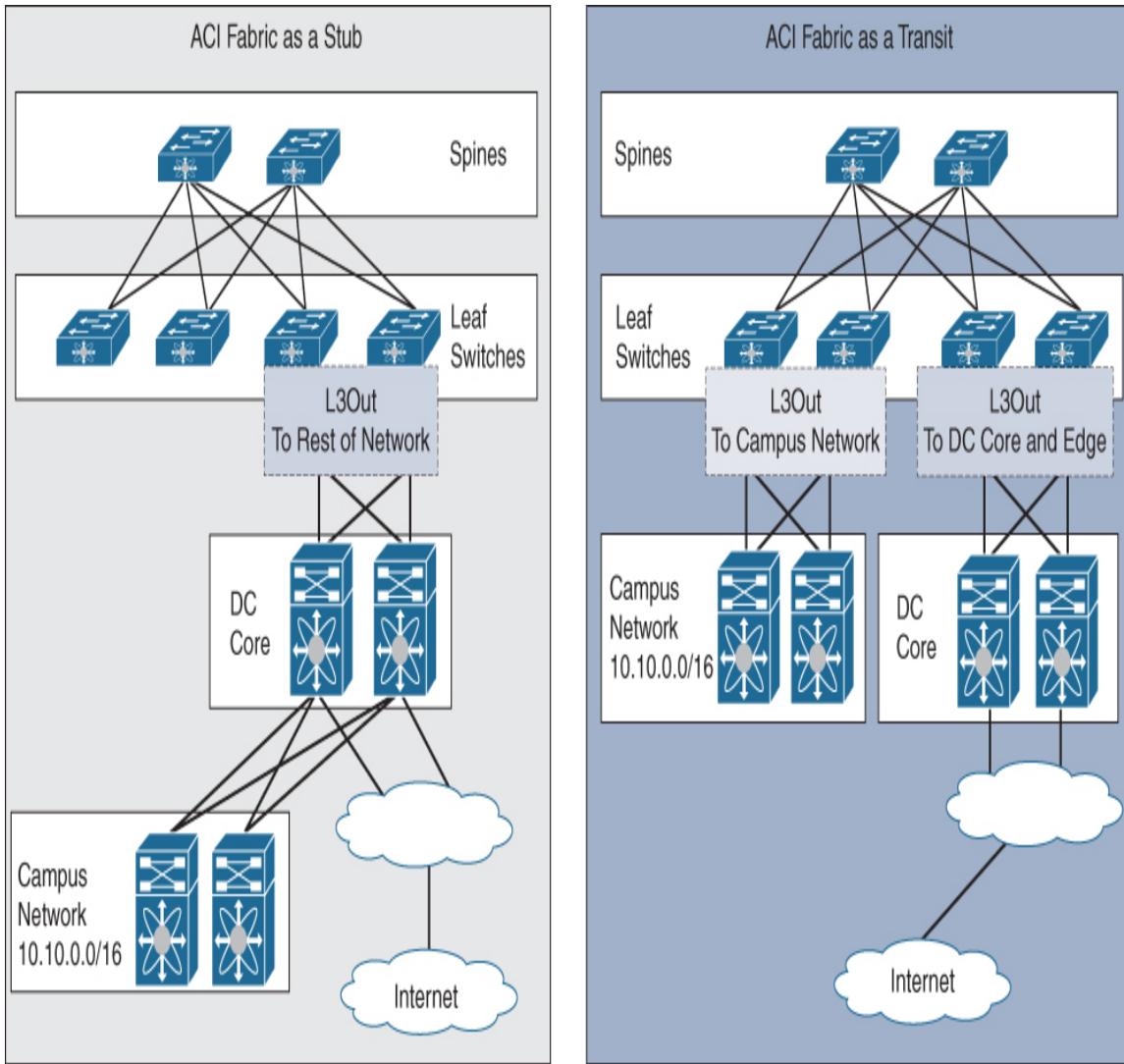
### Stub Network and Transit Routing

To better understand L3Outs, it helps to first understand the difference between a stub network and a transit network. ACI was originally built to function as a stub network. As a

stub, the intent was for ACI to house all data center endpoints but to not be used to aggregate various routing domains (for example, core layer, firewalls, WAN, campus, mainframe) within a data center.

The community of network engineers then banded together and told Cisco that ACI as a stub was not enough. Shortly afterward, Cisco began to support ACI as a transit. The idea is that ACI performs route redistribution, mostly behind the scenes, to interconnect multiple routing domains (L3Outs) and transit routes as well as traffic between L3Outs.

[Figure 9-1](#) compares ACI as a stub and ACI as a transit. On the left-hand side, ACI is depicted as a stub. The fabric has only a single L3Out. In this case, ACI does not need to do anything to campus subnet 10.10.0.0/16 except to learn it. On the right-hand side, however, the campus core layer has been depicted connecting directly into ACI using an L3Out. Data center core and edge infrastructure connect to a separate L3Out. Because the data center core and campus layers do not have any direct connectivity with one another, ACI needs to be configured to transit routes between the L3Outs for machines in the 10.10.0.0/16 subnet to be able to reach the Internet. It is important to understand that it is not the existence of multiple L3Outs that implies transit routing; it is the expectation that ACI functions as a hub and routes traffic from one L3Out to another that necessitates transit routing.



**Figure 9-1** *Understanding ACI as a Stub Versus ACI as a Transit*

### Note

Route redistribution between Layer 3 domains is a large topic, and it is not something that the majority of ACI engineers deal with on a day-to-day basis. Therefore, transit routing is beyond the scope of the Implementing Cisco Application Centric Infrastructure DCACI 300-620 exam.

## Types of L3Outs

[Chapter 5](#), “Tenant Building Blocks,” provides a basic definition of L3Outs. It may be possible to interpret that definition as suggesting that each L3Out is only able to advertise subnets that exist within a single VRF. The following points expand on the previous definition by describing the numerous categories of L3Outs you can create in ACI:

- **L3Outs in the infra tenant:** This type of L3Out is actually more of a family of L3Outs that typically associate with the overlay-1 VRF instance and enable either integration of an ACI fabric with other fabrics that are part of a larger ACI Multi-Site solution or expansion of a fabric into additional pods as part of an ACI Multi-Pod or vPod solution. The overlay-1 VRF can also be used to interconnect a fabric with its Remote Leaf switches. These solutions should all be familiar from [Chapter 2](#), “[Understanding ACI Hardware and Topologies](#).” One particular infra tenant L3Out, however, has not been mentioned before. A GOLF L3Out uses a single BGP session to extend any number of VRF instances out an ACI fabric to certain OpFlex-capable platforms, such as Cisco ASR 1000 Series routers, Cisco ASR 9000 Series routers, and Nexus 7000 Series switches. One common theme across the infra tenant family of L3Outs is that traffic for multiple VRFs can potentially flow over these types of L3Outs. Another common theme is that routing adjacencies for these L3Outs are sourced from the spines.
- **User VRF L3Outs (VRF-lite):** This type of L3Out extends Layer 3 connectivity out one or more border leaf switches. It is the most commonly deployed L3Out in ACI. Administrators typically deploy these L3Outs in

user tenants, but they can also be deployed in VRFs in the common tenant or any other VRF used for user traffic. Each L3Out of this type is bound to a single user VRF and supports a VRF-lite implementation. Through the miracle of subinterfaces, a border leaf can provide Layer 3 outside connections for multiple VRFs over a single physical interface. This VRF-lite implementation requires one protocol session per tenant. A *shared service L3Out* is the name given to any user VRF L3Out that has additional configuration to leak routes learned from external routers into a VRF other than the VRF to which the L3Out is bound. This allows external routes to be consumed by EPGs in another VRF. This feature is also referred to as *shared L3Out* because a service behind the L3Out is being shared with another VRF.

If all this seems overwhelming, don't worry. The DCACI 300-620 exam only focuses on implementation of user VRF L3Outs. Route leaking and shared service L3Outs are also beyond the scope of the exam. Nonetheless, you need to be aware of the different types of L3Outs so you can better recognize which configuration knobs are relevant or irrelevant to the type of L3Out being deployed.

### Note

The term *shared service* is used here to refer to a user tenant using a component from the common tenant. This term can also apply to consumption of common tenant objects by other tenants, but use of the term to imply route leaking is now more prevalent.

## Key Functions of an L3Out

For an L3Out to be effective in connecting an ACI fabric to outside Layer 3 domains, it needs to be able to perform the following five critical functions:

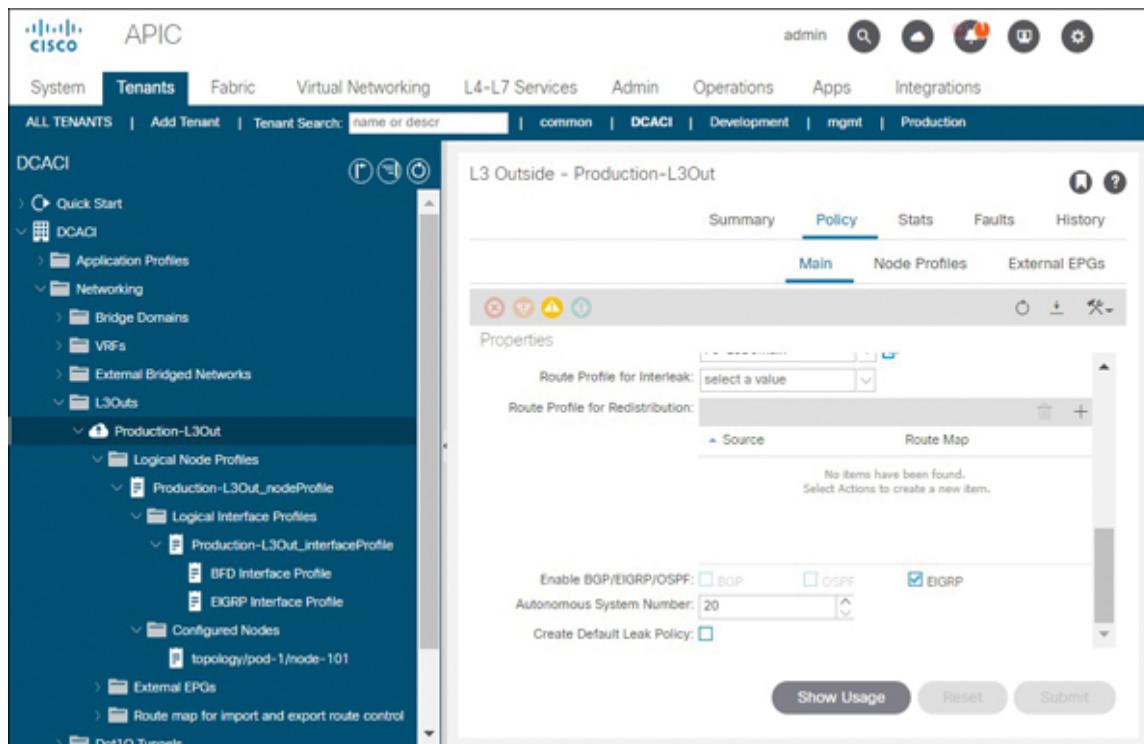


- **Learn external routes:** This entails one or more border leafs running a routing protocol and peering with one or more external devices to exchange routes dynamically. Alternatively, static routes pointing to a next-hop device outside the fabric can be configured on an L3Out.
- **Distribute external routes:** This involves ACI distributing external routes learned on an L3Out (or static routes) to other switches in the fabric using Multiprotocol BGP (MP-BGP) with VPNv4 configured in the overlay-1 VRF (tenant infra). ACI automates this distribution of external routes in the background. This distribution requires BGP route reflection.
- **Advertise ACI bridge domain subnets out an L3Out:** For external devices to have reachability to servers in the fabric, an ACI administrator needs to determine which BD subnets should be advertised out of an L3Out. ACI then automates creation of route maps in the background to allow the advertisement of the specified BD subnets via the selected L3Outs.
- **Perform transit routing:** Advertising external routes between Layer 3 domains can be achieved by using the L3Out EPG subnet scope Export Route Control Subnet. (The DCACI 300-620 exam does not cover transit routing, so neither does this book.)

- **Allow or deny traffic based on security policy:**  
Even with ACI exchanging routes with the outside world, there still needs to be some mechanism in place for ACI to classify traffic beyond an L3Out and determine whether a given source should be allowed to reach a particular destination. L3Outs accomplish this by using special EPGs referred to as external EPGs. EPGs classifying traffic beyond an L3Out are configured on the L3Outs themselves using the L3EPG scope External Subnets for External EPG.

## The Anatomy of an L3Out

Each L3Out is structured to include several categories of configuration. [Figure 9-2](#) shows a sample L3Out and its subobjects.



**Figure 9-2** *The Anatomy of an ACI L3Out*

The following points summarize the structure and objects within an L3Out:



- **L3Out root:** As shown in [Figure 9-2](#), the most critical L3Out configurations that are of a global nature can be found under the **Policy > Main** page at the root of the L3Out. Configurations you apply here include the routing protocol to enable on the L3Out, the external routed domain (L3 domain) linking the L3Out with the underlying access policies, the VRF where the L3Out should be deployed, the autonomous system number in the case of EIGRP, and the area type and area ID to be used for OSPF.
- **Logical node profiles:** The main function of a [\*\*logical node profile\*\*](#) is to specify which switches should establish routed connectivity to external devices for a given L3Out. ACI creates two subfolders under each logical node profile: Logical Interface Profiles and Configured Nodes.
- **Logical interface profiles:** An administrator can configure one or more [\*\*logical interface profiles\*\*](#) for each set of interface configurations. It is under logical interface profiles that interface IP addresses and MTU values for routing protocol peering can be configured. Protocol-specific policies such as authentication and timers for EIGRP, OSPF, and BGP can also be configured under logical interface profiles. Bidirectional Forwarding Detection (BFD) and custom policies for QoS, data plane policing, NetFlow, and IGMP can also be applied at the logical interface profile level.

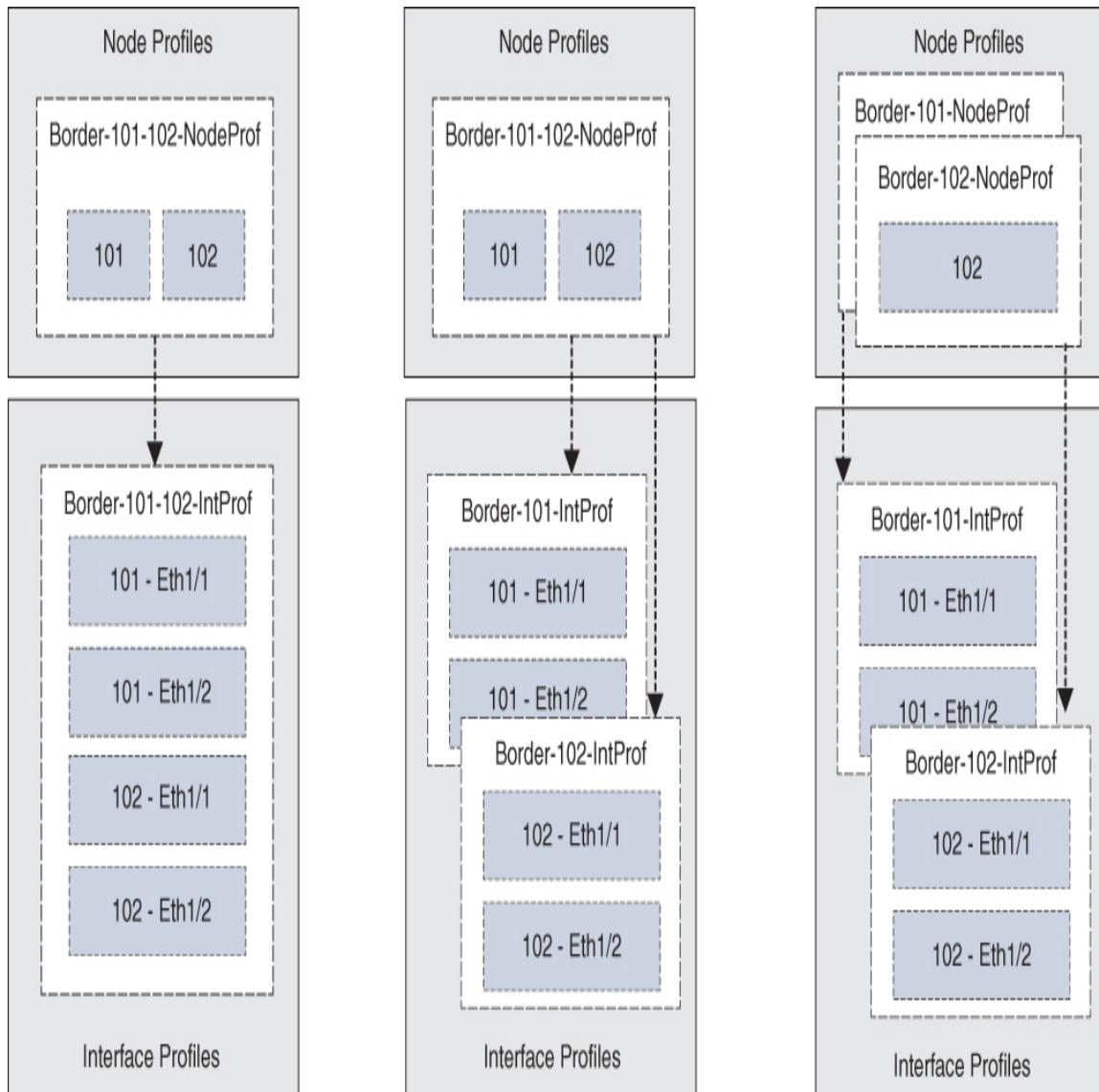
- **Configured nodes:** The Configured Nodes folder includes a single node association entry for each switch node that is part of an L3Out. Static routes, router IDs, and loopback IP addresses are all configured in node association entries in this folder.
- **External EPGs:** Any EPG that is defined directly on an L3Out can be called an external EPG. The only external EPGs of concern for the DCACI 300-620 exam are those used to classify traffic from external endpoints.
- **Route map for import and export route control:** These special route maps are applied on the entire L3Out and associated bridge domains. The two route maps that can be applied are default-import and default-export.

## Planning Deployment of L3Out Node and Interface Profiles

As mentioned earlier in this chapter, logical node profiles and logical interface profiles for user VRF L3Outs together define which switches are border leaf switches and what interface-level configurations and policies the L3Out should use for route peering with the outside world. But how are these two object types deployed side-by-side?

The first thing to remember is that logical interface profiles fall under the Logical Node Profiles folder. On the left-hand side of [Figure 9-3](#), an administrator has configured two leaf switches with node IDs 101 and 102 as border leaf switches using a single logical node profile. A single logical interface profile under the logical node profile then defines configurations for interfaces on both of these nodes. In the second design pattern, the administrator has created a single node profile but then deployed one interface profile for each switch. In the iteration on the right, separate logical

node profiles and interface profiles have been created for each individual border leaf switch. All three of these design patterns are correct.



**Figure 9-3** Logical Node Profile and Logical Interface Profile Design Patterns

Some of the confusion that occurs for engineers when deciding between these options is due to the fact that most engineers think in terms of design patterns for access policies. Deployment of an interface configuration under an

interface profile that is bound to multiple switches under the Access Policies menu can lead to the simultaneous deployment of interface configurations on multiple switches. In contrast, each interface or port channel added under an L3Out configuration requires explicit configuration before routing can be enabled on the port. Therefore, logical node profiles and logical interface profiles are used more to achieve organizational hierarchy than for automating simultaneous interface configuration.

## Understanding L3Out Interface Types

The following types of interfaces can be part of an L3Out:



- Routed subinterfaces
- Routed interfaces
- Switch virtual interfaces (SVIs)
- Floating SVIs

The design considerations for the first three options listed here are mostly the same as what you would expect when configuring connectivity between any set of Layer 3 devices.

Use of routed subinterfaces is very common in multitenancy environments because one VLAN ID and subinterface can be allocated for each user VRF L3Out, enabling multiple tenant L3Outs to flow over a single set of physical ports.

Use of router interfaces is most common in single-tenant environments or in instances when the fabric is expected to have dedicated physical connectivity to a specific switch block or segregated environment via a single L3Out.

If an administrator has trunked an EPG out a port or port aggregation, the same port or port channel cannot be used as for a routed subinterface or routed interface because the port has already been configured as a (Layer 2) switchport. Such a port or port aggregation can still be configured as part of an L3Out that leverages SVIs or floating SVIs.



### Note

It is has become very common for engineers to build ACI L3Outs using pure Layer 3 solutions such as routed interfaces and routed subinterfaces, but there are many great use cases for building L3Outs using SVIs. One common use case is establishing a redundant peering with firewalls over a vPC from a pair of border leaf switches.

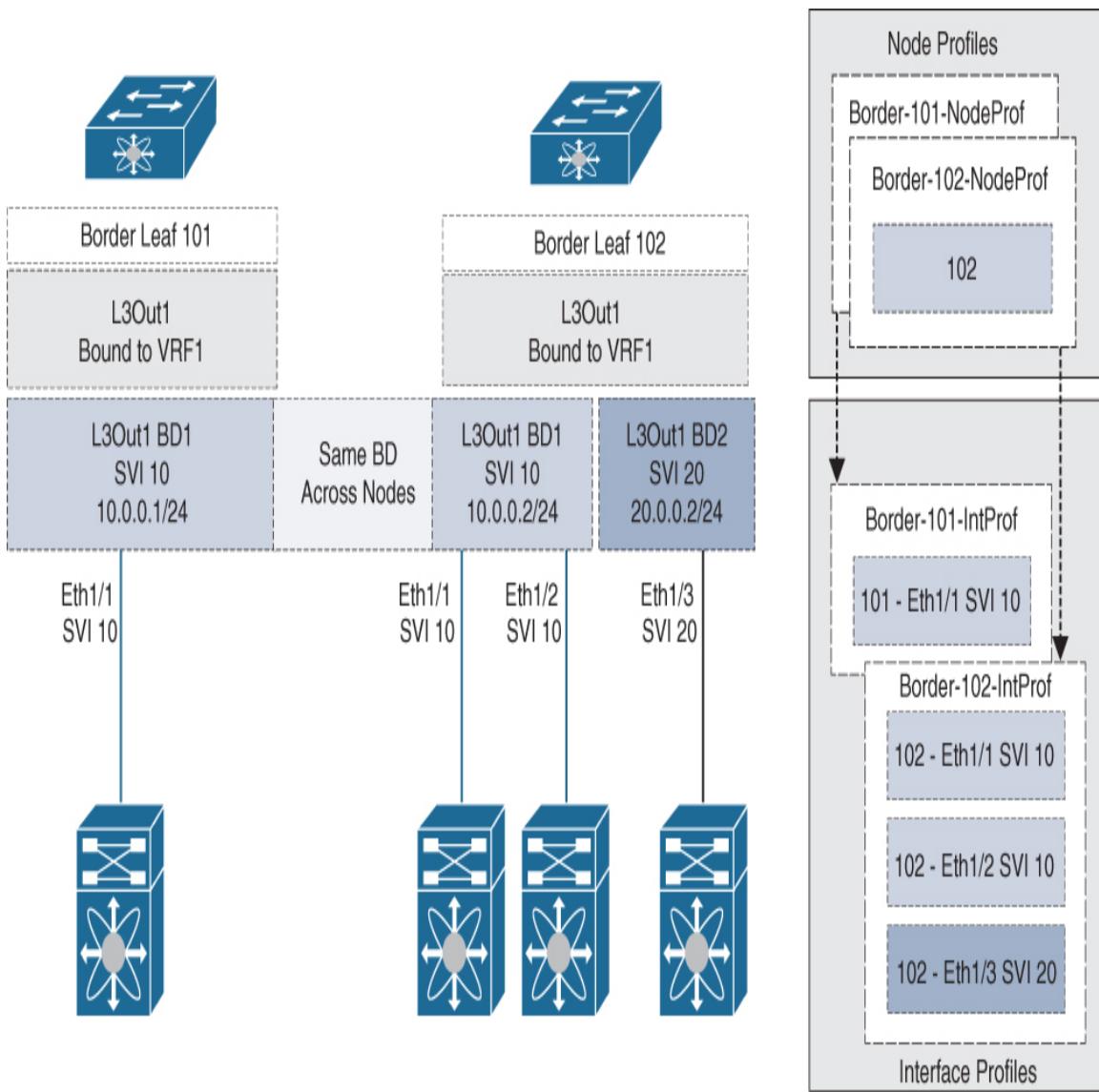
The new ***floating SVI*** option, first introduced in ACI Release 4.2(1), with further enhancements in ACI Release 5.0(1), enables users to configure an L3Out without locking down the L3Out to specific physical interfaces. This feature enables ACI to establish routing adjacencies with virtual machines without having to build multiple L3Outs to accommodate potential VM movements.

The floating SVI feature is only supported for VMM integrated environments in ACI Release 4.2(1) code. Enhancements in ACI Release 5.0(1) allow floating SVIs to be used with physical domains, eliminating the requirement for VMM integration.

## Understanding L3Out Bridge Domains

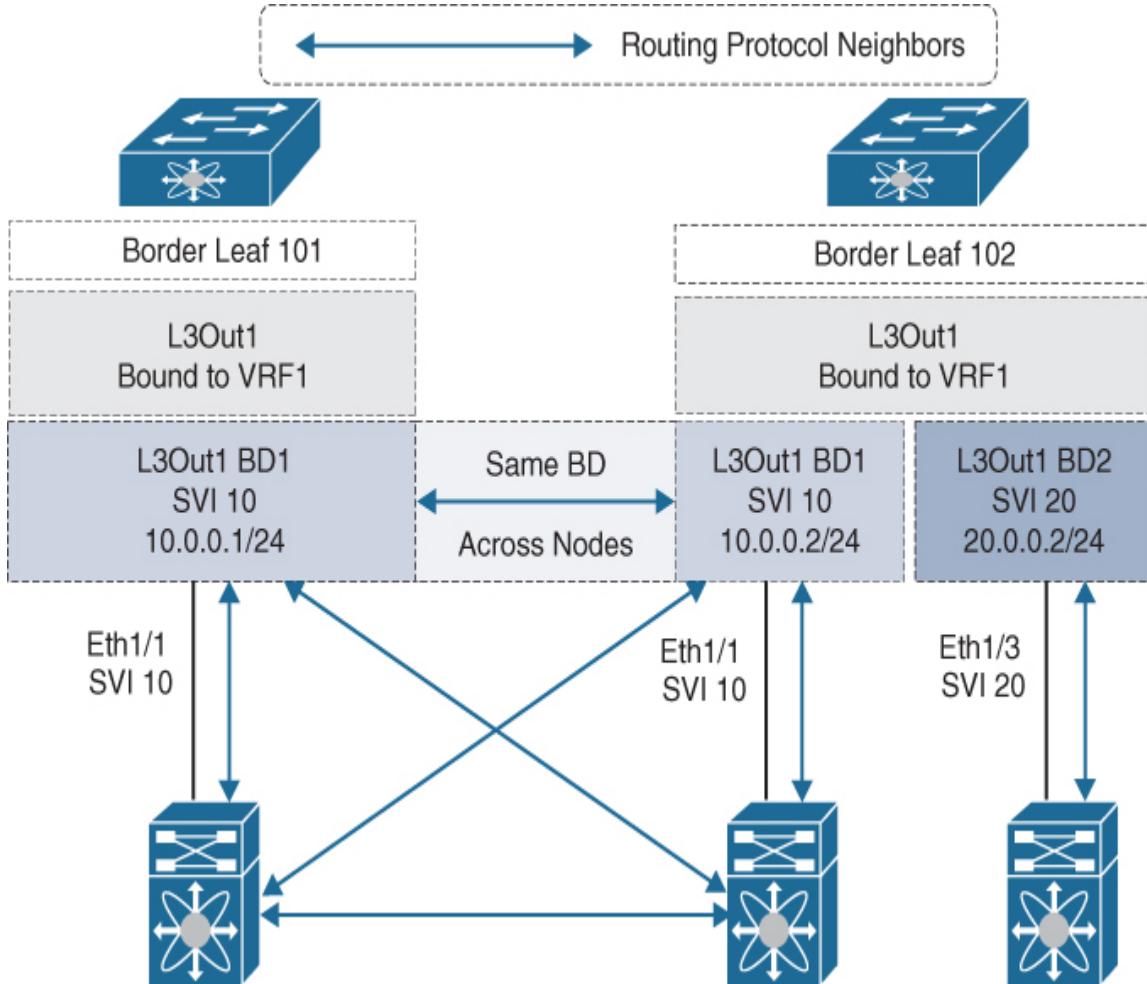
When instantiating an L3Out SVI, ACI creates an **L3Out bridge domain (BD)** internally for the SVI to provide a Layer 2 flooding domain. This BD is called an *L3Out BD* or *external BD* and is not visible to ACI administrators.

ACI creates a different L3Out BD for each encapsulation used in an SVI-based L3Out. If an administrator uses a common VLAN encapsulation for SVIs on multiple border leaf nodes in a single L3Out, ACI spans the L3Out BD and the associated flooding domain across the switches. In [Figure 9-4](#), L3Out SVI encapsulation 10, deployed to both border leafs 101 and 102, prompts ACI to place all interfaces associated with the SVI in a common flooding domain. Meanwhile, selection of encapsulation 20 for another SVI on the same L3Out triggers ACI to create a new L3Out BD with a different Layer 2 flooding domain.



**Figure 9-4** Significance of SVI VLAN Encapsulation Settings in an L3Out

Effectively, what happens as a result of this configuration is that the border leaf switches and any other routers in the stretched flooding domain establish somewhat of a full mesh of routing protocol adjacencies with one another (in the case of OSPF or EIGRP). [Figure 9-5](#) shows the neighbor adjacencies established for the VLAN 10 L3Out BD. Meanwhile, the single router connecting to the L3Out via encapsulation 20 forms only a single adjacency with ACI.

**Key Topic**

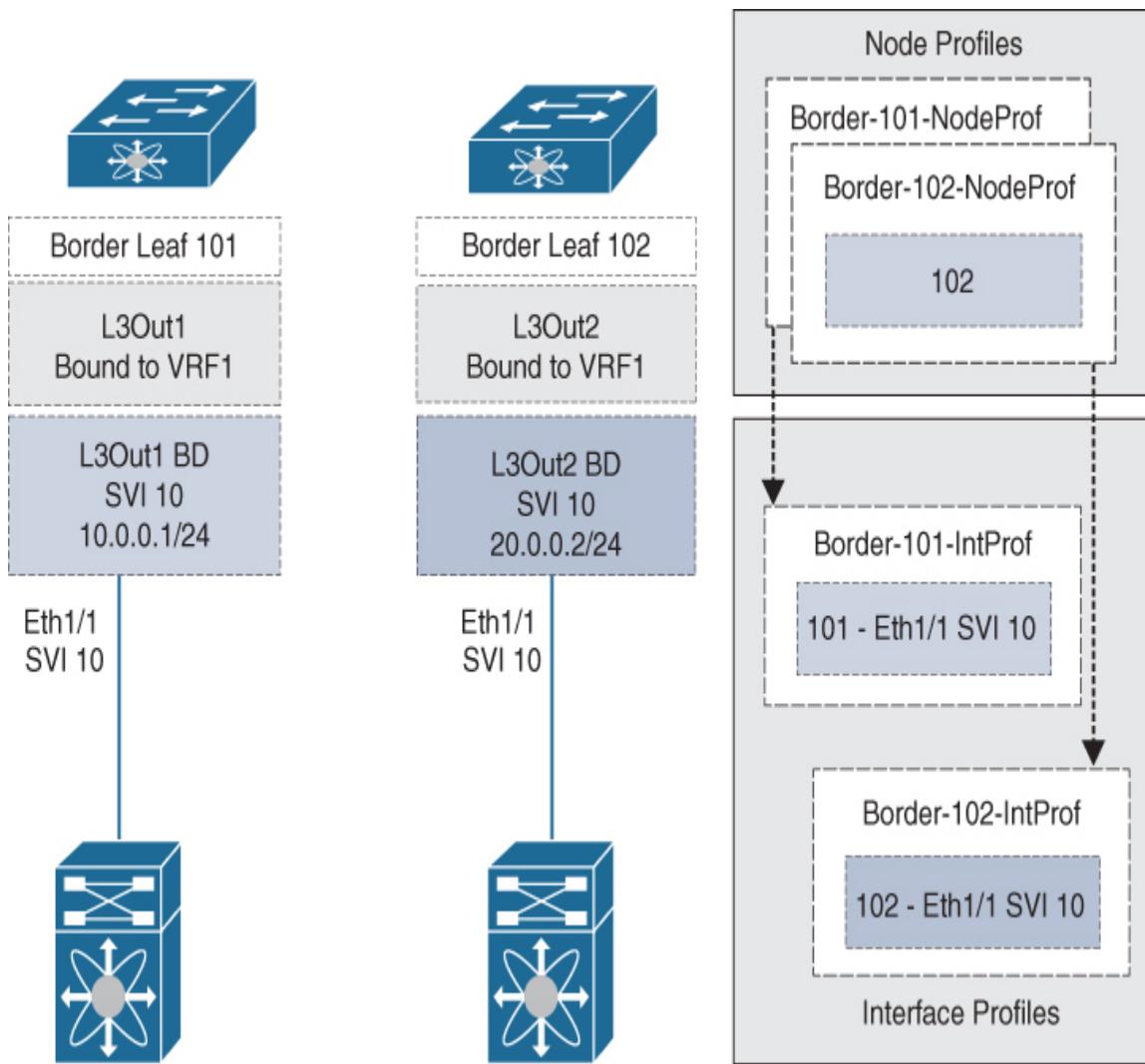
**Figure 9-5 Impact of L3Out BD Flooding Domains on Neighbor Relationships**

While this full mesh of adjacencies is generally a good thing, in some cases it can lead to ACI unintentionally transiting traffic between external routers. This is especially true when the routers connecting to ACI each have different routes in their routing tables. Because these routers are able to establish routing adjacencies with one another through the L3Out BD Layer 2 flooding domain, they are then able to use ACI as a transit to forward data plane traffic to each other.

But what happens if an administrator tries to instantiate a common SVI encapsulation in multiple L3Outs? It depends. If the administrator is attempting to deploy the SVI encapsulation to different ports across different L3Outs on the same border leaf, ACI allows use of the encapsulation for one L3Out but generates a fault when the second instance of the encapsulation is used, indicating that the encapsulation is already in use. Because of this, multiple L3Outs that need to use the same encapsulation cannot coexist on the same border leaf. However, this behavior can be changed with the SVI Encap Scope option under the L3Out SVI.



If, on the other hand, an administrator attempts to reuse an encapsulation in a new L3Out and on a different border leaf switch, ACI accepts the configuration and deploys a new L3Out BD for the second L3Out. This is the case in [Figure 9-6](#). The assumption, of course, is that the two L3Out BDs are intended to represent different subnets if deployed in the same VRF.



**Figure 9-6** Common SVI Encapsulation Used for L3Outs on Different Switches

## Understanding SVI Encap Scope

When configuring L3Outs using SVIs, one important setting is the Encap Scope. The acceptable values for this setting are VRF and Local.

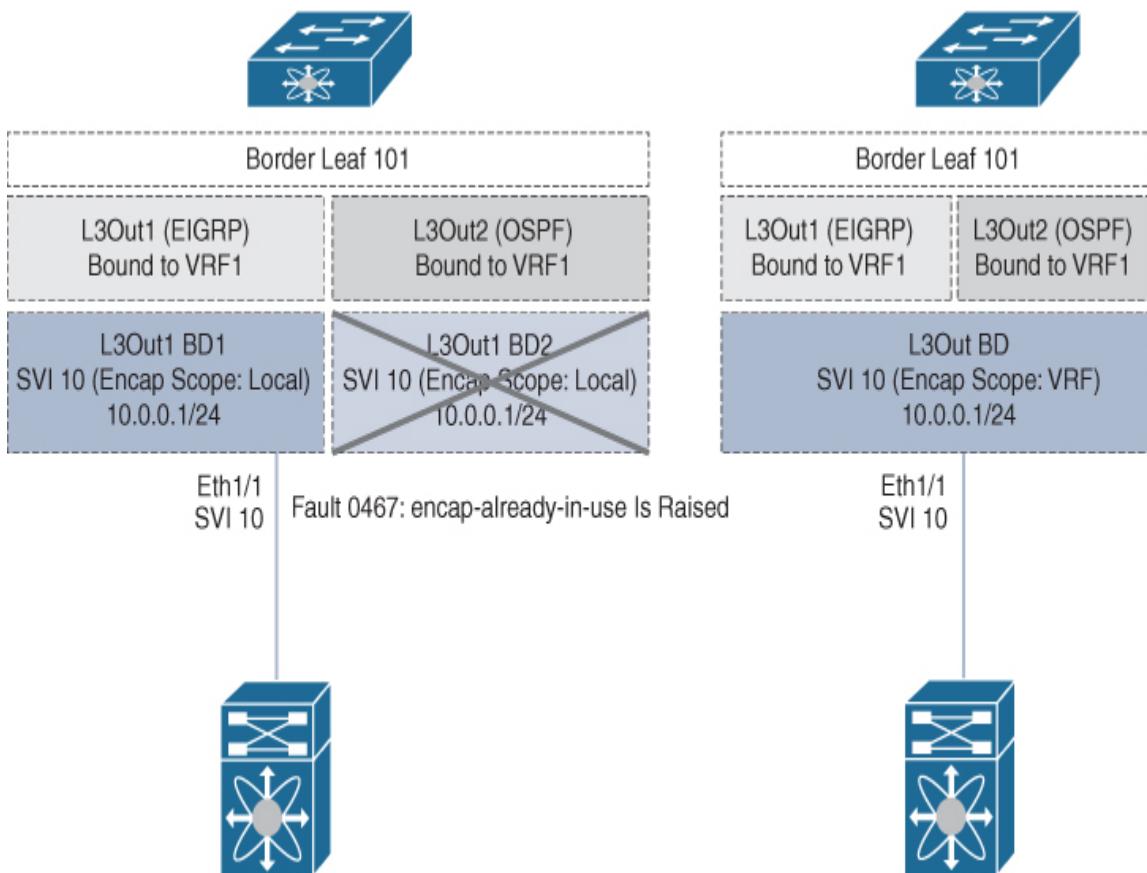
The only reason to modify this setting from its default value of Local is to be able to reuse an SVI in other L3Outs within a VRF. There are two specific use cases for this:

- Establishing adjacencies using multiple routing protocols from a single leaf using a common SVI
- Establishing granular route control over each BGP peer on the same leaf by using a dedicated L3Out for each BGP peer



[Figure 9-7](#) compares the two values for this setting. Let's say that an engineer needed to run both OSPF and EIGRP to an external device. In ACI, each L3Out can be configured for only one routing protocol; therefore, two L3Outs are needed to fulfill this requirement. The engineer determines that it would not be feasible to run multiple physical connections between ACI and the external device and that routed subinterfaces cannot be used in this design. In this case, an SVI encapsulation and IP address on a border leaf needs to be shared between the two L3Outs. If Encap Scope is set to Local for the SVI, ACI expects a unique external SVI for each L3Out and generates a fault when an L3Out SVI encapsulation is reused for a secondary L3Out on the switch. On the other hand, setting Encap Scope to VRF for an SVI tells ACI to expect a unique SVI encapsulation for each VRF and the two L3Outs are therefore allowed to share the encapsulation.





**Figure 9-7** Using SVI Encap Scope to Deploy an L3Out BD Across Multiple L3Outs

There is one exception to the aforementioned rule that each L3Out enable only a single routing protocol: OSPF can be enabled on a BGP L3Out to provide IGP reachability for BGP. When OSPF is enabled in the same L3Out as BGP, OSPF is programmed to only advertise logical node profile loopback addresses and interface subnets.

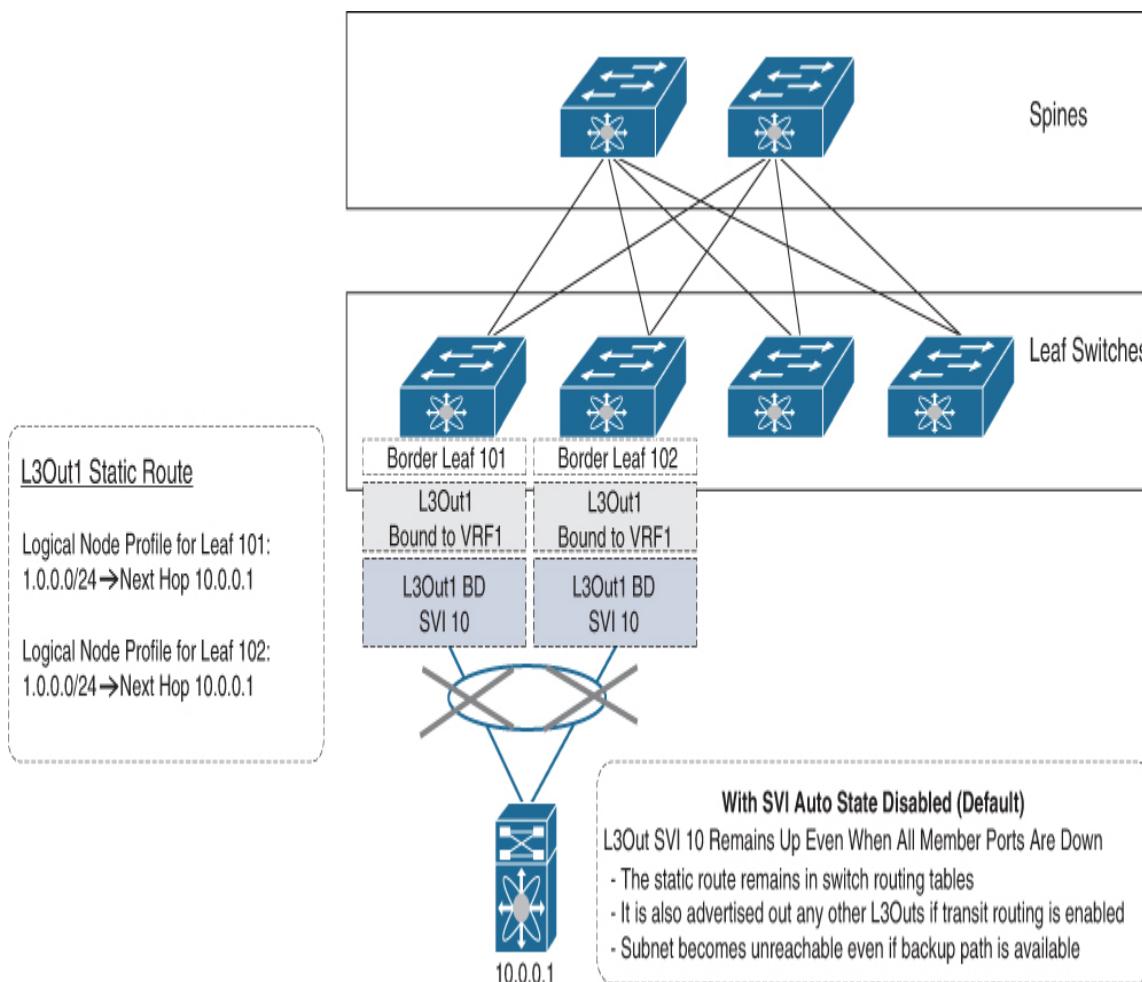
## Understanding SVI Auto State



Under default configurations, SVIs on an ACI leaf are always up, even if associated physical ports are down. Although this is typically not a problem, it could pose a problem when using static routes.

Figure 9-8 illustrates that a static route pointing out an L3Out SVI will remain in the routing tables of the border leaf switches and will continue to be distributed to other switches in the fabric if SVI Auto State is set at its default value, Disabled.

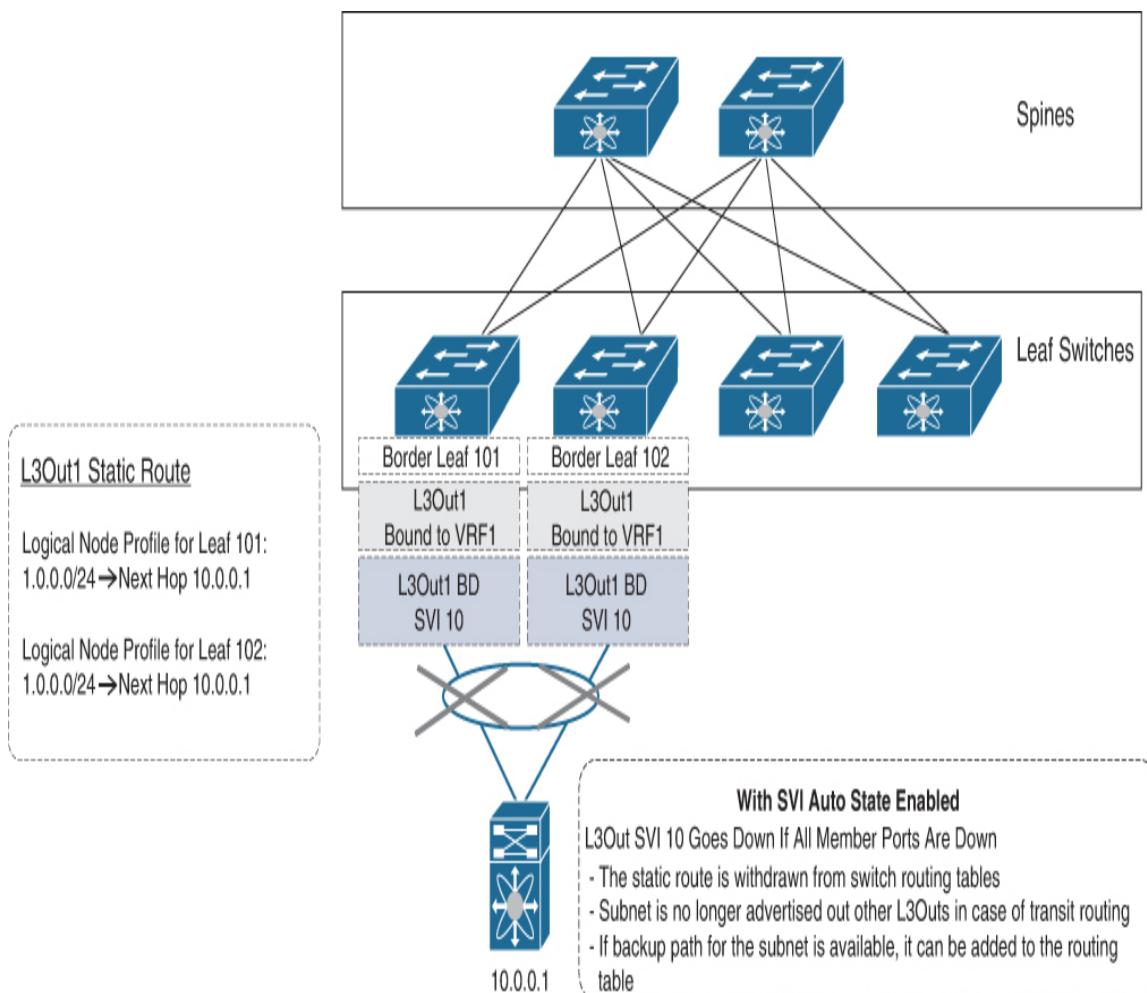
**Key Topic**



**Figure 9-8 Static Route Not Withdrawn on Downlink Failure with Auto State Disabled**

Figure 9-9 shows what happens to the static route when SVI Auto State is toggled to Enabled. In this case, the border leaf disables the L3Out SVI once it detects that all member ports for the L3Out SVI have gone down. This, in turn, prompts the static route to be withdrawn from the routing table and halts distribution of the static route to the rest of the fabric.

**Key Topic**



**Figure 9-9 Static Route Withdrawn on Downlink Failure with Auto State Set to Enabled**

Note that the implementation of BFD for the static route could also resolve this problem.

## Understanding Prerequisites for Deployment of L3Outs

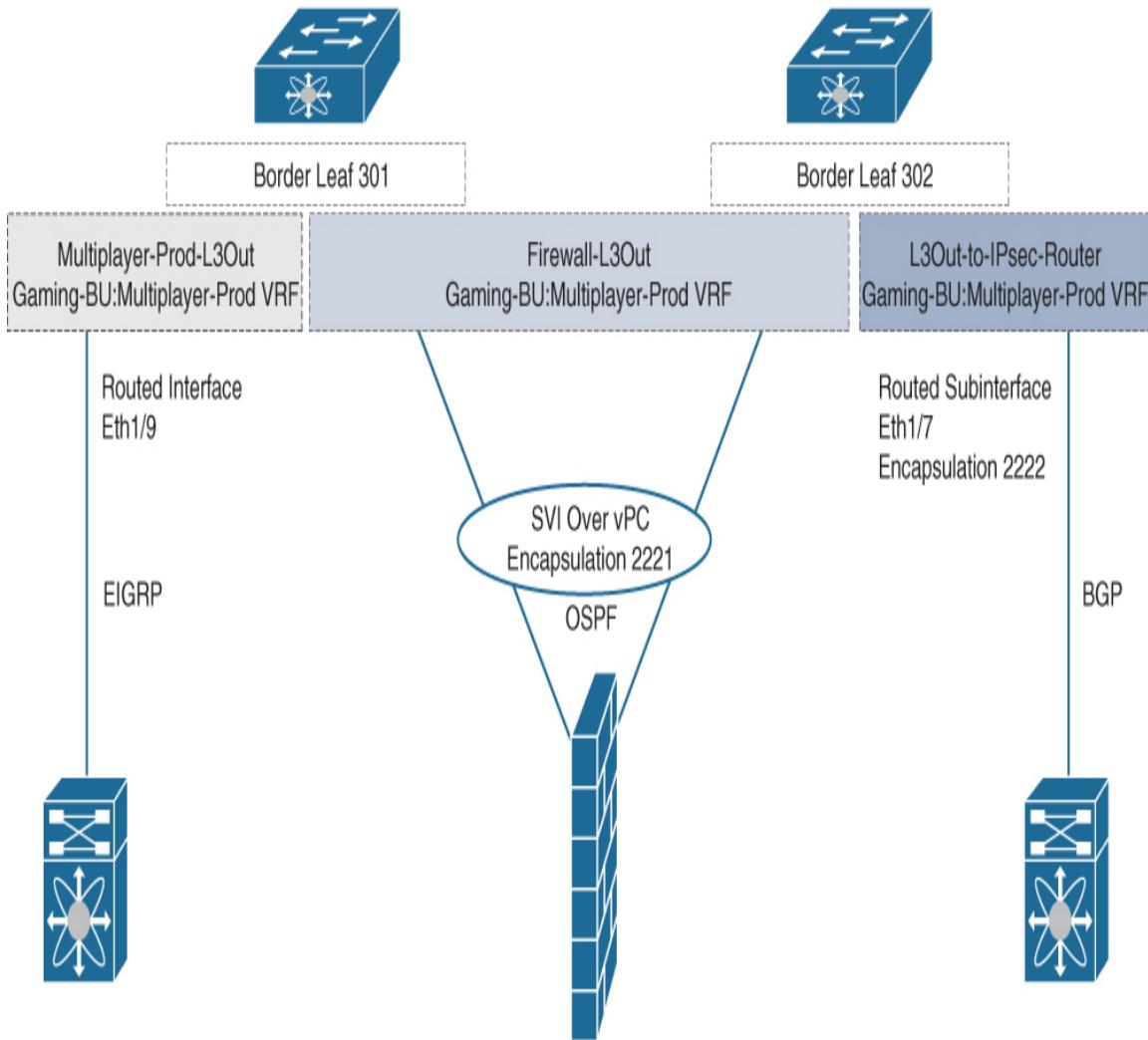
Before going into a tenant and creating an L3Out, access policies should be implemented for any physical switch ports that will be part of the L3Out. This includes assignment of an AAEP to an interface policy group and assignment of the interface policy group to an interface profile. The interface profile needs to be bound to the intended border leaf node IDs for the access policies to work. If access policies for the underlying physical ports are not in place to begin with, Layer 3 adjacencies will never be established.

Let's say that you have created several L3 domains and have already associated a VLAN pool with each of them. It is also important to ensure that an L3 domain is associated with the AAEP controlling connectivity to the underlying physical ports. It is the L3 domain that an L3Out references. If an L3 domain has not been associated with the AAEP that governs connectivity for the intended L3Out ports, you may be unable to deploy the L3Out.

## L3 Domain Implementation Examples

Say that the business unit (BU) from earlier chapters wants to deploy three L3Outs in its tenant, as shown in [Figure 9-10](#). One of the L3Outs uses EIGRP and connects to a dedicated switch on the floor where most of the BU

employees work. An OSPF connection to a firewall and a BGP connection to a router toward a partner network are also needed.



**Figure 9-10** Hypothetical L3Out Connectivity Desired for a VRF

Before deploying these L3Outs, the L3 domain objects would need to be configured. [Figure 9-11](#) shows how an L3 domain for connectivity to the firewall might be configured. Because the firewall will peer with ACI using an SVI, a VLAN pool *does* need to be defined.

Create L3 Domain

Name:

Associated Attachable Entity Profile:

VLAN Pool:

Security Domains:

Select	Name	Description
<input type="checkbox"/>	Development	
<input type="checkbox"/>	Production	
<input type="checkbox"/>	Sec-Domain	
<input type="checkbox"/>	infra	

**Figure 9-11** Sample L3 Domain Configuration That Includes VLAN Pool Assignment

Figure 9-12 shows the configuration of an L3 domain for connectivity to a router via subinterfaces. Notice that a VLAN pool has *not* been assigned to the L3 domain. This is a valid configuration. Administrators do not need to assign VLAN pools to L3 domains that will be used solely for routed interfaces or routed subinterfaces.

Create L3 Domain

Name:

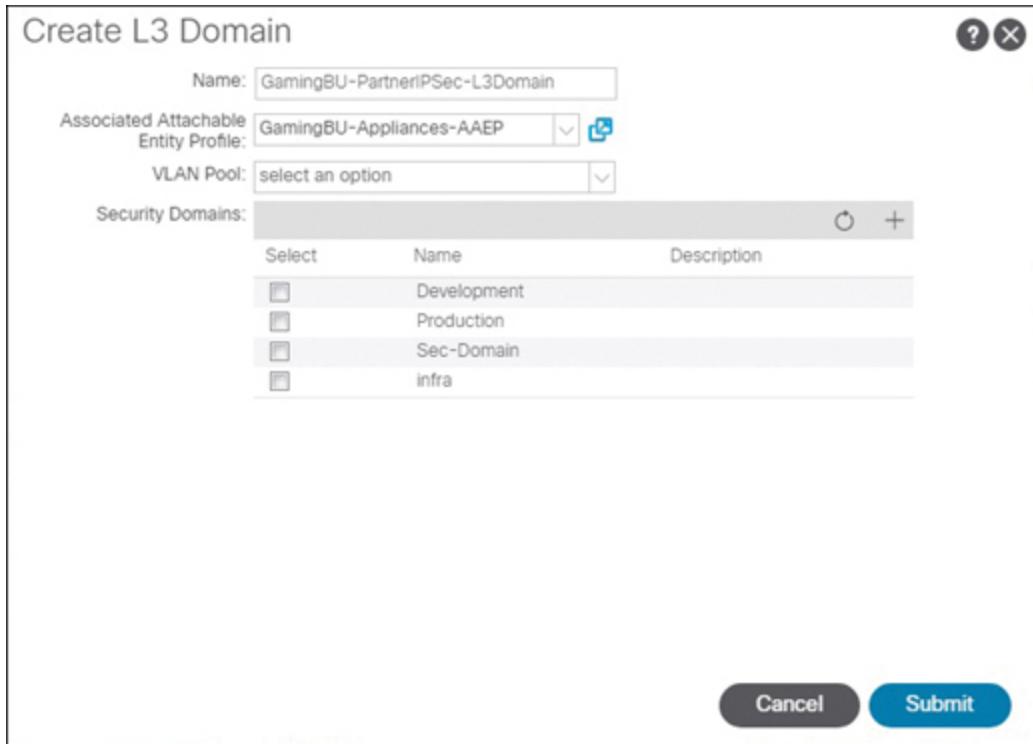
Associated Attachable Entity Profile:  

VLAN Pool:

Security Domains:

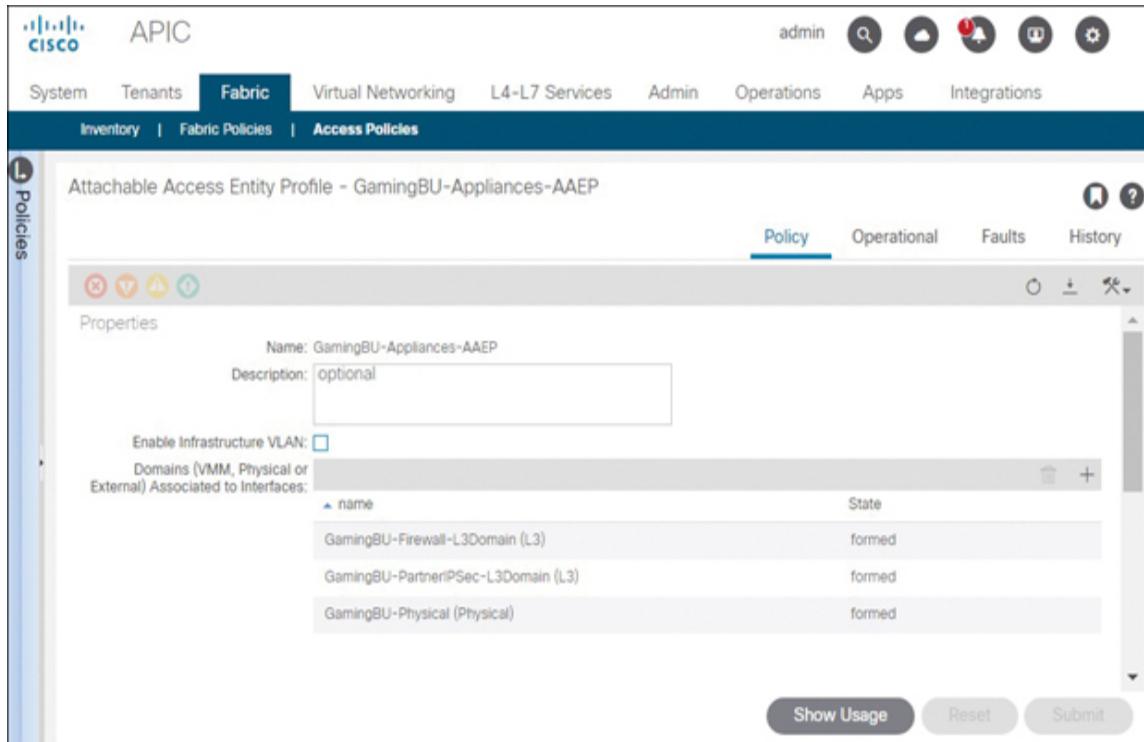
Select	Name	Description
<input type="checkbox"/>	Development	
<input type="checkbox"/>	Production	
<input type="checkbox"/>	Sec-Domain	
<input type="checkbox"/>	infra	

**Cancel** **Submit**



**Figure 9-12** *Creation of an L3 Domain Without VLAN Pool Assignment*

Finally, [Figure 9-13](#) shows the AAEP for connectivity to all appliances owned by the gaming BU. There was no need for the AAEP to be specific to the gaming BU systems since AAEP objects are designed with multitenancy in mind. Note here that a physical domain has also been associated with this AAEP. This makes sense if there is a need for the firewall to simply trunk some VLANs down to ACI using its port channel.



**Figure 9-13** *Associating Physical Domains and L3 Domains Simultaneously with an AAEP*

## Understanding the Need for BGP Route Reflection

To distribute external routes across the fabric, ACI leaf and spine switches establish iBGP peerings with one another within a user-defined BGP autonomous system number (ASN).

Whenever iBGP peerings are involved, BGP split-horizon rules apply. These rules state that a BGP router that receives a BGP route from an iBGP peer shall not advertise that route to another router that is an iBGP peer.

While BGP split horizon works wonders in preventing intra-ASN route loops, it requires that all routers in a BGP ASN form a full mesh to ensure that all routes propagate

correctly between all iBGP peers. This can impact scalability given that a full-mesh design involves  $N(N - 1)/2$  unique iBGP sessions, where  $N$  is the number of routers. Imagine an ACI fabric with 50 switches, all having to form BGP relationships with one another. The fabric would need to manage  $50(49)/2 = 1225$  BGP sessions to make this possible. Full-mesh iBGP is not scalable and therefore is not used in ACI fabrics.

The scalable alternative to an iBGP full mesh is the deployment of **route reflectors**. A route reflector is a BGP speaker that is allowed to advertise iBGP-learned routes to certain iBGP peers. Route reflection bends the rules of BGP split horizon just a little by introducing a new set of BGP attributes for route loop prevention. In modern Clos fabrics such as ACI, spines make ideal route reflectors because they often have direct connections with all leaf switches.

Using BGP route reflection, the number of unique iBGP peerings in a single-pod ACI fabric drops down to  $RR \times RRC$ , where  $RR$  is the number of route reflectors and  $RRC$  is the number of route reflector clients. In a hypothetical 50-switch single-pod ACI fabric consisting of 2 spines that have been configured as route reflectors and 48 leaf switches (route reflector clients), the sum total number of unique iBGP sessions needed stands at 96.

None of these points may be critical DCACI trivia. What should be important for DCACI 300-620 exam candidates is first and foremost to recognize the symptoms of forgetting to implement BGP route reflection within a fabric. Next, you need to know how to configure BGP route reflection in the first place.



If an ACI fabric has not been configured for BGP route reflection, border leaf switches can learn routes from external routers, but ACI does not distribute such routes to other nodes in the fabric. External routes therefore never appear in the routing tables of other leaf switches in the fabric.

## Implementing BGP Route Reflectors

Administrators usually configure BGP route reflection during fabric initialization or when they deploy the first L3Out in the fabric.

An administrator needs to do two things for BGP route reflection to work in ACI:

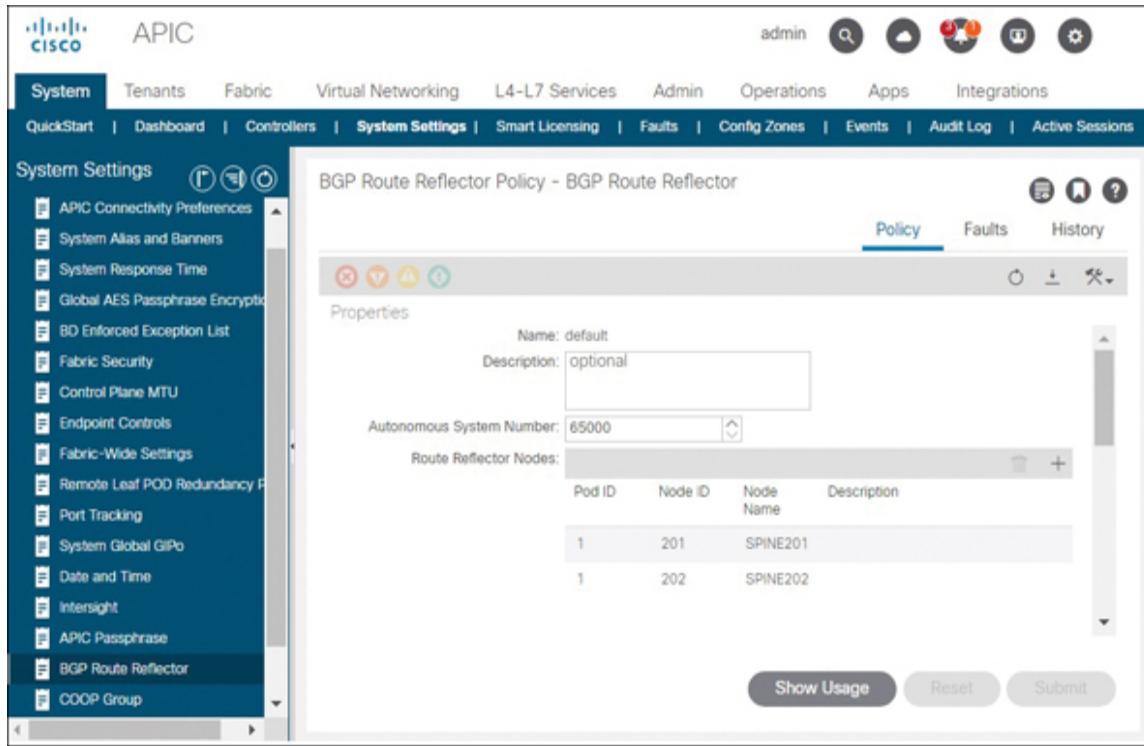


- Enter the BGP ASN the fabric should use internally.
- Select the spines that will function as route reflectors.

Not all spines need to be configured as route reflectors, but it usually makes sense to have at least two for redundancy.

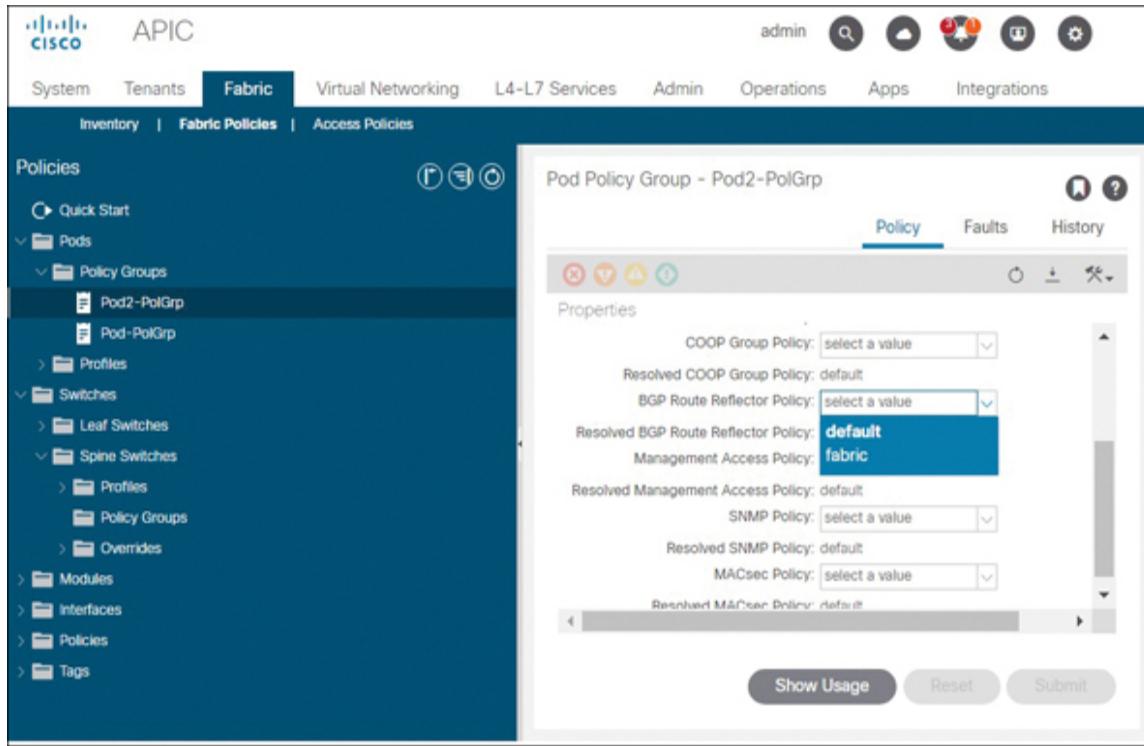
To configure route reflection in ACI, navigate to **System > System Settings > BGP Route Reflector**. [Figure 9-14](#) shows the selection of two spines with node IDs 201 and 202 as BGP route reflectors for Pod 1. The fabric uses BGP ASN 65000 in this example. Note that route reflectors for additional pods can also be configured on this page.





**Figure 9-14** Configuring BGP Route Reflectors Within a Fabric

You might be wondering whether you can deploy an alternative BGP route reflector policy. The answer is that you cannot do so in ACI Release 4.2. [Figure 9-15](#) indicates that even when creating a new pod policy group, ACI does not allow modification of the BGP route reflector policy named default shown earlier, in [Figure 9-14](#).



**Figure 9-15** Assigning a BGP Route Reflector Policy to a Pod Policy Group

## Understanding Infra MP-BGP Route Distribution

Once BGP route reflection has been configured, ACI deploys MP-BGP on all leaf and spine switches. The following steps summarize what takes place for external routes to propagate from border leaf switches to all other switches in the fabric:

**Step 1.** The BGP IPv4/IPv6 address family (AF) is deployed on all leaf switches (both border and non-border leaf switches) in all user VRF instances.

**Step 2.** The BGP VPNv4/VPNv6 AF is also deployed on all leaf and route reflector spine switches in the infra VRF (overlay-1 VRF). All leaf switches establish iBGP sessions with route reflector spine switches in

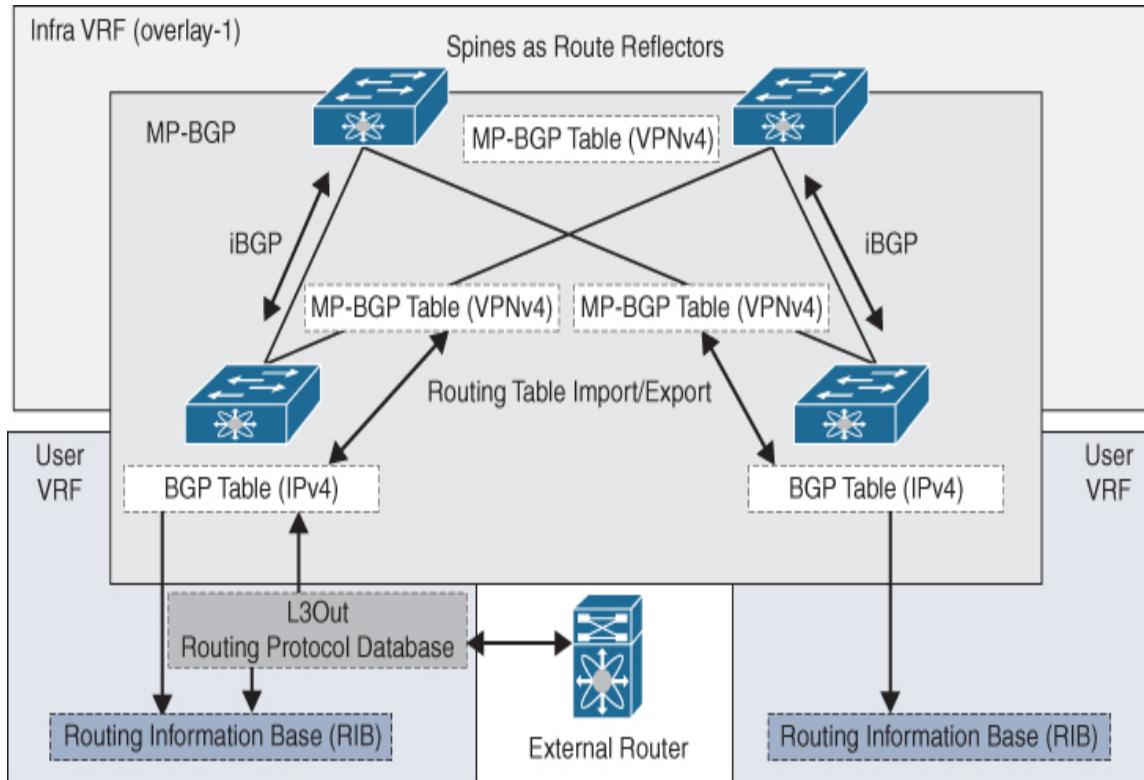
the infra VRF and are then able to exchange their VPNv4/VPNv6 routes.

**Step 3.** Once an L3Out is deployed on a leaf, the BGP IPv4/IPv6 AF on the same border leaf automatically creates a redistribution rule for all the routes from the routing protocol of the L3Out within the same user VRF. This redistribution is called **interleak**. If the L3Out is using BGP, no redistribution (interleak) is required for routes learned via BGP because the BGP process for the L3Out and for the infra MP-BGP is the same.

**Step 4.** The redistributed IPv4/IPv6 routes are exported from the user VRF to the infra VRF as VPNv4/VPNv6.

**Step 5.** On other leaf switches, the VPNv4/VPNv6 routes distributed through route reflector spines are imported from the infra VRF to the user VRF as IPv4/IPv6.

[Figure 9-16](#) recaps this route distribution process.



**Figure 9-16** Infra MP-BGP Architecture and Route Distribution

Those interested in learning more about the MP-BGP route distribution process can begin their exploration process using the commands **show bgp process detail vrf all**, **show bgp ipv4 unicast vrf all**, and **show bgp vpnv4 unicast vrf overlay-1**.

For those not interested in the route distribution process, perhaps the only thing of importance on this issue is to be able to verify spine-to-leaf BGP peerings. [Example 9-1](#) shows how to verify the number of BGP adjacencies on a leaf switch. It also demonstrates how you can correlate neighbor router IDs with their hostnames, node IDs, and TEP addresses.

**Example 9-1** Validating BGP Peerings Between a Leaf and Route Reflector Spines

[Click here to view code image](#)

```
LEAF101# show bgp sessions vrf overlay-1
Total peers 5, established peers 5
ASN 65000
VRF overlay-1, local ASN 65000
peers 2, established peers 2, local router-id 10.233.46.32
State: I-Idle, A-Active, 0-Open, E-Established, C-Closing, S-
Shutdown

Neighbor          ASN   Flaps LastUpDn|LastRead|LastWrit St
Port(L/R) Notif(S/R)
10.233.46.33      65000 0       16w06d |never    |never    E
60631/179 0/0
10.233.46.35      65000 0       16w06d |never    |never    E
44567/179 0/0
LEAF101# acidiag fnvread | grep spine
 201      1      SPINE201      FD0XXXX1
10.233.46.33/32  spine  active  0
 202      1      SPINE202      FD0XXXX2
10.233.46.35/32  spine  active  0
```

The external route distribution process in ACI is extremely reliable. If all expected BGP peerings appear to be in an *established* state, you should see external routes pop up in the routing table of all leaf switches where the relevant user VRF has been deployed.

## Deploying L3Outs

ACI Release 4.2 has a streamlined wizard for setting up L3Outs. While use of the wizard is required when configuring an L3Out via the GUI, administrators can make modifications to L3Outs afterward.

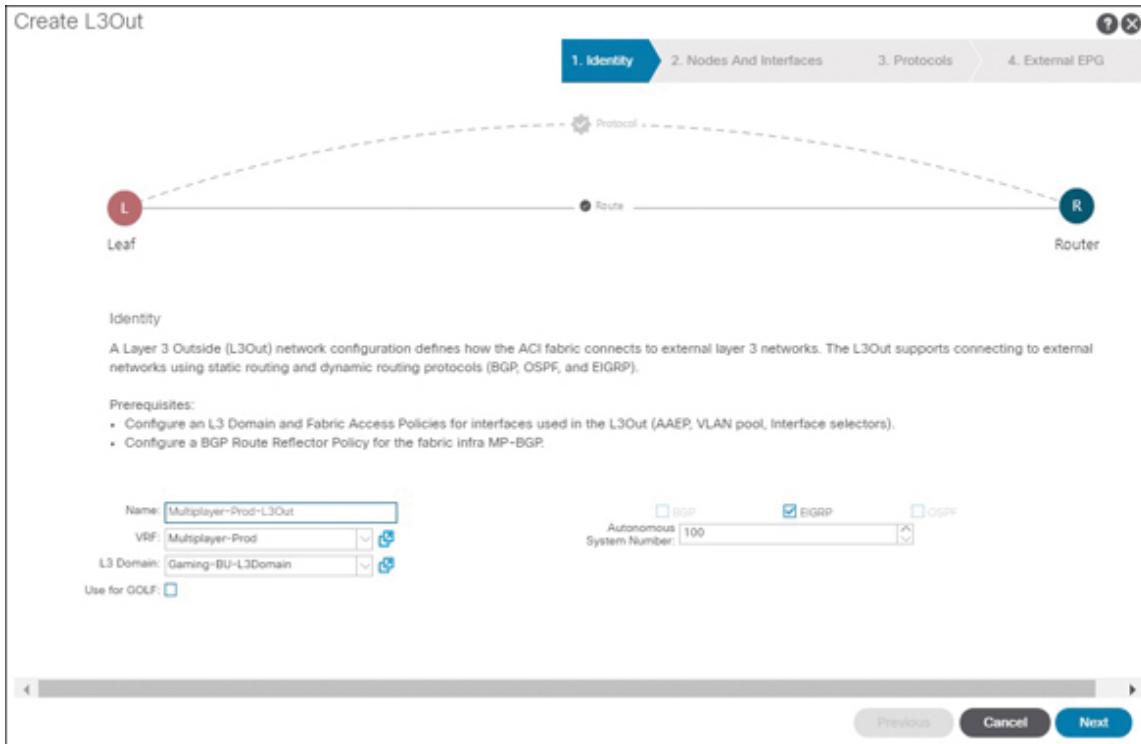
This section demonstrates the process of going through the streamlined wizard to create L3Outs for each routing protocol and interface type. It also touches on deployment of external EPGs on L3Outs and some configuration changes you might want to make manually.

## Configuring an L3Out for EIGRP Peering

To launch the L3Out creation wizard, navigate to Tenants, select the tenant in question, open the Networking folder, right-click L3Outs, and select Create L3Out.

[Figure 9-17](#) shows the first page of the wizard. Following completion of the wizard, the settings on this page all appear at the root of the L3Out on the Main subtab of the Policy page. Enter a name for the L3Out, select the VRF instance with which the L3Out should be associated, and select the L3 domain created for this individual L3Out. Then select the routing protocol. When EIGRP has been selected as the protocol of choice, the wizard disables the OSPF and BGP checkboxes, and the Autonomous System Number text box appears. Entry of an autonomous system number is mandatory. Click Next to continue.





**Figure 9-17** Entering the EIGRP and ASN Configuration in the L3Out Creation Wizard

### Note

If you are unfamiliar with domains in general, see [Chapter 6, “Access Policies.”](#) For coverage of VRF configuration, see [Chapter 5](#).

In [Figure 9-17](#), notice the checkbox Use for GOLF. Selecting the Use for GOLF checkbox on a user VRF L3Outs tells ACI that the L3Out will be a mere placeholder for external EPGs and other policies for a GOLF L3Out. There is no reason to select this option when the intent is to establish Layer 3 peerings for border leaf switches.

The next page in the wizard pertains to logical node profiles and logical interface profiles. The wizard selects a node profile name by default. Deselect Use Defaults, as shown in

**Figure 9-18**, if you need to customize the logical node profile name. Next, select an interface type. This example shows a routed interface selected with the port option in line with the L3Out designs presented in **Figure 9-10**, earlier in this chapter. Select a node ID and enter a router ID for this border leaf switch. Do not use CIDR notation for router IDs. Entering a loopback address is required only if you expect to implement BGP Multihop by sourcing a loopback address. Finally, enter information for the interfaces on the border leaf that need to be enabled for EIGRP. CIDR notation for these interfaces is required. Click Next to move on to the next page.



Create L3Out

1. Identity    2. Nodes And Interfaces    3. Protocols    4. External EPG

Nodes and Interfaces

The L3Out configuration consists of node profiles and interface profiles. An L3Out can span across multiple nodes in the fabric. All nodes used by the L3Out can be included in a single node profile and is required for nodes that are part of a VPC pair. Interface profiles can include multiple interfaces. When configuring dual stack interfaces a separate interface profile is required for the IPv4 and IPv6 configuration, that is automatically taken care of by this wizard.

Use Defaults:  Node Profile Name:

Interface Types

Layer 3	Routed	Routed Sub	SVI	Floating SVI
Layer 2	Port	Direct Port Channel		

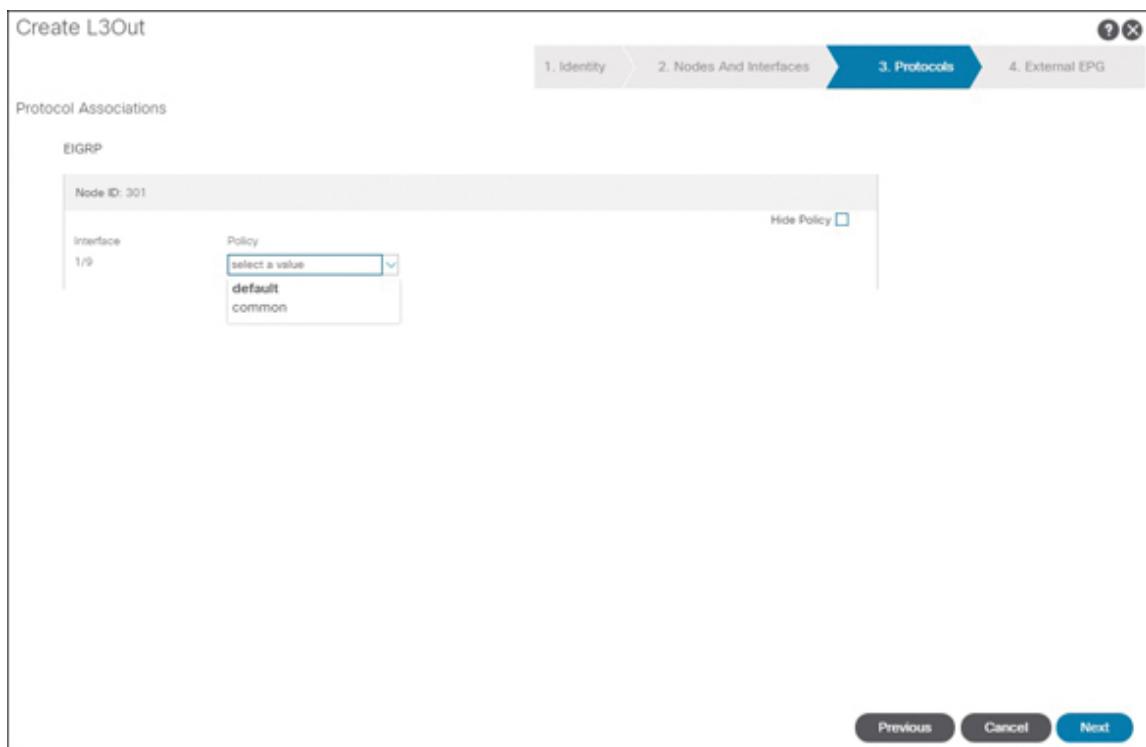
Nodes

Node ID	Router ID	Loopback Address	
LEAF301 (Node-301)	10.233.75.169	<input type="text"/> Leave empty to not configure any Loopback	
Interface	IP Address	Interface Profile Name	MTU (bytes)
eth1/9	10.233.75.162/30 Address/mask	Multiplayer-Prod-L3Out_k	9000

Previous    Cancel    **Next**

**Figure 9-18** Entering Node and Interface Information for an EIGRP L3Out

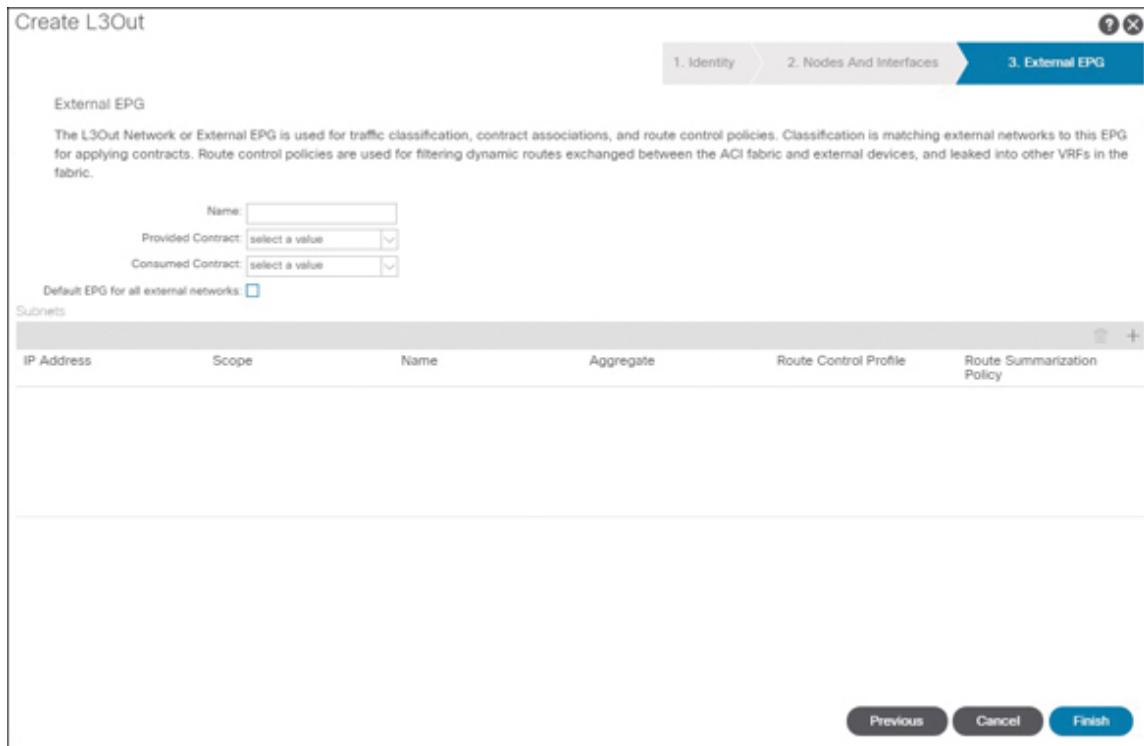
The Protocol Associations page, shown in [Figure 9-19](#), allows association of a custom EIGRP interface policy with the EIGRP interface profile for this L3Out. By default, ACI selects an EIGRP interface policy called default from the common tenant. New protocol interface policies cannot currently be created in the L3Out creation wizard. Custom EIGRP interface policies are discussed and applied to L3Outs later in this chapter. Click Next to continue.



**Figure 9-19** *Associating an EIGRP Interface Policy with the EIGRP Interface Profile*

[Figure 9-20](#) shows the final page of the L3Out creation wizard. This page allows you to define external EPGs for the L3Out. When the Default EPG for All External Networks checkbox is enabled, ACI automatically generates an EPG that matches all traffic not matched by a more specific external EPG. Disable this checkbox if you want to manually create external EPGs after the L3Out has been deployed. Click Finish to deploy the L3Out.

## Key Topic



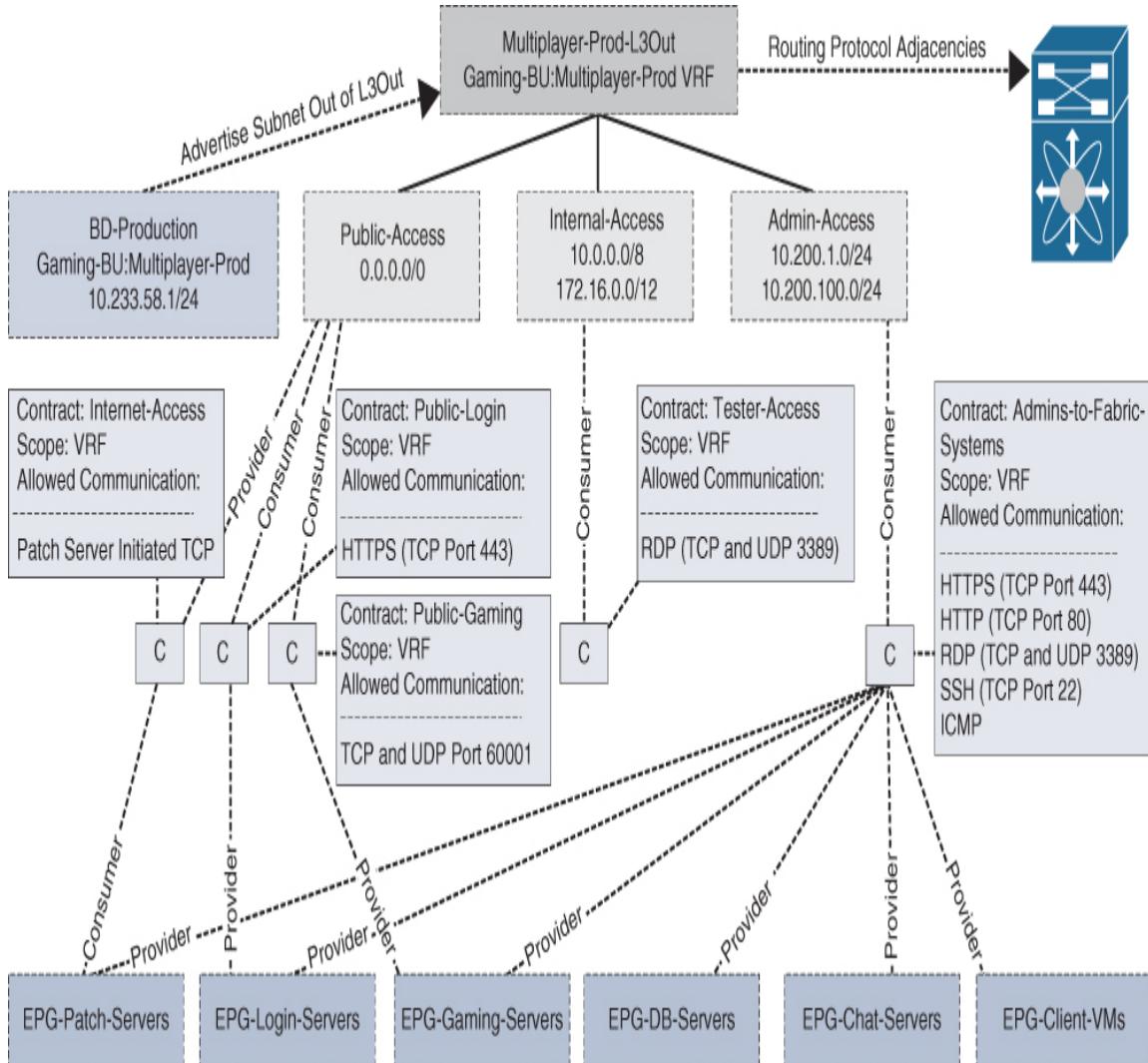
**Figure 9-20** External EPG Creation Page in the L3Out Creation Wizard

Once the L3Out is deployed, if you check to see whether EIGRP adjacencies have been established, you will find that they have not. ACI does not attempt to establish route peerings out an L3Out until at least one external EPG has been deployed on the L3Out.

## Deploying External EPGs

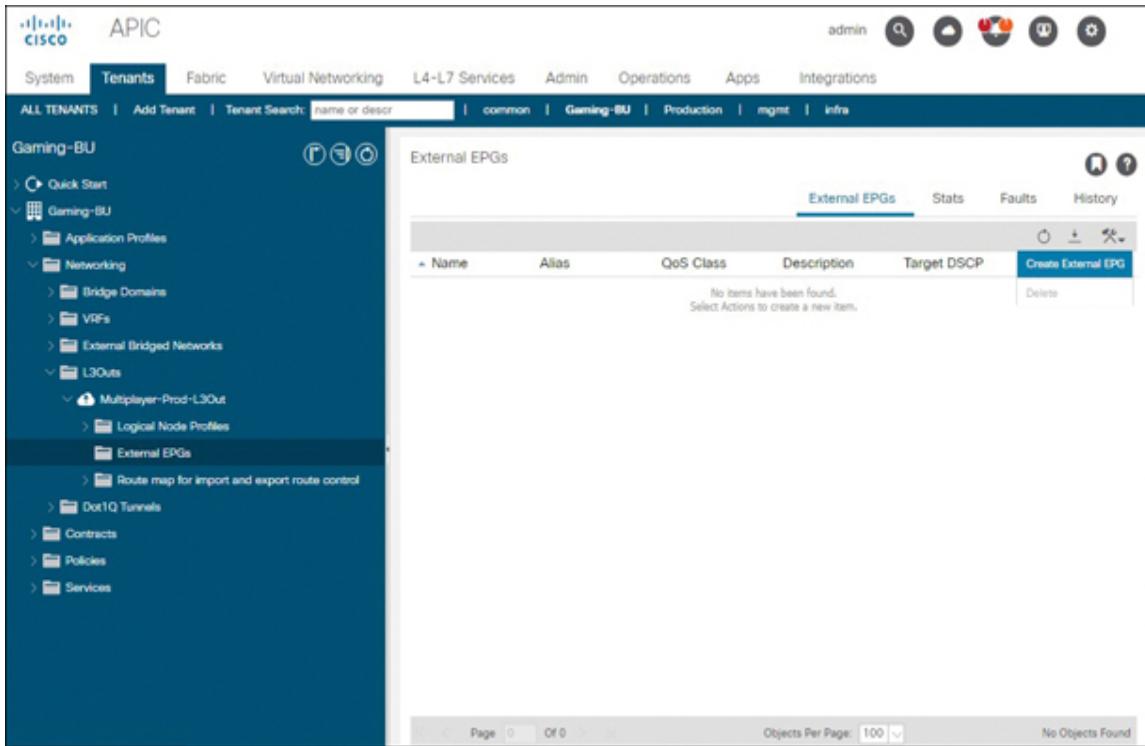
Chapter 5 describes the significance of external EPGs in classifying traffic behind L3Outs, and Chapter 8, “Implementing Tenant Policies,” covers contract implementation. These topics are not repeated here, but Figure 9-21 outlines hypothetical connectivity that needs to

be whitelisted between external endpoints and internal EPGs. In this figure, Public-Access, Internal-Access, and Admin-Access are external EPGs.



**Figure 9-21** External EPG Design in a Specific VRF

To create an external EPG on an L3Out, drill down into the subfolders of the L3Out and select the External EPG folder. Then right-click the Tools menu and select Create External EPG (see [Figure 9-22](#)).



**Figure 9-22** Kicking Off the External EPG Creation Wizard

Enter a name for the external EPG. [Chapter 10, “Extending Layer 2 Outside ACI,”](#) briefly touches on the idea of the preferred group member. When contracts are used to allow communication, the Preferred Group Member parameter can be left at the default value, Exclude. To define subnets for the external EPG, click the + sign in the Subnet part of the screen shown in [Figure 9-23](#).

The screenshot shows the 'Create External EPG' wizard. At the top, there are buttons for Help (?) and Close (X). The main area contains the following fields:

- Name:** Admin-Access
- Alias:** (empty input field)
- Tags:** (dropdown menu) enter tags separated by comma
- Contract Exception Tag:** (empty input field)
- QoS Class:** Unspecified (dropdown menu)
- Description:** optional (text input field)
- Target DSCP:** Unspecified (dropdown menu)
- Preferred Group Member:** (button) Exclude (selected) / Include

Below these fields is a section titled 'Subnet' with a table:

IP Address	Scope	Name	Aggregate	Route Control Profile	Route Summarization Policy

At the bottom right are 'Cancel' and 'Submit' buttons.

**Figure 9-23** Main Page of the Create External EPG Wizard

Note that this figure shows configuration of the Admin-Access external EPG. Two subnets have been called out for association with this particular external EPG: 10.200.1.0/24 and 10.200.100.0/24.



Figure 9-24 shows the addition of the subnet 10.200.100.0/24 to Admin-Access. The Name field on this page reflects the function of the particular subnet being added. Several groups of checkboxes exist on this page. These checkboxes are called *Scope* options, and they determine the function(s) of each external EPG. The checkboxes in the Route Control section predominantly relate to transit routing scenarios. The Shared Security

Import Subnet checkbox relates to shared service L3Outs. The only checkbox of interest for DCACI candidates is External Subnets for External EPG. When enabled, this checkbox tells ACI that this external EPG should be used to classify external traffic matching the subnet for contract enforcement. Enable this checkbox and click Submit.

The screenshot shows the 'Create Subnet' dialog box. At the top, there are fields for 'IP Address' (10.200.100.0/24) and 'Name' (Network-Admin). Below these are sections for 'Route Control' and 'Aggregate'. Under Route Control, there are checkboxes for 'Export Route Control Subnet' (unchecked), 'Import Route Control Subnet' (unchecked), and 'Shared Route Control Subnet' (unchecked). Under Aggregate, there are checkboxes for 'Aggregate Export' (unchecked), 'Aggregate Import' (unchecked), and 'Aggregate Shared Routes' (unchecked). A 'OSPF Route Summarization Policy' section is also present. In the center, there is a 'Route Control Profile' table with columns 'Name' and 'Direction'. At the bottom, a note states: 'Route control is used for filtering external routes advertised out of the fabric, allowed into the fabric, or leaked to other VRFs within the fabric.' Below this is an 'External EPG classification' section with a checked checkbox for 'External Subnets for External EPG'. A note below it says: 'External EPG classification is used to identify the external networks associated with this external EPG for policy enforcement (Contracts)'. At the bottom right are 'Cancel' and 'Submit' buttons.

**Figure 9-24** Adding Subnets as External Subnets for an External EPG

After you add a subnet to an external EPG, the Create External EPG page reappears so you can assign additional subnets to the external EPG. When you finish this, click Submit. [Figure 9-25](#) shows the General tab for an external EPG that has been created. Notice in this figure that the external EPGs Internal-Access and Public-Access have been created in the background. This view is particularly useful because it verifies that the external EPG has been deployed and also that all intended subnets have been assigned to it, using the proper scope.

## Key Topic

The screenshot shows the Cisco Application Policy Infrastructure Controller (APIC) web interface. The top navigation bar includes links for System, Tenants, Fabric, Virtual Networking, L4-L7 Services, Admin, Operations, Apps, and Integrations. The Tenant navigation bar shows 'ALL TENANTS' and 'Add Tenant'. The main content area is titled 'External EPG Instance Profile - Admin-Access'. It has tabs for Policy, Operational, Stats, Health, Faults, and History, with 'General' selected. Below the tabs is a status bar with icons for healthy, warning, and error. A 'Properties' section lists 'Configuration Issues' and 'Preferred Group Member' (Exclude). The 'Subnets' table lists two entries:

IP Address	Scope	Name	Aggregate Route Control Profile	Route Summarization Policy
10.200.1.0/24	External Subnets for the External EPG	VM-Admins		
10.200.100.0/24	External Subnets for the External EPG	Network-Admins		

At the bottom are buttons for Show Usage, Reset, and Submit.

**Figure 9-25** The General Tab of an External EPG

Now that at least one external EPG has been deployed on the L3Out, ACI should attempt to establish routing adjacencies with external routers.

## Verifying Forwarding Out an L3Out

Example 9-2 shows how to verify that an EIGRP adjacency has been established in a VRF.

**Example 9-2** Verifying EIGRP Adjacencies Resulting from L3Out Configuration

[Click here to view code image](#)

```

LEAF301# show ip eigrp neighbors vrf Gaming-BU:Multiplayer-Prod
EIGRP neighbors for process 100 VRF Gaming-BU:Multiplayer-Prod
      H   Address           Interface      Hold   Uptime
      SRTT    RT0    Q    Seq
                                         (sec)
      (ms)          Cnt Num
      0   10.233.75.161        eth1/9       13   07:30:37
      1      50    0    19

```

## Note

Keep in mind that you have not seen any switch ports configured in this chapter. L3Out adjacencies cannot form unless access policies for underlying switch ports have been configured.

Following adjacency verification, it makes sense to confirm whether ACI has learned any routes. [Example 9-3](#) shows the routing table of node 301 and node 302. Notice the highlighted routes learned by node ID 301, where the L3Out has been deployed. These all appear to have been learned via EIGRP. This is expected because the L3Out runs EIGRP. Node 302, on the other hand, has learned these same routes from bgp-65000. This is because node 302 learned these routes from MP-BGP route distribution, and route reflectors in this particular fabric have been configured with BGP ASN 65000. Notice that these route entries on node 302 all have next-hop addresses pointing to the TEP address of node 301. This is expected behavior.

### **Example 9-3 Verifying the Routing Table for a Specific VRF**

[Click here to view code image](#)

```

LEAF301# show ip route vrf Gaming-BU:Multiplayer-Prod
IP Route Table for VRF "Gaming-BU:Multiplayer-Prod"
(...output truncated for brevity...)
10.199.90.0/24, ubest/mbest: 1/0
    *via 10.233.75.161, eth1/9, [90/128576], 22:35:22, eigrp-
        default, internal
10.200.1.0/24, ubest/mbest: 1/0
    *via 10.233.75.161, eth1/9, [90/128576], 22:43:09, eigrp-
        default, internal
10.200.100.0/24, ubest/mbest: 1/0
    *via 10.233.75.161, eth1/9, [90/128576], 23:01:31, eigrp-
        default, internal

LEAF302# show ip route vrf Gaming-BU:Multiplayer-Prod
IP Route Table for VRF "Gaming-BU:Multiplayer-Prod"
(...output truncated for brevity...)
10.199.90.0/24, ubest/mbest: 1/0
    *via 10.233.60.234%overlay-1, [200/128576], 22:43:13,
        bgp-65000, internal, tag 65000
10.200.1.0/24, ubest/mbest: 1/0
    *via 10.233.60.234%overlay-1, [200/128576], 22:51:01,
        bgp-65000, internal, tag 65000
10.200.100.0/24, ubest/mbest: 1/0
    *via 10.233.60.234%overlay-1, [200/128576], 23:09:23,
        bgp-65000, internal, tag 65000

```

Finally, recall from [Chapter 8](#) that ACI does not store information about external endpoints learned via an L3Out in endpoint tables. ARP is used to keep track of next-hop MAC-to-IP address bindings for endpoints behind L3Outs. [Example 9-4](#) shows the ARP table for the VRF from the perspective of border leaf 301.

## **Example 9-4 Checking the ARP Table for Next-Hop MAC-to-IP Address Binding**

[Click here to view code image](#)

```
LEAF301# show ip arp vrf Gaming-BU:Multiplayer-Prod

Flags: * - Adjacencies learnt on non-active FHRP router
      + - Adjacencies synced via CFSoE
      # - Adjacencies Throttled for Glean
      D - Static Adjacencies attached to down interface

IP ARP Table for context Gaming-BU:Multiplayer-Prod
Total number of entries: 1
Address          Age       MAC Address        Interface
10.233.75.161   00:02:30  a0e0.af66.c5a1  eth1/9
```

Unless data in one of these tables is inaccurate, ACI should be able to forward traffic to external devices without issue.

## **Advertising Subnets Assigned to Bridge Domains via an L3Out**

When ACI is learning subnets behind an L3Out, it is time to advertise ACI subnets out of the fabric. The most basic and yet common form of bridge domain subnet advertisement involves a two-step process. First, navigate to the desired BD and add one or more L3Out in the Associated L3Outs view. Then, drill down into an individual subnet that needs to be advertised and update its scope to Advertised Externally. These two configurations do not need to be done in the order specified, but together, they tell ACI to internally create a route map rule on the border leaf switches to redistribute the desired BD subnets into the routing protocol of the associated L3Out.

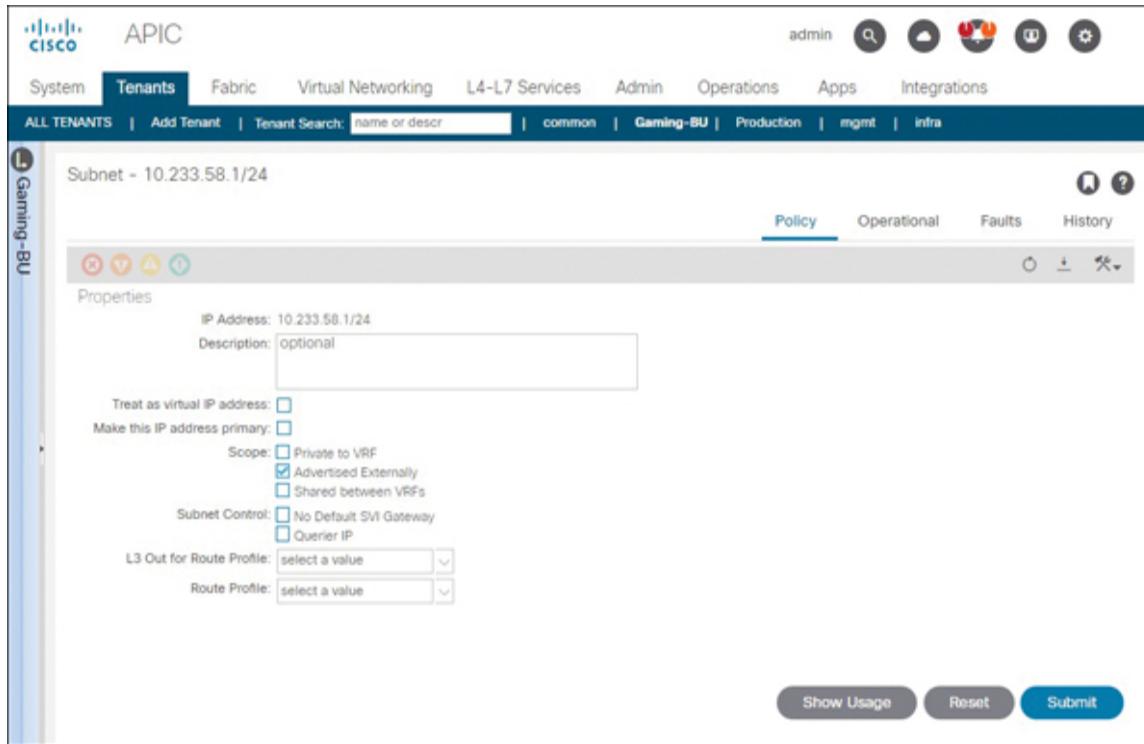
**Figure 9-26** shows the addition of a bridge domain named BD-Production to the L3Out named Multiplayer-Prod-L3Out.

The screenshot shows the Cisco Application Policy Infrastructure Controller (APIC) web interface. The top navigation bar includes tabs for System, Tenants, Fabric, Virtual Networking, L4-L7 Services, Admin, Operations, Apps, and Integrations. The Tenant navigation bar shows 'Gaming-BU' as the selected tenant. The main content area is titled 'Bridge Domain - BD-Production'. It displays tabs for Summary, Policy, Operational, Stats, Health, Faults, and History, with 'Policy' being the active tab. Under the 'L3 Configurations' sub-tab, there is a section for 'Associated L3 Outs'. A dropdown menu under 'Associated L3 Outs' shows 'L3 Out' and 'Multiplayer-Prod-L3Out', where 'Multiplayer-Prod-L3Out' is highlighted with a blue selection bar. Below this, there are two dropdown menus: 'L3 Out for Route Profile:' and 'Route Profile:', both currently set to 'select a value'. At the bottom right are buttons for 'Show Usage', 'Reset', and 'Submit'.

**Figure 9-26** *Marking a BD as a Candidate for Subnet Redistribution into an L3Out*

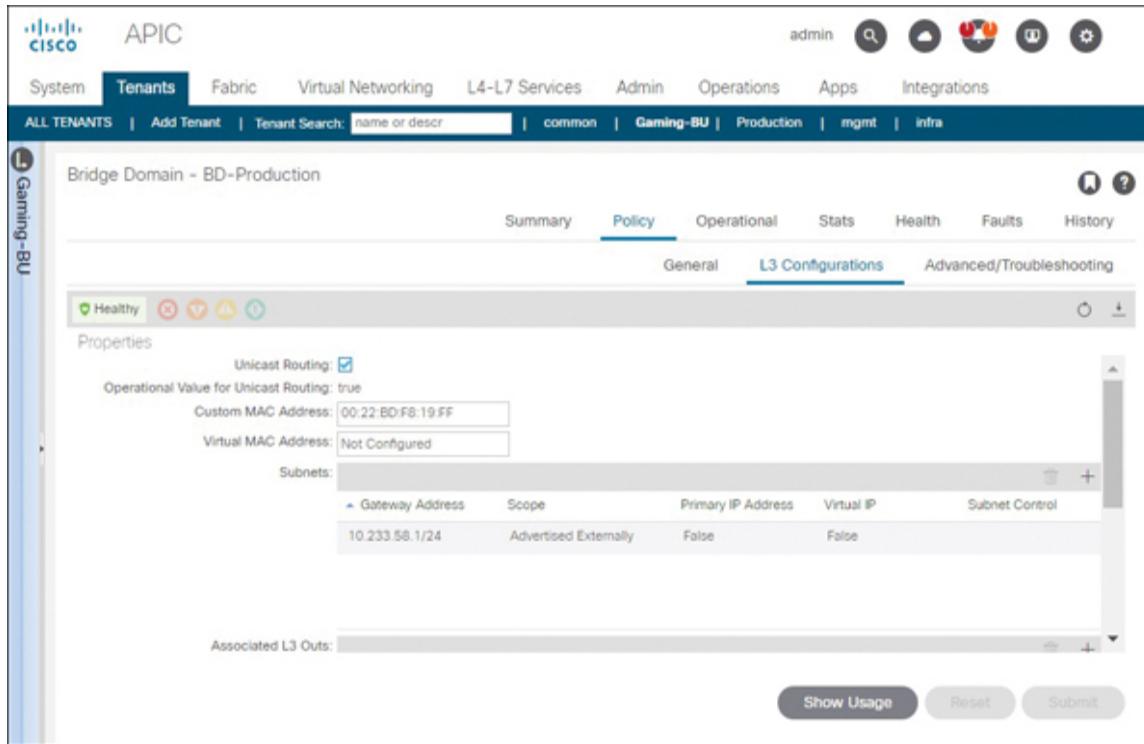
**Figure 9-27** shows a BD subnet scope with Advertised Externally selected.





**Figure 9-27** *Marking a BD Subnet for Redistribution into One or More L3Outs*

Figure 9-28 shows the L3 Configuration tab for a BD after an associated subnet has been marked for advertisement out an L3Out. Notice that this view shows not only the scope of associated subnets but also the fact that Unicast Routing has been enabled. One issue that prevents advertisement of a subnet out an L3Out is Unicast Routing being disabled in the first place.



**Figure 9-28** Verifying That Unicast Routing Is Enabled for BDs with Subnets Advertised Out an L3Out

## Enabling Communications over L3Outs Using Contracts

Once routing into and out of a fabric has been enabled, the next step is to whitelist desired endpoint communication. [Figure 9-21](#) earlier in this chapter provides a list of desired contracts. This section takes a look at how you might go about creating the contract named Admins-to-Fabric-Systems. To match interesting traffic, first create a filter, as shown in [Figure 9-29](#). The only thing that's new here should be the filter entry matching ICMP traffic. As indicated earlier, ICMP filters do not require definition of ports in either direction.



The screenshot shows the APIC interface with the 'Tenants' tab selected. A new filter is being created with the name 'Admin-Protocols'. The filter table lists the following entries:

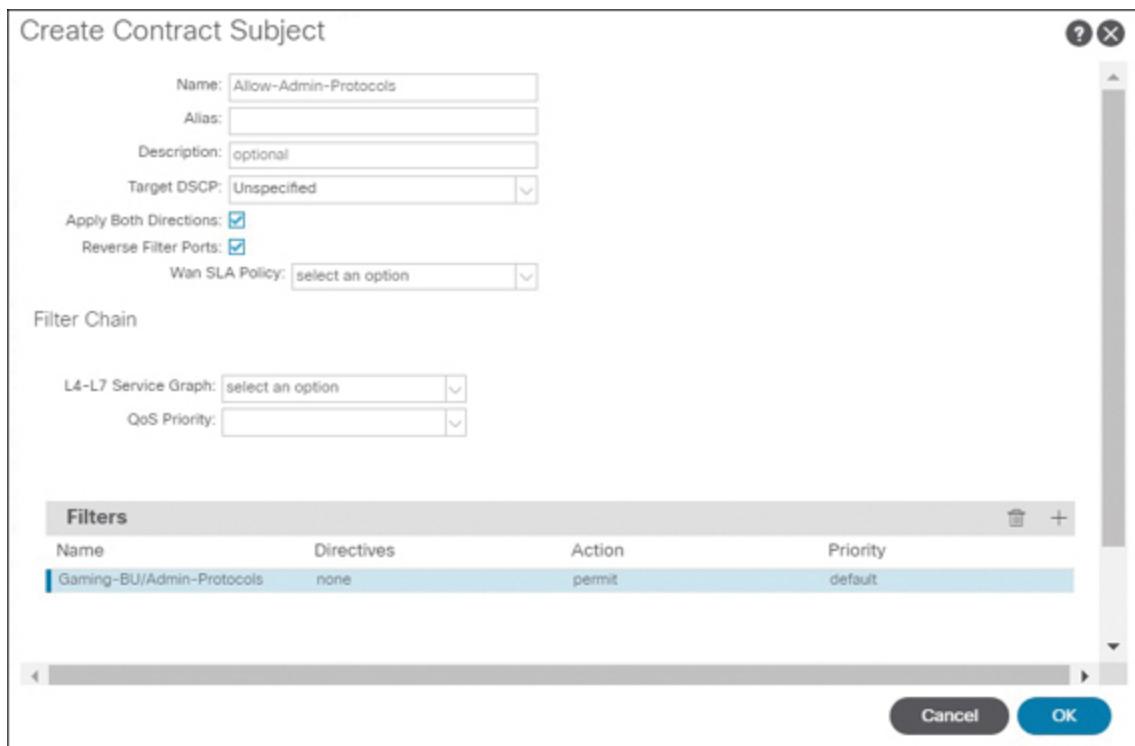
Name	EtherType	ARP Flag	IP Protocol	Match Only Fragments	Stateful	Source Port / Range	Destination Port / Range	TCP Session Rules		
HTTP	IP		tcp	False	False	unspecified	unspecified	80	80	Unspecified
HTTPS	IP		tcp	False	False	unspecified	unspecified	443	443	Unspecified
RDP-TCP	IP		tcp	False	False	unspecified	unspecified	3389	3389	Unspecified
RDP-UDP	IP		udp	False	False	unspecified	unspecified	3389	3389	Unspecified
SSH	IP		tcp	False	False	unspecified	unspecified	22	22	Unspecified
ICMP	IP		icmp	False	False					

At the bottom right are 'Cancel' and 'Submit' buttons.

**Figure 9-29 A Filter That Matches Interesting Traffic**

Figure 9-30 shows that the filter has been added to a subject under the desired contract that permits forwarding of all matched traffic as well as all return traffic.

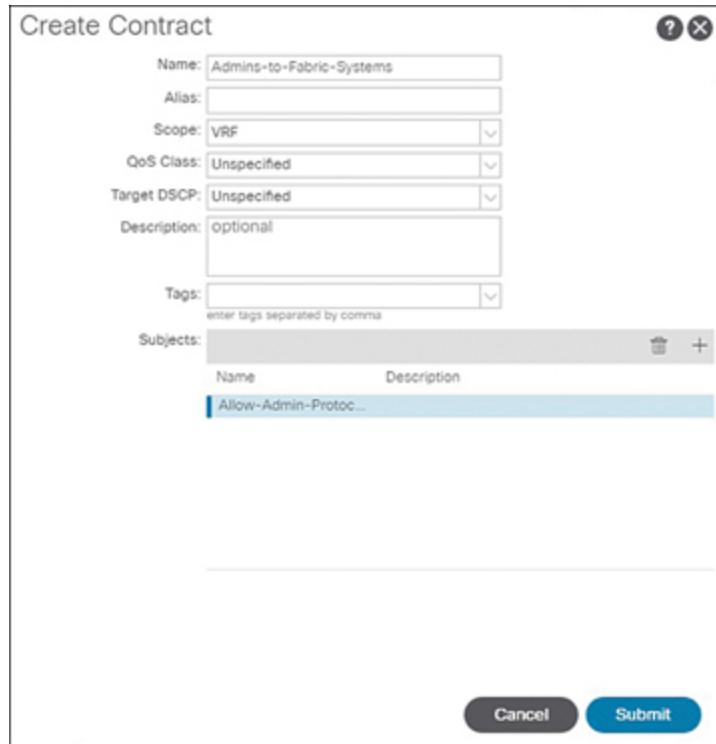




**Figure 9-30** Contract Subject Allowing Traffic Matching the Previous Filter

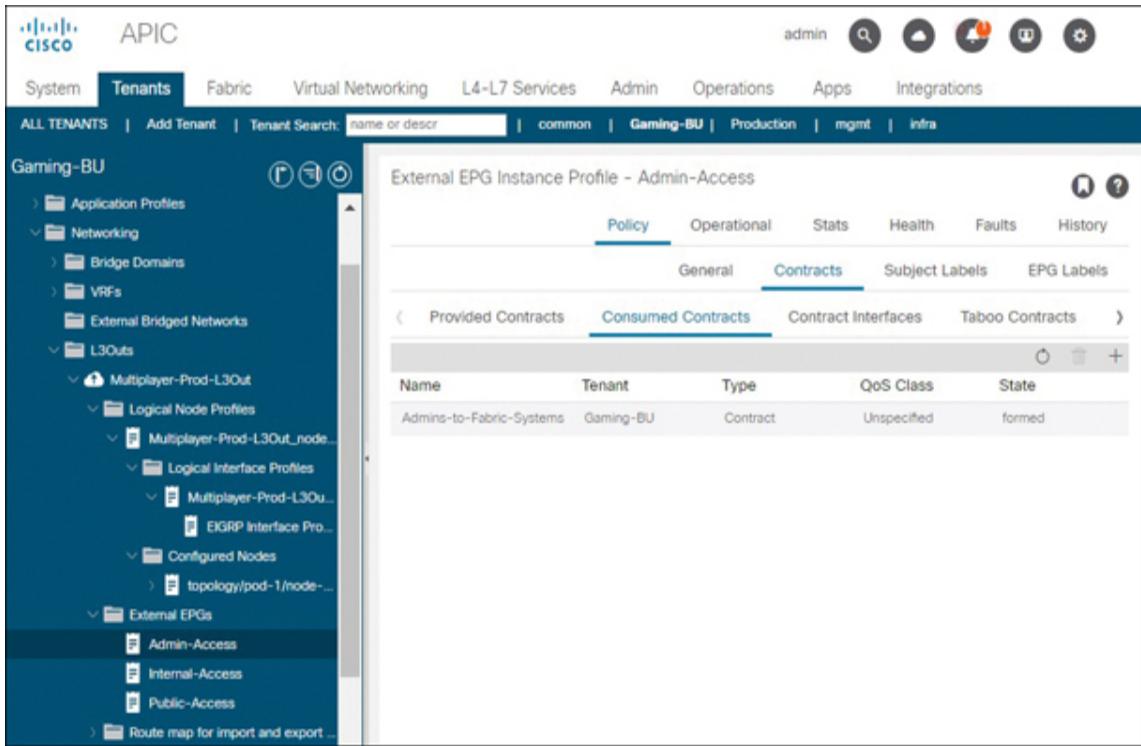
**Key Topic**

Finally, you see that the contract Admins-to-Fabric-Access is ready to be created. Note that the default value, VRF, has been selected as the contract scope in [Figure 9-31](#). It is important to understand that even though this contract is between devices external to the fabric and internal devices, no route leaking between ACI user VRFs is taking place. This is why the scope of VRF is sufficient when the L3Out is not being used as a shared service L3Out.



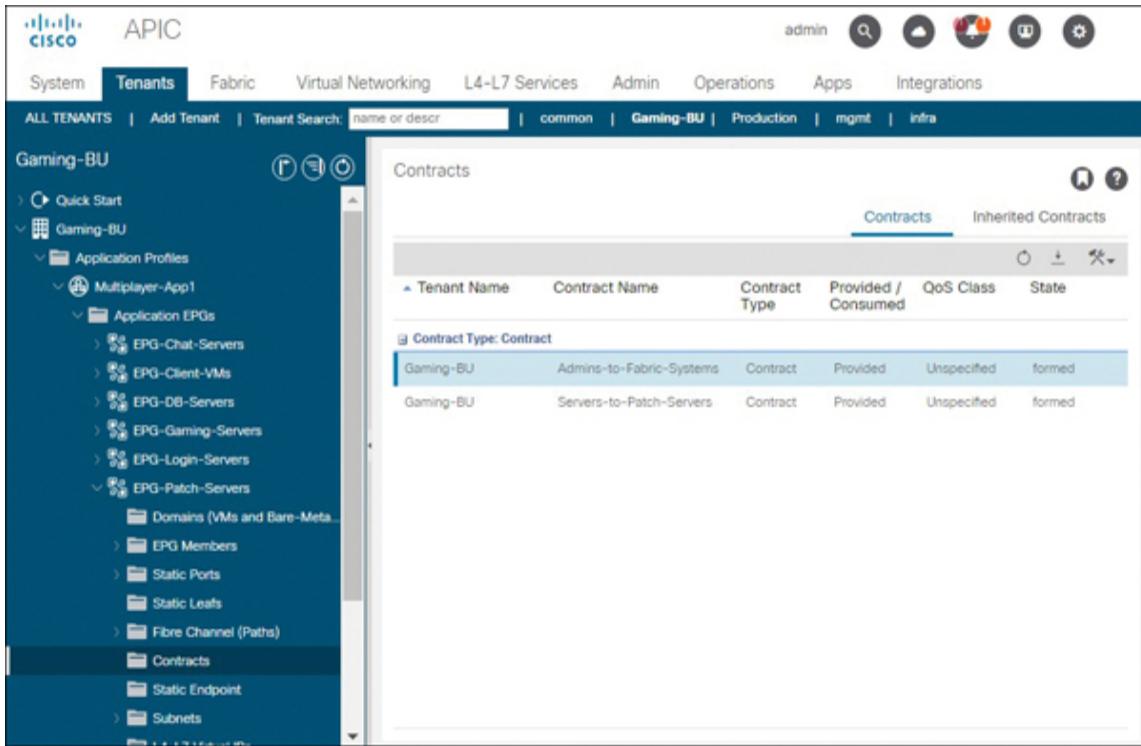
**Figure 9-31** Contract Scope Shown in the Create Contract Wizard

Next, the contract needs to be assigned between external EPGs and internal EPGs. The easiest way to understand provider and consumer directionality for external EPGs is to first realize that external EPGs represent external systems. Will these external systems be initiating requests, or will they be waiting for data center systems to send traffic? Because administrators need to initiate connections such as SSH and HTTPS, Admin-Access should be configured as a consumer of the contract, as indicated in [Figure 9-32](#).



**Figure 9-32** An External EPG Consuming a Contract

Meanwhile, internal EPGs need to be configured as providers of services to the administrators. [Figure 9-33](#) shows the Admins-to-Fabric-Systems being added to an internal EPG in the provided direction.



**Figure 9-33 An Internal EPG Configured to Provide a Service to Administrators**

After the desired contracts are put in place, all endpoints should have the desired connectivity.

Before moving on to the next section, revisit [Figure 9-21](#). An internal EPG called EPG-Patch-Servers is shown consuming a service from the Internet. Basically, for the only EPG with access to initiate communication with Internet systems, the desire is to ensure that endpoints in this EPG can initiate TCP sessions destined to any port, but Internet systems cannot initiate sessions to these patch servers. This is a good example of where you might avoid use of bidirectional filter application and instead use a return filter matching the TCP established bits.

## Deploying a Blacklist EPG with Logging

Sometimes, IT teams identify the need to block or log certain traffic that should be able to enter the data center but should not be allowed to reach servers within ACI. In such cases, you can deploy an external EPG that either has no contracts associated or has a contract that denies and perhaps logs the traffic.

Because external EPGs use a longest-prefix match, all you need to do to classify the intended traffic is to add the relevant subnets or host routes to the blacklist EPG and make sure more specific IP addresses in the blacklisted range(s) have not been allocated to other external EPGs.

[Figure 9-34](#) shows creation of an external EPG for this purpose. Note that one particular host, 10.200.1.5/32, falls within the administrator subnet range. This is completely valid. In this case, ACI would always classify traffic from 10.200.1.5 into this new EPG and not into the Admin-Access external EPG.

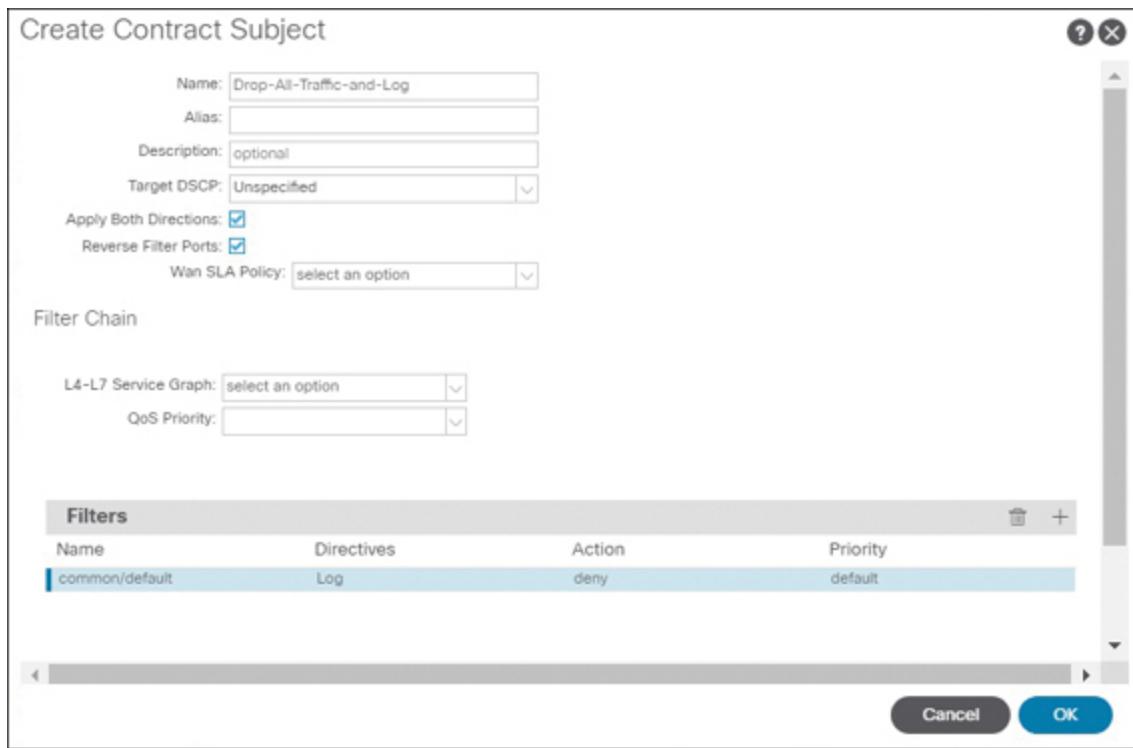
Create External EPG

Name:	Blacklisted-Subnets				
Alias:					
Tags:	<input type="text"/>				
Contract Exception Tag:					
QoS Class:	Unspecified				
Description:	optional				
Target DSCP:	Unspecified				
Preferred Group Member:	<input checked="" type="radio"/> Exclude <input type="radio"/> Include				
Subnet					
IP Address	Scope	Name	Aggregate	Route Control Profile	Route Summarization Policy
10.200.1.5/32	External Subnets for the External EPG				
172.17.0.0/16	External Subnets for the External EPG				

Cancel  Submit

**Figure 9-34** Creation of an External EPG to Classify Specific Endpoints and Ranges

If all you want is to drop traffic from these ranges, you have already succeeded because no contracts have been enforced on this external EPG. However, if you also want to log the traffic, you can create a specific contract that has a subject whose filter matches all traffic. The directive in this case should be Log, and the action should be Deny, as shown in [Figure 9-35](#).



**Figure 9-35** Creation of a Contract Subject to Log and Drop All Traffic

As indicated in [Figure 9-36](#), the contract scope VRF is sufficient for this contract.

Create Contract

Name:

Alias:

Scope:

QoS Class:

Target DSCP:

Description:

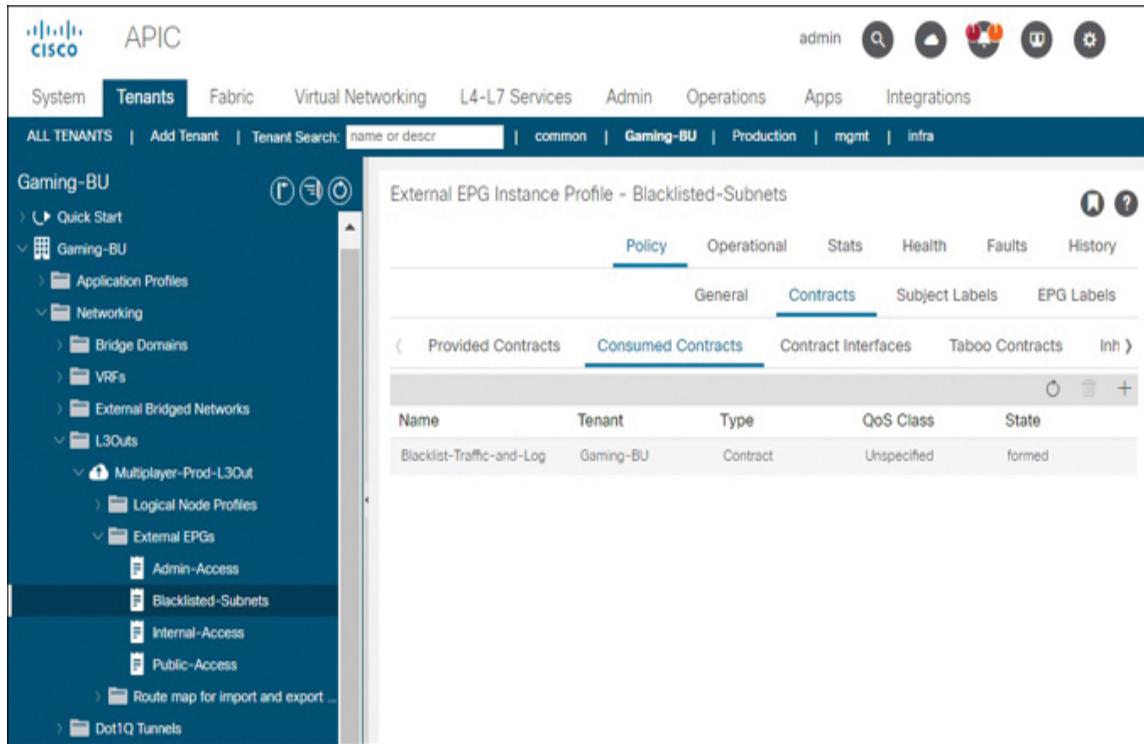
Tags:

Subjects:

Name	Description
Drop-All-Traffic-and-Log	

**Figure 9-36** Contract for Blacklisting Traffic from and to Certain External Sources

Next, you need to apply the contract on the new external EPG in both the consumed and provided directions. [Figure 9-37](#) shows its application as a consumed contract.



**Figure 9-37** Applying a Contract to an External EPG to Blacklist Traffic

Navigate to Tenants and open the tenant in question. Go to the tenant's Operational tab, select Flows, and select L3 Drop. As shown in [Figure 9-38](#), if traffic has been initiated by any of the blacklisted external devices or if any traffic in the VRF was destined to the blacklisted devices, ACI should have a deny log for the traffic flow. Note that contract logging uses processor resources and is rate limited in ACI.

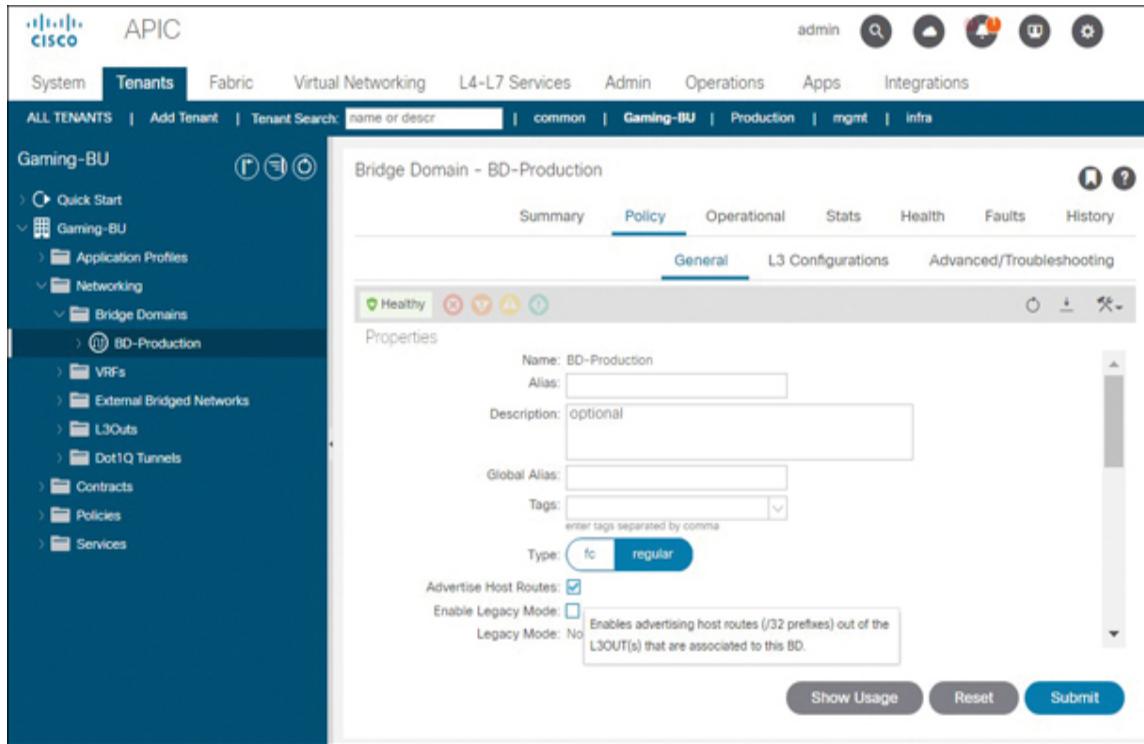
The screenshot shows the APIC interface with the 'Tenants' tab selected. On the left, a navigation tree shows 'Gaming-BU' expanded, with 'Application Profiles' selected. The main area is titled 'Tenant - Gaming-BU' and has tabs for 'Operational', 'Stats', 'Health', 'Faults', and 'History'. Under 'Operational', there are sub-tabs for 'Flows', 'Packets', and 'Resource IDs', with 'Flows' selected. Below these are tabs for 'L2 Permit', 'L3 Permit', 'L2 Drop', and 'L3 Drop', with 'L3 Drop' selected. A table header includes columns for VRF, Src IP, Dest IP, Protocol, Src Port, Dest Port, Src MAC, Dest MAC, and Action. One row in the table is highlighted, showing 'Multiplayer-Prod' as the VRF, '10.200.1.5' as the Src IP, '10.233.58.32' as the Dest IP, 'icmp' as the Protocol, and 'unspecified' for both ports and MAC addresses.

**Figure 9-38** Verifying Dropped Traffic Logged as a Result of a Contract

## Advertising Host Routes Out an ACI Fabric



One of the configuration knobs under bridge domains in recent ACI code revisions is Advertise Host Routes. When this option is enabled, ACI advertises not only BD subnets but also any host routes that ACI has learned within any BD subnet ranges selected for external advertisement. [Figure 9-39](#) shows this configuration checkbox.



**Figure 9-39 Advertising Host Routes Out an L3Out**

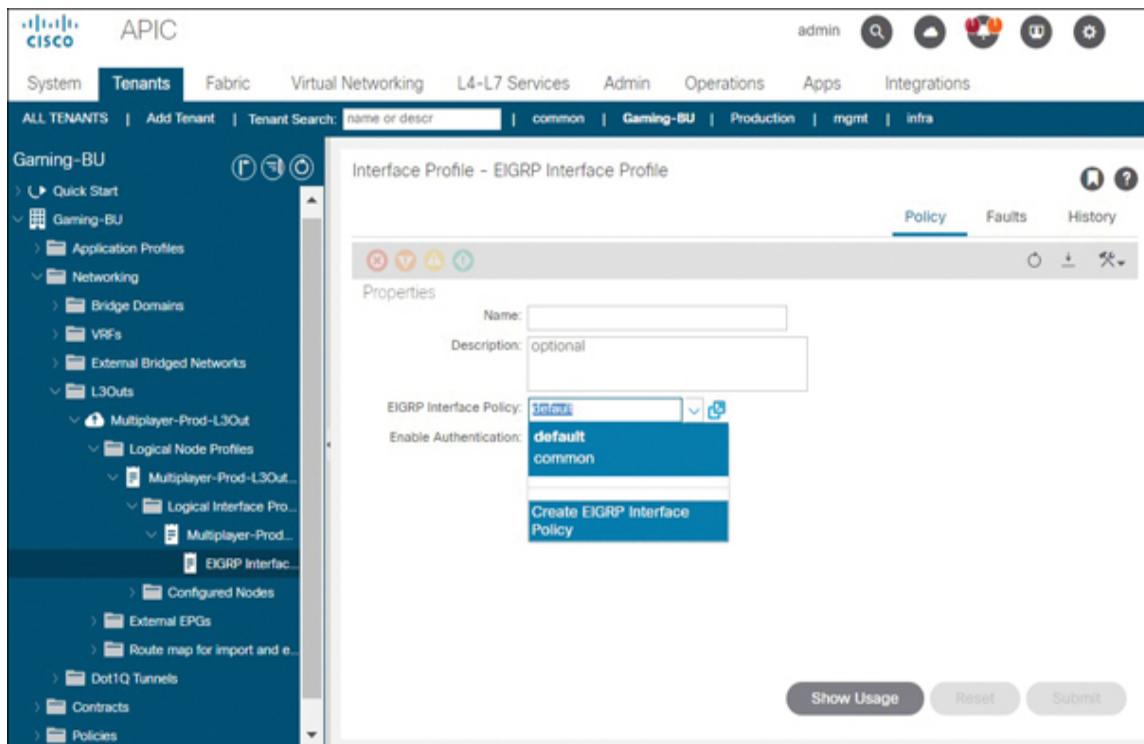
## Implementing BFD on an EIGRP L3Out



A common desire in modern data centers is to implement BFD to enable subsecond routing failover capabilities. To implement BFD on an EIGRP L3Out, enable BFD under the associated EIGRP interface policy. If further customization of BFD settings and timers is required, you can create a BFD interface policy and associate it with the EIGRP L3Out.

[Figure 9-40](#) shows an EIGRP interface policy that is active on an L3Out. This particular L3Out has the default EIGRP interface policy from the common tenant set. If BFD were to be enabled on this particular EIGRP interface policy, all L3Out interfaces in other tenants that consume this policy

would also attempt to establish BFD neighbor relationships out their respective L3Outs. This is likely not what you want to do if you are configuring this in a production fabric. Instead, you can create and associate a new EIGRP interface policy with the L3Out EIGRP interface profile.



**Figure 9-40** Verifying an Operational EIGRP Interface Policy on an L3Out

Figure 9-41 shows the Create EIGRP Interface Policy page. Notice that this page also allows tuning of EIGRP hello and hold intervals as well as tuning of bandwidth and delay for route metric manipulation. Enable the BFD checkbox and click Submit.

Create EIGRP Interface Policy

Name: EIGRP-with-BFD

Description: optional

Control State:    
 BFD  
 Self Nexthop  
 Passive  
 Split Horizon

Hello Interval (sec): 5

Hold Interval (sec): 15

Bandwidth: 0

Delay: 0 tens of microseconds

Cancel Submit



**Figure 9-41** Configuring a New EIGRP Interface Policy with BFD Enabled

ACI comes preconfigured with a default global BFD policy located under **Fabric > Access Policies > Policies > Switch > BFD > BFD IPv4/v6 > default**. To customize timers or to enable subinterface optimization for BFD for an individual L3Out, navigate to a logical interface profile on the L3Out and select Create BFD Interface Profile. The Create BFD Interface Profile window shown in [Figure 9-42](#) appears. Select Create BFD Interface Policy.

**Figure 9-42** Launching the BFD Interface Profile Creation Wizard

Then, on the Create BFD Interface Policy page, shown in [Figure 9-43](#), select the desired settings and click Submit. If

neighboring devices that peer with ACI over the L3Out also have BFD enabled, BFD should become fully operational.

The screenshot shows a configuration dialog titled 'Create BFD Interface Policy'. It includes fields for 'Name' (Gaming-BU-BFD-Int-Pol), 'Description' (optional), and 'Admin State' (Enabled). Other settings include 'Control State' (checked for 'Enable sub-interface optimization'), 'Detection Multiplier' (3), 'Minimum Transmit Interval (msec)' (50), 'Minimum Receive Interval (msec)' (50), 'Echo Receive Interval (msec)' (50), and 'Echo Admin State' (Enabled). At the bottom are 'Cancel' and 'Submit' buttons.

**Figure 9-43** L3Out BFD Timer and Policy Customization

BFD interface policies are not protocol specific and so are not revisited in future sections.

### Note

In addition to BFD, the following EIGRP interface policy Control State options are available:

- **Self Nexthop:** This option is enabled by default. By default, EIGRP sets its local IP address as the next hop when advertising routes. When you disable this option, the border leaf does not overwrite the next hop and keeps the original next-hop IP address.
- **Passive:** This option is used to configure the interfaces as an EIGRP passive interface. This option is disabled by

default.

- **Split Horizon:** Split horizon is a feature that helps prevent routing loops by not sending EIGRP updates or queries to the interface where it was learned. This option is enabled by default.

It is very uncommon for engineers to need to modify any of these three settings from their defaults.

## Configuring Authentication for EIGRP

ACI supports authentication of EIGRP peers using MD5, but routing protocol authentication is not very common in data centers today. EIGRP authentication, therefore, does not warrant extensive coverage here.

To enable authentication with EIGRP neighbors, navigate to the EIGRP interface profile under the EIGRP L3Out and select Enable Authentication. Then either select the default keychain policy from the common tenant from the EIGRP KeyChain Policy drop-down or define a new one.

You can define EIGRP keychain policies by navigating to **Tenant > Policies > Protocol > EIGRP > EIGRP KeyChains**.

A keychain policy is a collection of key policies. Each key policy consists of a key ID, a key name, a pre-shared key, a start time, and an end time.

The only caveat to point out is that because EIGRP authentication is implemented at the logical interface profile level, the use of multiple logical interface profiles becomes necessary if some EIGRP peers are required to authenticate over the L3Out while others are not.

# EIGRP Customizations Applied at the VRF Level

If you select a VRF within a tenant and click the Policy menu, one of the configuration sections you see on the Policy page is EIGRP Context per Address Family. This is where you can modify certain VRF-wide settings for EIGRP by deploying a custom EIGRP address family context policy for IPv4 or IPv6. The configuration settings that together form an EIGRP address family context policy are described in [Table 9-2](#).



**Table 9-2** Customizable Settings for an EIGRP Address Family Context Policy

Customizable Setting	Description
neighbor	Identifies the neighbors to which EIGRP routes are advertised.
filter	Specifies the filter applied to the routes learned from a neighbor.
group	Specifies the group to which the EIGRP process belongs.
range	Specifies the range of IP addresses for the EIGRP process.
timers	Specifies the timers for the EIGRP process.
parameters	Specifies the parameters for the EIGRP process.

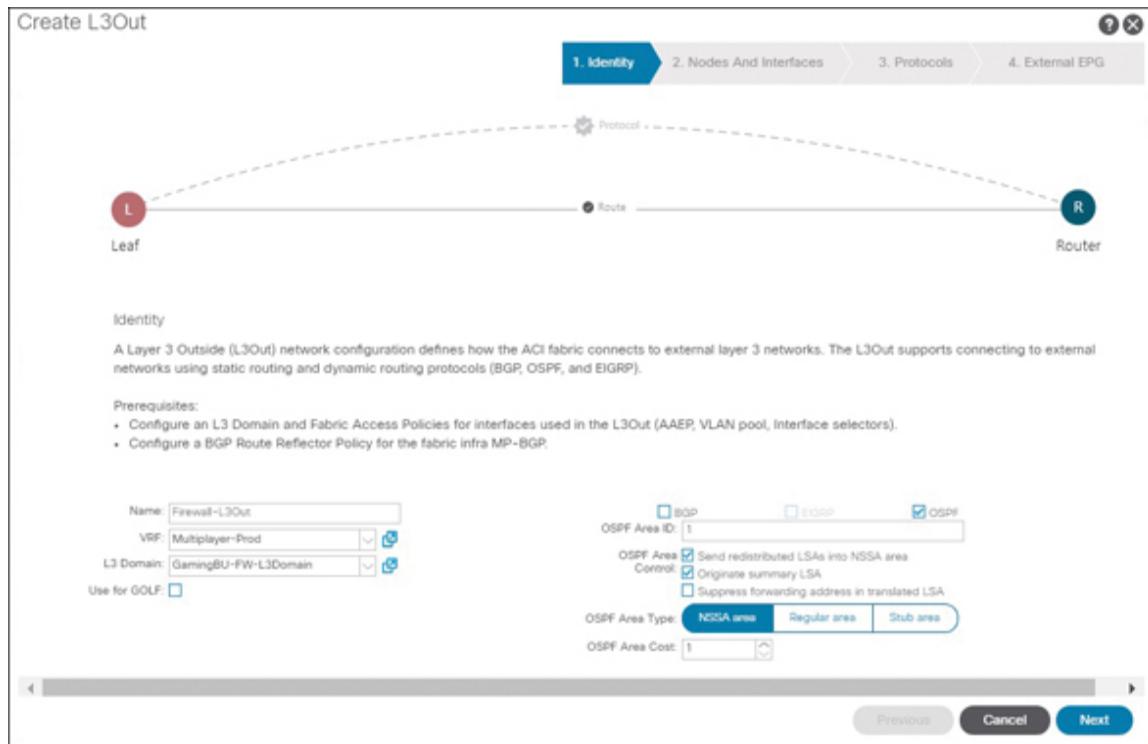
Ac tiv er va l (m in)	The interval the border leaf waits after an EIGRP query is sent before declaring a stuck in active (SIA) situation and resetting the neighborship. The default is 3 minutes.
Ex ter na l Di st an ce	The administrative distance (AD) for external EIGRP routes. The default AD for external routes is 170.
Int er na l Di st an ce	The AD for internal EIGRP routes. The default AD for internal routes is 90.

M ax im u m Pa th Li mi t	The maximum number of equal-cost multipathing (ECMP) next-hop addresses EIGRP can install into the routing table for a prefix. The default is eight paths.
M etr ic St yl e	EIGRP calculates its metric based on bandwidth and delay along with default K values. However, the original 32-bit implementation cannot differentiate interfaces faster than 10 Gigabit Ethernet. This original implementation is called the classic, or narrow, metric. To solve this problem, a 64-bit value with an improved formula was introduced for EIGRP; this is called the wide metric. Valid values for metric style are narrow metric and wide metric. The default is the narrow metric.

## Configuring an L3Out for OSPF Peering

Let's revisit the requirements from [Figure 9-10](#). The next L3Out that needs to be provisioned is Firewall-L3Out, which will leverage SVIs over a vPC to peer with a firewall via OSPF. [Figure 9-44](#) shows the first page of the L3Out creation wizard. Enter parameters for name, VRF, and L3 domain. Enable OSPF by selecting the OSPF checkbox. In this example, the OSPF area ID 1 has been selected, and NSSA

Area is selected for OSPF Area Type. Select the parameters that meet your requirements and click Next. Note that each OSPF L3Out can place interfaces in a single area. If peerings in multiple areas are required, you need to deploy additional L3Outs.



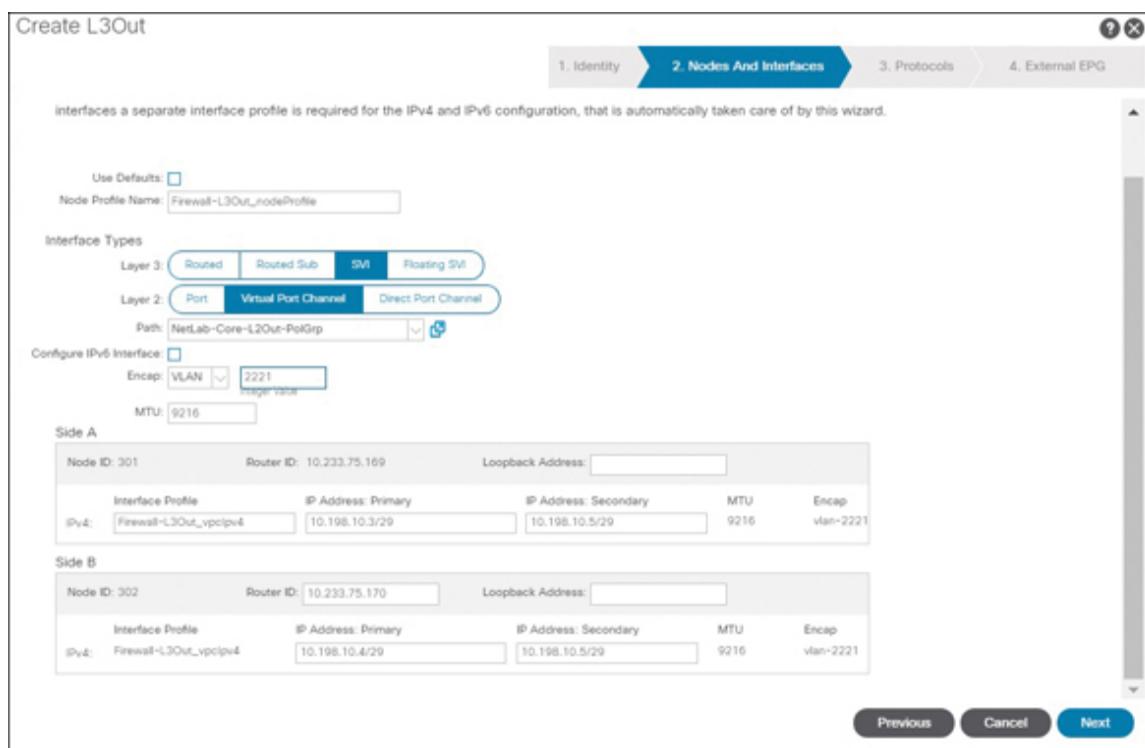
**Figure 9-44** Select Global Routing Protocol Parameters for OSPF L3Out



On the Nodes and Interfaces page, shown in [Figure 9-45](#), select the interface type that will be deployed for the L3Out. In this case, SVIs will be used. When deploying SVIs on a vPC in ACI, select different primary IP addresses for each

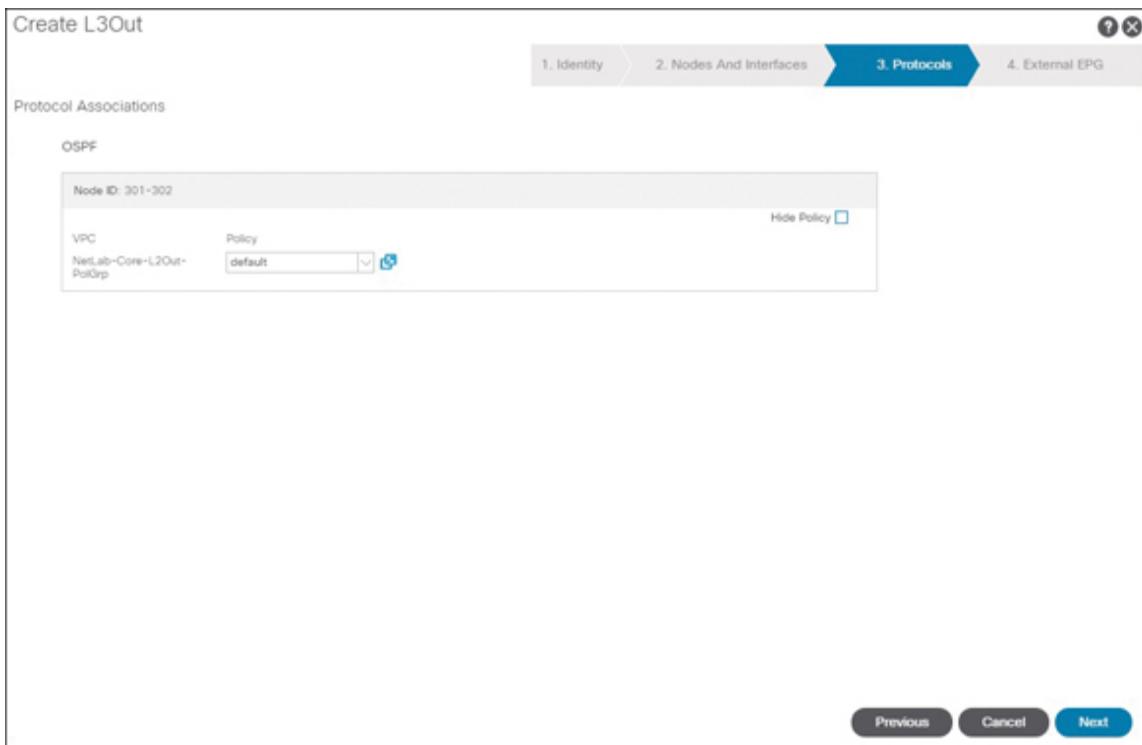
vPC peer. If the external device behind the L3Out needs to point to a common IP address that both border leaf switches respond to, you can define a **secondary IP address** by using the IP Address: Secondary field. That is not required in this case due to the reliance on OSPF, but this same L3Out will also be used to demonstrate static routing in ACI. Also, note the Encap field. Enter the VLAN ID that needs to be trunked out the L3Out to the external device here. The VLAN ID selected here must be in the VLAN pool associated with the L3 domain for the L3Out. Click Next when you're finished making selections.

### Key Topic



**Figure 9-45** Configuring L3Out SVIs Trunked onto a vPC with Secondary IP Addresses

The Protocol Associations page for OSPF, shown in [Figure 9-46](#), allows you to associate an OSPF interface policy to the L3Out OSPF interface profile. The default OSPF interface policy resides in the common tenant. Click Next.



**Figure 9-46** *Associating an OSPF Interface Policy with the OSPF Interface Profile*

The final step in creating the L3Out is to configure an external EPG. A critical thing to understand about external EPGs used for traffic classification is that their scope is VRF-wide. This means that if you have already configured all your desired subnets for classification using the scope External Subnets for External EPG on another L3Out associated with a VRF, there is no need to duplicate these configurations on a secondary L3Out in the same VRF. In fact, if you try to add a subnet previously classified by another external EPG in the VRF to an external EPG on a second L3Out, ACI raises an error and does not deploy the erroneous configuration. However, one problem remains.

ACI L3Out still expects to see at least one external EPG before it attempts to form routing protocol adjacencies with external devices. [Figure 9-47](#) shows the creation of a dummy external EPG called Placeholder. No subnets need to be associated with this external EPG, whose only function is to ensure that ACI enables the L3Out. Click Finish to deploy the L3Out.



Create L3Out

External EPG

The L3Out Network or External EPG is used for traffic classification, contract associations, and route control policies. Classification is matching external networks to this EPG for applying contracts. Route control policies are used for filtering dynamic routes exchanged between the ACI fabric and external devices, and leaked into other VRFs in the fabric.

Name: Placeholder  
Provided Contract: select a value  
Consumed Contract: select a value

Default EPG for all external networks:

Subnets

IP Address	Scope	Name	Aggregate	Route Control Profile	Route Summarization Policy

Previous Cancel Finish

This screenshot shows the 'Create L3Out' configuration interface. The top navigation bar has four tabs: 'Identity', 'Nodes And Interfaces', 'Protocols', and 'External EPG'. The 'External EPG' tab is selected and highlighted in blue. Below the tabs, there's a section titled 'External EPG' with a detailed description. Underneath, there are input fields for 'Name' (Placeholder), 'Provided Contract' (a dropdown menu), and 'Consumed Contract' (another dropdown menu). A checkbox labeled 'Default EPG for all external networks' is present. At the bottom, there's a table titled 'Subnets' with columns for IP Address, Scope, Name, Aggregate, Route Control Profile, and Route Summarization Policy. The table is currently empty. At the very bottom of the interface are three buttons: 'Previous', 'Cancel', and 'Finish'.

**Figure 9-47** Configuring a Dummy External EPG to Enable the L3Out

To verify OSPF adjacency establishment via the leaf CLI, you can execute the command **show ip ospf neighbors vrf < vrf name >**. [Example 9-5](#) shows that ACI has learned a default route from the adjacent firewall.

## **Example 9-5 Verifying Routes Learned via OSPF**

[Click here to view code image](#)

```
LEAF301# show ip route ospf vrf Gaming-BU:Multiplayer-Prod
IP Route Table for VRF "Gaming-BU:Multiplayer-Prod"
*' denotes best ucast next-hop
**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

0.0.0.0/0, ubest/mbest: 1/0
  *via 10.198.10.2, vlan76, [110/5], 00:01:02, ospf-
    default, inter
```

## **A Route Advertisement Problem for OSPF and EIGRP L3Outs**



One problem commonly experienced by engineers who dedicate a pair of switches to border leaf functions and need to create both OSPF and EIGRP L3Outs on these switches within the same VRF is that any BD subnets marked for advertisement out one L3Out also gets advertised out the other L3Out.

The reason for this behavior is that the route map ACI automatically generates as a result of BD subnet advertisements is common across OSPF and EIGRP. There are two common and recommended solutions for avoiding this behavior:

**Key Topic**

- Deploy OSPF and EIGRP L3Outs for a given VRF on different border leaf switches.
- Use BGP instead of OSPF and EIGRP. This recommendation is due to the fact that ACI generates route maps for BGP on a per-L3Out basis.

## Implementing BFD on an OSPF L3Out

To enable BFD on an OSPF L3Out, select the OSPF interface profile for the L3Out. Either edit the default interface policy associated with the common tenant or create a new OSPF interface policy to associate with the L3Out OSPF interface profile. [Figure 9-48](#) shows creation of a new OSPF interface policy that enables BFD.

**Create OSPF Interface Policy**

Name:	Custom-OSPF-Int-Policy		
Description:	optional		
Network Type:	Broadcast	Point-to-point	Unspecified
Priority:	1		
Cost of Interface:	unspecified		
Interface Controls:	<input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> Advertise subnet <input checked="" type="checkbox"/> BFD <input type="checkbox"/> MTU ignore <input type="checkbox"/> Passive participation	
Hello Interval (sec):	10		
Dead Interval (sec):	40		
Retransmit Interval (sec):	5		
Transmit Delay (sec):	1		

**Cancel** **Submit**

**Figure 9-48** Enabling BFD on a Custom OSPF Interface Policy

### Note

In addition to BFD, the following OSPF interface policy Control State options are available:

- **Advertise Subnet:** This allows OSPF to advertise a loopback IP address with its subnet instead of /32 without requiring that the network type be changed from loopback to point-to-point. However, in ACI, a loopback IP address is always configured with /32. Hence, at the time of writing, this option does not do anything in particular.
- **MTU Ignore:** This option allows the OSPF neighborship to form even with a mismatching MTU. This option is intended to be enabled on an OSPF interface with a lower MTU. Use of this option is not recommended in general.
- **Passive Participation:** This option configures the interfaces as OSPF passive interfaces.

Modifying any of these settings from their defaults is very uncommon in ACI.

## OSPF Customizations Applied at the VRF Level

Just as with EIGRP, there are some VRF-wide settings for OSPF. A number of these settings are documented in [Table 9-3](#). All OSPF timer values have been removed to limit coverage. A custom OSPF timer policy can be applied either to all address families within a VRF using the OSPF Timers

drop-down box on the VRF Profile menu or to an individual address family (IPv4 or IPv6).



**Table 9-3** Customizable Settings Besides Timers in an OSPF Timer Policy

Config uration Paramet er	Description
Bandwid th Referen ce (Mbps)	Specifies the reference bandwidth used to calculate the default metrics for an OSPF interface. The default is 40,000 Mbps (40 Gbps).
Admin Distanc e Preferen ce	Specifies the administrative distance (AD) for OSPF routes. The default is 110.
Maximu m ECMP	Specifies the maximum number of ECMP that OSPF can install into the routing table. The default is 8 paths.

Enable Name Lookup for Router IDs	Prompts ACI to display router IDs as DNS names in OSPF <b>show</b> commands. This is disabled by default.
Prefix Suppression	Reduces the number of Type 1 (router) and Type 2 (network) LSAs installed in the routing table. This option is disabled by default.
Graceful Restart Helper	Keeps all the LSAs that originated from the restarting router during the graceful restart period. ACI border leaf switches do not themselves perform OSPF graceful restarts. ACI enables this option by default.

## Adding Static Routes on an L3Out

Say that members of an IT team think that a firewall or an appliance they have procured and enabled for dynamic routing is not very reliable and that the routing protocol in the current appliance code may crash. Or say that they want to disable an appliance altogether and just leverage static routing. Sometimes, an appliance may lack the capability to leak a default route into a specific dynamic routing protocol. In such cases, static routes can be deployed on an L3Out either alongside a dynamic routing protocol or by themselves, and they can be distributed throughout the fabric via MP-BGP.

**Key Topic**

If an L3Out does not require dynamic routing, you can leave the checkboxes for EIGRP, OSPF, and BGP within the L3Out unchecked. Static routing does not need to be configured alongside dynamic routing protocols.

**Key Topic**

Static routes in ACI are configured on individual fabric nodes. Select the node of interest under the Configured Nodes folder and click the + sign under the Static Routes view. [Figure 9-49](#) shows the configuration of a static default route. The Preference value ultimately determines the administrative distance of the static route. There are two places where such values can be defined. Each next hop can assign its own preference. If next-hop preference values are left at 0, the base Preference setting in the Prefix field takes effect. Configure the Prefix, Preference, and Next Hop IP settings for a static route and click Submit. If you create a route without setting a next-hop IP address, ACI assigns the NULL interface as the next hop.

**Key Topic**

Create Static Route

Prefix: 0.0.0.0/0

Preference: 1

Nexthop Type: Static Route

Route Control:  BFD

Track Policy: select an option

Description: optional

Next Hop Addresses:

Next Hop IP	Preference
10.198.10.2	0

If there is no next hop address added, a NULL interface will be automatically created.

**Cancel** **Submit**

**Figure 9-49** Adding a Static Route on an L3Out

Note the BFD checkbox. Enabling BFD for a static route leads to the subsecond withdrawal of the static route from the routing table in the event that the BFD session on the egress interface for the static route goes down.

## Implementing IP SLA Tracking for Static Routes

With IP service-level agreement (SLA) tracking for static routes, introduced in ACI Release 4.1(1), you have the option to collect information about network performance in real time by tracking an IP address using ICMP or TCP probes; you can influence routing tables by allowing a static route to be removed when tracking results are negative and returning the routes to the table when the results become positive again. IP addresses tracked can be next-hop IP

addresses, external addresses several hops away, or internal endpoints.

To implement IP SLA tracking for static routes, perform the following steps:



- Step 1.** Create an IP SLA monitoring policy if custom probe frequency and multipliers are desired.
- Step 2.** Define the IP addresses to be probed. These are called track members.
- Step 3.** Create and associate a track list with either an individual static route or a next-hop address for the static route.



To create an IP SLA monitoring policy, navigate to the tenant in which the policy should be created, double-click the Policies folder, double-click the Protocol folder, right-click the IP SLA folder, and select Create IP SLA Monitoring Policy. [Figure 9-50](#) shows creation of an IP SLA policy using default settings. The SLA Frequency field specifies the frequency, in seconds, to probe the track member IP address. Acceptable values range from 1 second to 300 seconds. The Detect Multiplier setting defines the number of missed probes in a row that moves the track object into a down state. The SLA Type setting defines which protocol is used for probing the track member IP. For L3Out static routes, the supported options are either ICMP or TCP with a specified destination port.

Create IP SLA Monitoring Policy

Name:  ? X

Description:

SLA Frequency (sec):  ↑ ↓

Detect Multiplier:  ↑ ↓

SLA Type: ICMP L2Ping TCP

Cancel Submit

The dialog box has a title 'Create IP SLA Monitoring Policy'. It contains several input fields: 'Name' with the value 'IP-SLA-Policy', 'Description' with the value 'optional', 'SLA Frequency (sec)' set to 60, and 'Detect Multiplier' set to 3. Below these are three buttons for 'SLA Type': 'ICMP' (which is highlighted in blue), 'L2Ping', and 'TCP'. At the bottom are two buttons: 'Cancel' and 'Submit'.

**Figure 9-50** Creating an IP SLA Monitoring Policy



Define each IP address that needs to be probed in a track member object by right-clicking the Track Members or IP SLA folder and selecting Create Track Member. [Figure 9-51](#) shows the track member IP address entered in the Destination IP field. The Scope of Track Member drop-down allows you to define the component (L3Out or BD) on which the destination IP address should exist. The IP SLA monitoring policy should also be specified in the IP SLA Policy drop-down unless the default IP SLA policy in the common tenant is desired.

Create Track Member

Name: Next-Hop-to-Firewalls

Description: optional

Destination IP: 10.198.10.2

Scope of Track Member: L3 Outside - Firewall-L3Out

IP SLA Policy: IP-SLA-Policy

Cancel Submit

The dialog box has a title 'Create Track Member' at the top right with a question mark icon and a close button. Below the title are five input fields: 'Name' containing 'Next-Hop-to-Firewalls', 'Description' containing 'optional', 'Destination IP' containing '10.198.10.2', 'Scope of Track Member' set to 'L3 Outside - Firewall-L3Out' with a dropdown arrow, and 'IP SLA Policy' set to 'IP-SLA-Policy' with a dropdown arrow. At the bottom are two buttons: 'Cancel' in a grey rounded rectangle and 'Submit' in a blue rounded rectangle.

**Figure 9-51** Creating a Track Member



Create a track list by right-clicking either the IP SLA folder or the Track Lists folder and selecting Create Track List. Alternatively, you can create a new track list and simultaneously associate it with a static route or next-hop address on the Static Route page. The Type of Track List field has Threshold Percentage and Threshold Weight as options. Use the percentage option if all track members are at the same level of importance. Use the weight option to assign a different weight to each track member for more granular threshold conditions. With Threshold Percentage selected, ACI removes the associated static route(s) from the routing table when probes to the percentage of track members specified by the Percentage Down field become unreachable. ACI then reenables the static route(s) only.

when probes to the percentage of track members specified by Percentage Up become reachable. Similarly, the Weight Up Value and Weight Down Value fields, which are exposed when Threshold Weights is selected, determine the overall weight that needs to be reached for associated static routes to be removed or re-introduced into the routing table.

[Figure 9-52](#) shows a track list which demands that probes be sent against two track member IP addresses. Because two objects have been included in the track list, reachability to each destination contributes 50% to the overall threshold of the track list. The Percentage Up value 51 indicates that both track members need to be up for the static route to be added to the routing table. In this example, ACI withdraws the associated route(s) even if one track member becomes unreachable due to the Percentage Down value 50.

**Create Track List**

Name:  Description: optional

Type of Track List:  Threshold percentage  Threshold weight

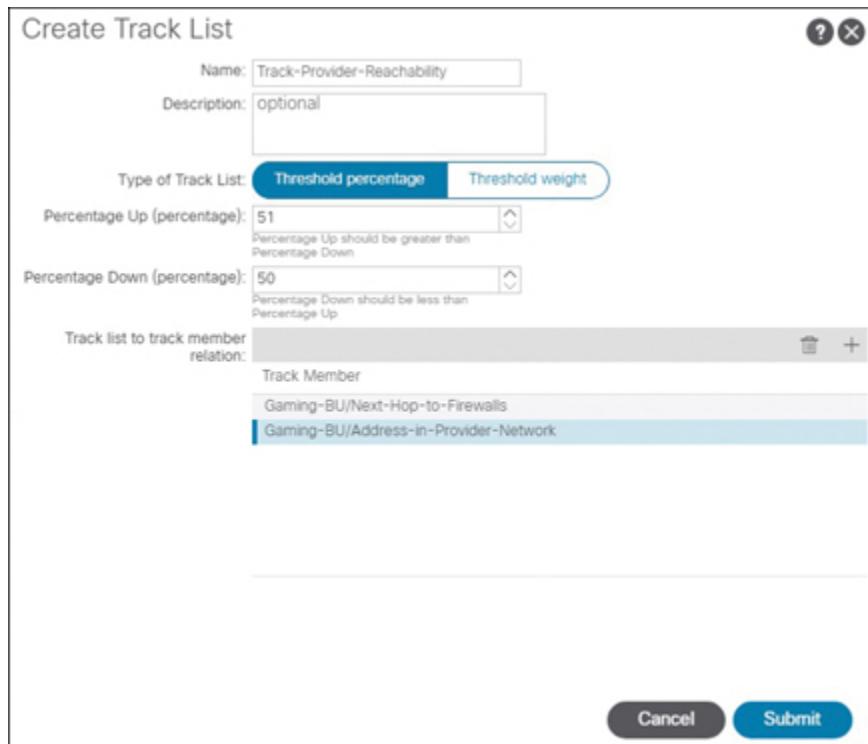
Percentage Up (percentage):  Percentage Up should be greater than Percentage Down

Percentage Down (percentage):  Percentage Down should be less than Percentage Up

Track list to track member relation:

Track Member
Gaming-BU/Next-Hop-to-Firewalls
Gaming-BU/Address-in-Provider-Network

**Cancel** **Submit**



**Figure 9-52** Creating a Track List

[Example 9-6](#) shows that when this track list is assigned to the static default route created earlier, ACI implements a track object for each track member and track list. The overall status of the track list is *down* because the Percentage Down condition has been hit due to lack of reachability to one track member. This prompts the associated route prefix to be withdrawn from the routing tables of all ACI switches.

### **Example 9-6 Verifying Tracking of Static Routes**

[Click here to view code image](#)

```
LEAF301# show track

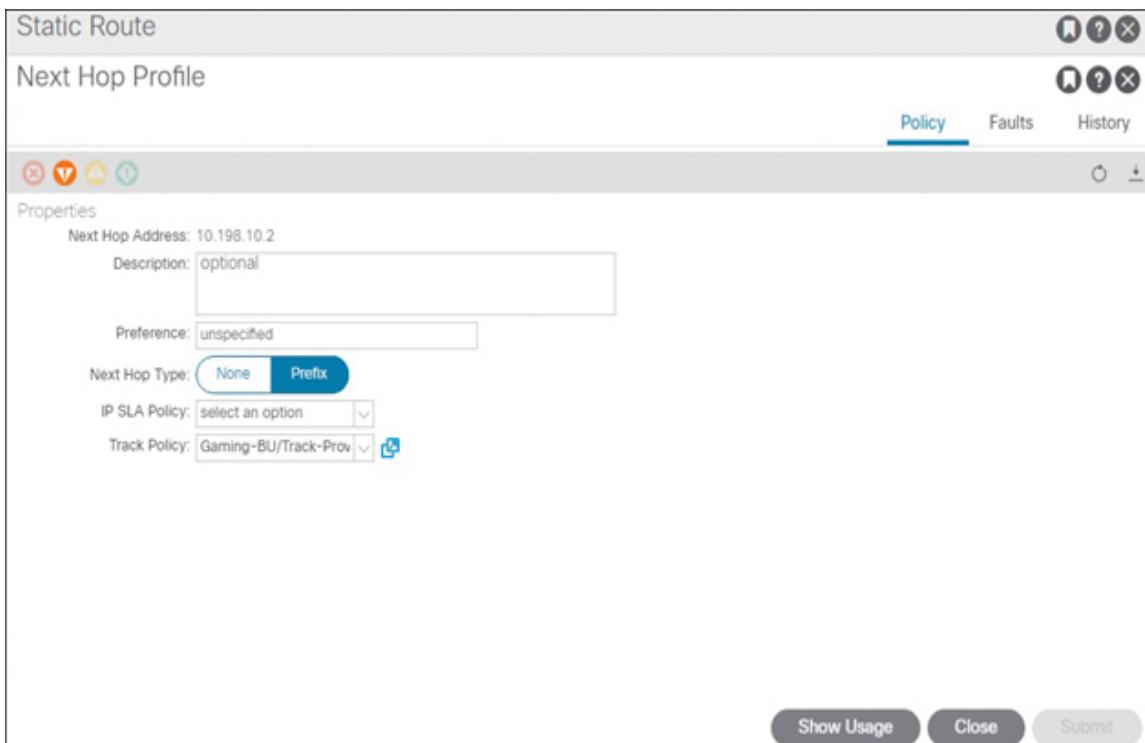
Track 2
    IP SLA 2256
        reachability is down
        1 changes, last change 2020-07-11T19:23:25.179+00:00
        Tracked by:
            Track List 1

Track 3
    IP SLA 2288
        reachability is up
        2 changes, last change 2020-07-11T19:23:25.181+00:00
        Tracked by:
            Track List 1

Track 1
    List Threshold percentage
    Threshold percentage is down
    1 changes, last change 2020-07-11T19:23:25.176+00:00
    Threshold percentage up 51% down 50%
    Tracked List Members:
        Object 3 (50)% up
```

```
Object 2 (50)% down  
Attached to:  
Route prefix 0.0.0.0/0
```

As noted earlier, track lists can also be associated with a next-hop IP address for a static route. [Figure 9-53](#) shows configurable options besides the next-hop IP address that are available on the Next Hop Profile page. Next Hop Type can be set to None or Prefix. None is used to reference the NULL interface as the next hop of a static route. ACI accepts None as a valid next-hop type only if 0.0.0.0/0 is entered as the prefix. Alternatively, a static route without a next-hop entry is essentially a route to the NULL interface. Prefix is the default option for Next Hop Type and allows users to specify the actual next-hop IP address.



**Figure 9-53** Options Available on the Next Hop Profile Page

Applying an IP SLA policy on a next-hop entry for a static route is functionally equivalent to creating a track list with the next-hop address as its only track member and applying the resulting track list to the next-hop entry. In other words, referencing an IP SLA policy on a next-hop entry provides a shortcut whereby the APIC internally creates a track list with the next-hop IP address as the probe IP address.

### Note

One difference between applying a track list to a static route and applying it to a next-hop entry is apparent when a backup floating static route for the same prefix exists on the same leaf switch(es) even when configured on a different L3Out. In this case, a track list applied to the primary static route (lower preference value) prevents the floating static route (higher preference value) from being added to the routing table. This behavior is not experienced when the track list is applied to the next-hop entry of the primary static route.

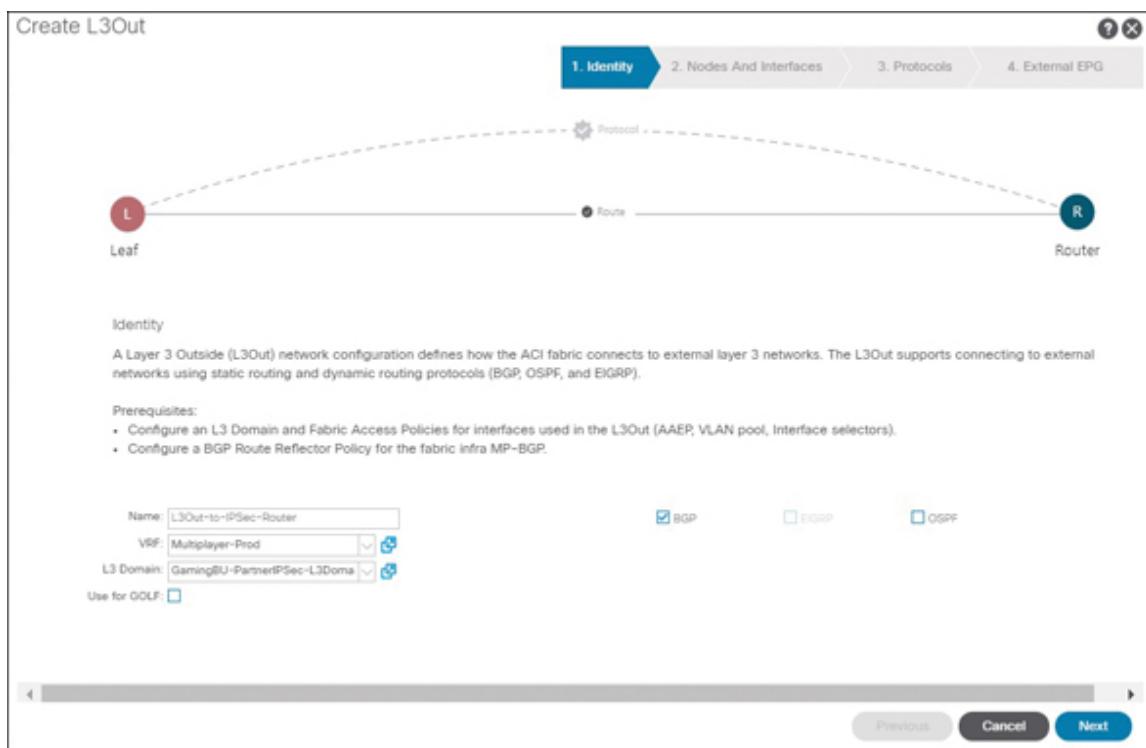
## Configuring an L3Out for BGP Peering

ACI supports both iBGP and eBGP peerings with external routers. A BGP-enabled L3Out automatically belongs to the same BGP ASN configured for route reflection. You need to define each BGP peering under a BGP peer connectivity profile. If there is a need for the fabric to appear as an ASN other than the one configured for route reflection, you can tweak the *local-as* settings in the BGP peer connectivity profile for the intended BGP neighbor.

Given that BGP designs often require peering with neighbors that may be several hops away, an important topic with BGP

is establishing IP reachability to potential neighbor peering addresses. Supported methods for BGP peering IP reachability in ACI are direct connections, static routes, and OSPF.

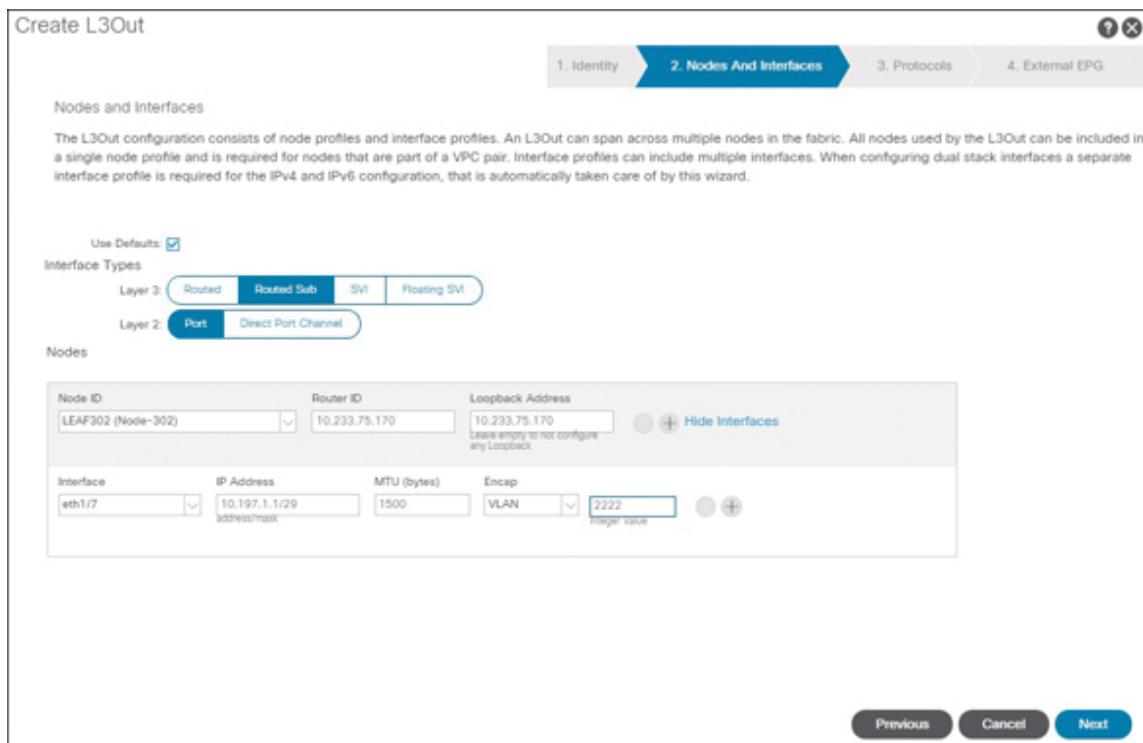
**Figure 9-54** shows that the first step in creating a BGP L3Out via the GUI is to enable the BGP checkbox. Notice that this does not disable the OSPF checkbox. Enter the L3Out name, VRF instance, and L3 domain and click Next.



**Figure 9-54** The L3Out Configuration Wizard Identity Page with BGP Enabled

**Figure 9-55** indicates that the Use Defaults checkbox has been enabled for this L3Out. Because of this checkbox, ACI does not show the logical node profile or logical interface profile names it intends to create. Even though there is little benefit in creating a loopback for EIGRP and OSPF L3Outs, the Loopback Address field in this case has been populated. This is recommended for BGP because a design change at

any time may require the addition of redundant multihop peerings. Enter interface configuration parameters and click Next to continue.



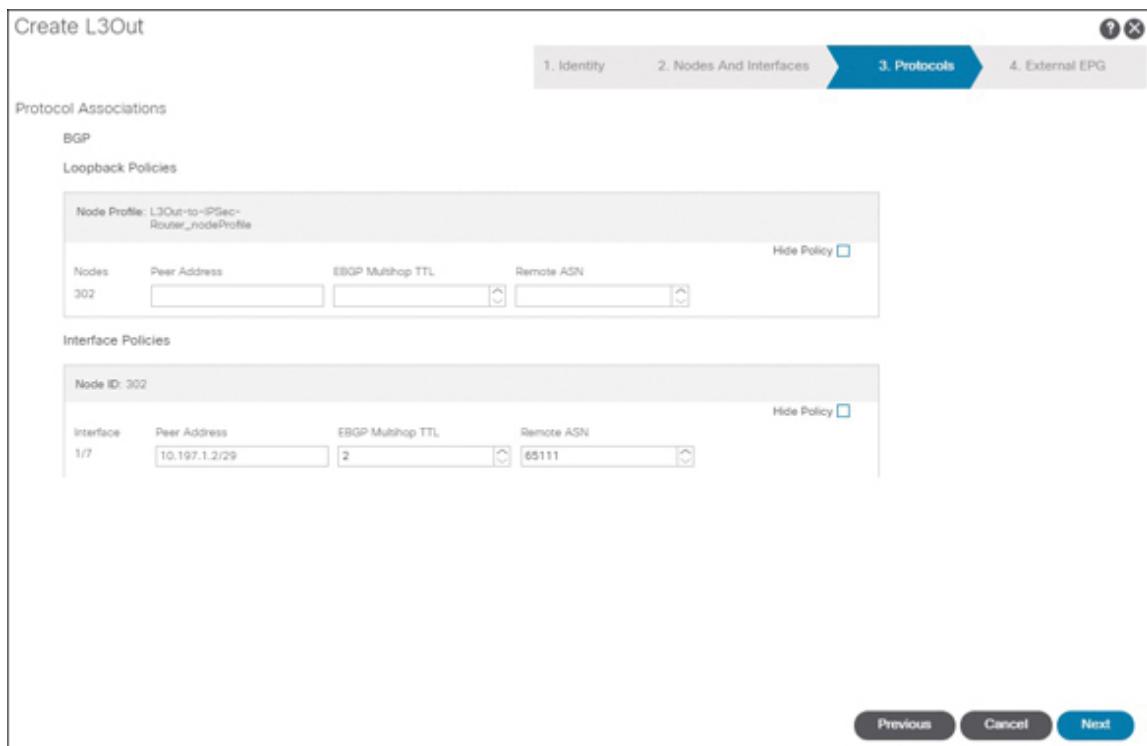
**Figure 9-55** Configuration Options on the Nodes and Interfaces Page

### Note

In L3Outs with interface type SVI, configuration of secondary addresses is an option. However, BGP sessions can only be sourced from the primary IP address of each interface.

**Key Topic**

For BGP L3Outs, the Protocol Association page is where all the action is. Each row completed on this page leads to the creation of a BGP peer connectivity profile. While BGP peer connectivity profiles contain many options, the wizard allows configuration of only the minimum number of required settings, which in the case of eBGP consist of the neighbor IP address (Peer Address parameter), the remote ASN, and the maximum number of hops to the intended peer (EBGP Multihop TTL parameter). BGP peers need to be defined either under logical interface profiles or under logical node profiles. When the intent is to source a BGP session from the node loopback interface, you can configure the BGP peer under a logical node profile. The first row in [Figure 9-56](#) represents a configuration that will be placed under the logical node profile. BGP peers whose sessions should be sourced from non-loopback interfaces need to be configured under a logical interface profile. The second row in this figure represents a configuration object that will be deployed under a logical interface profile sourcing the subinterface earlier defined for interface Ethernet1/7. Click Next to continue.



**Figure 9-56** Defining a BGP Peer with Session Sourced from a Routed Subinterface

### Note

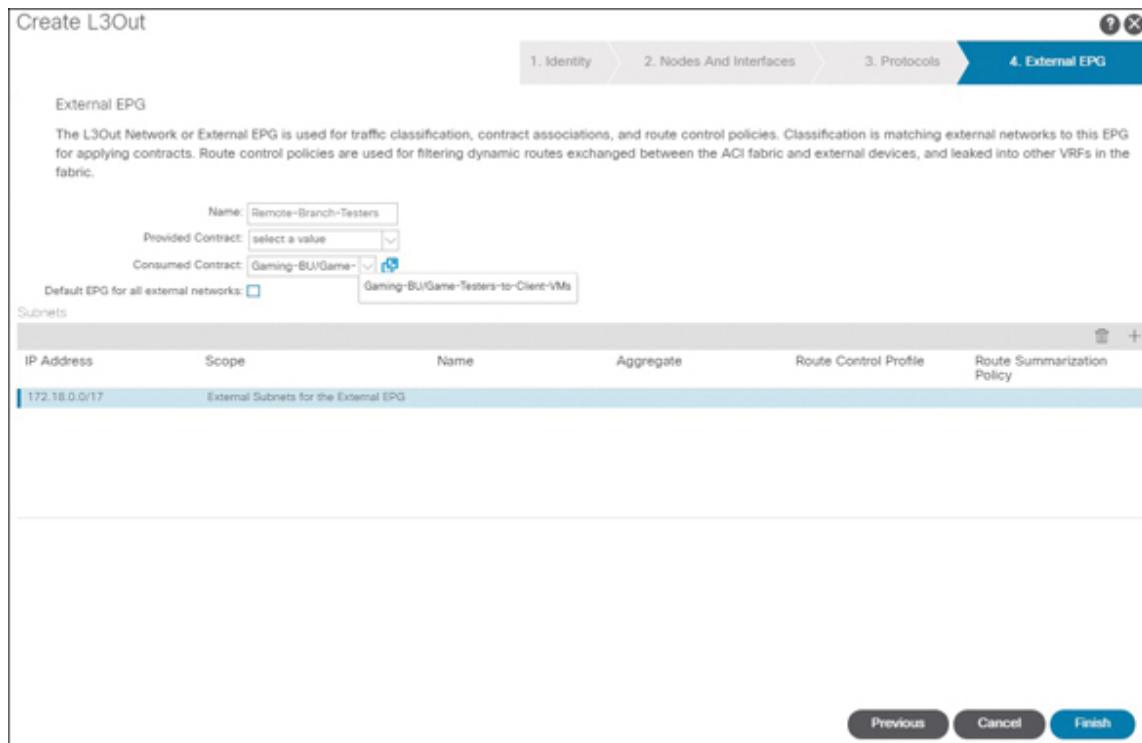
If you decide to populate only the Peer Address field, ACI builds a BGP peer connectivity profile suitable for a directly attached iBGP peer.

### Note

Instead of creating multiple BGP peer connectivity profiles for neighbors that are all in a single subnet and that require the same set of policies, you can use a feature called Dynamic Neighbor. With this feature, you can have ACI dynamically establish BGP peerings with multiple neighbors by configuring a subnet instead of an individual IP address in the Peer Address field. When BGP is configured with dynamic neighbor configuration,

ACI does not attempt to initiate sessions to IP addresses in the peer subnet. The other side needs to explicitly configure the ACI border leaf IP address to start the BGP session. The Dynamic Neighbor feature was called Prefix Peers in standalone NX-OS, and this term is also used in some Cisco ACI documentation.

Finally, [Figure 9-57](#) shows how to create a new external EPG straight from the L3Out creation wizard. In this case, the external EPG has the scope External Subnets for External EPG set, which means the external EPG can be used for classification of outside traffic. Because in this example remote testers need to log in to systems in EPG-Client-VMs, created in [Chapter 8](#), the external EPG is consuming a contract that grants such access. Click Finish to deploy the BGP L3Out. BGP peerings should come up if equivalent configuration has been deployed on the peer router.



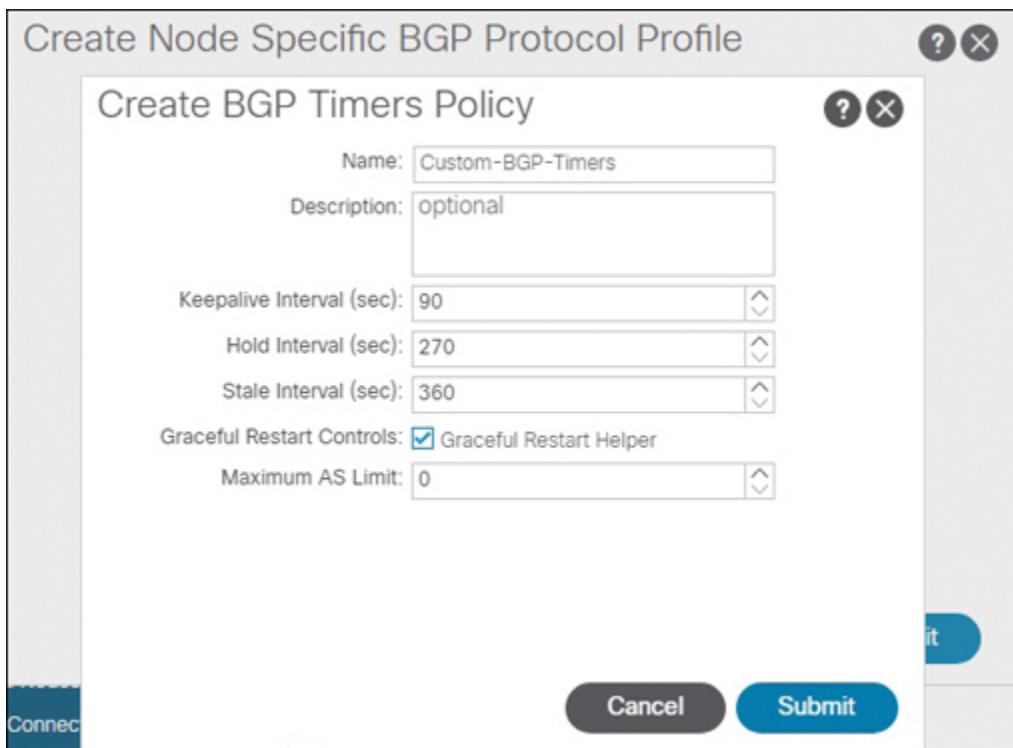
**Figure 9-57** *Creating an External EPG Classifying External Traffic in the L3Out Wizard*

## Implementing BGP Customizations at the Node Level

Once a BGP L3Out has been created, you may want to customize certain BGP settings at the border leaf level. Assuming that different logical node profiles have been used for each border leaf, different node-specific policies such as BGP timers can be applied to each switch.

To configure a node-specific BGP protocol profile, right-click the node profile of interest and select Create BGP Protocol Profile. The Create Node Specific BGP Protocol Profile page contains two drop-downs at the time of writing. One is for BGP timer customization, and the other is for AS\_PATH policy customizations. Recent versions of ACI code no longer support AS\_PATH policy customizations to enable ECMP across eBGP peers in different AS\_PATHs; therefore, this second set of policies is not addressed in this book. [Figure 9-58](#) shows the creation of a custom BGP timer policy.





**Figure 9-58** Creating a Custom BGP Timer Policy

None of the settings in Figure 9-58 should be new for you if you deal with BGP on a regular basis, but Table 9-4 provides a review of the options.

**Table 9-4** Configuration Parameters for BGP Timer Policies

## **Co Description**

n  
fi  
gu  
ra  
ti  
on  
Pa  
ra  
m  
et  
er

Ke ep ali ve Int er va l (s ec )	Specifies the interval at which keepalive messages are sent after a BGP peer is established. The default value for this setting is 60 seconds.
--	--

Ho Id Int er va l (s ec )	<p>Specifies the interval at which keepalive messages must be received for ACI to consider the BGP peer operational. The default value for this setting is 180 seconds.</p>
St al e Int er va l (s ec )	<p>Specifies when to delete stale routes in case a session is not reestablished within the established interval. When a graceful restart is in progress, the routes previously received from the peer are still used for forwarding but marked as stale. Once the session between two routers is reestablished and route information is synced again, all the stale routes are deleted and the routes from the latest exchange are used. This interval is applied locally. The default value is 300 seconds.</p>

Gr ac ef ul Re st art He Ip er	<p>Specifies the restarting router triggered when a graceful restart is in progress. The peer is likely simply helping the graceful restart operation with the restarting router.</p> <p>Cisco ACI provides only graceful restart helper capability because ACI does not support stateful supervisor switchover within each individual switch node. Only a cold reboot is available. Instead, routing protocol high availability (HA) should be achieved by using multiple switch nodes.</p>
M ax im u m AS Li mi t	<p>When nonzero, prompts ACI to discard eBGP routes received if the number of AS_PATH segments exceeds the stated limit. The default value of zero implies no maximum AS limit.</p>

Note that configured intervals take effect only after a new BGP session is established.



BGP timer policies can also be applied at the VRF level.

## Implementing Per-Neighbor BGP Customizations

**Figure 9-59** shows the majority of settings that can be customized on a per-BGP-peer basis. Two additional settings, Admin State and Route Control Profile, have been left out of this screenshot. Depicted customizations include modification of the Local-AS Number setting 65600 for establishing an eBGP session to a router in remote ASN 65700. The Local-AS Number Config parameter has been left blank, prompting ACI to advertise routes to this external neighbor by appending 65600 to the fabric route reflector ASN.

**Figure 9-59** Customization of the Local AS Number for a Specific BGP Peer

**Table 9-5** describes the customization options available in BGP peer connectivity profiles.

**Table 9-5** Configuration Parameters in BGP Peer Connectivity Profiles

Co Description	
nfifowSelffAS	All Allows ACI to receive routes from eBGP neighbors when the routes have the ACI BGP AS number in the AS_PATH. This option is valid only for eBGP peers.
ASoverride	Allows ACI to overwrite a remote AS in the AS_PATH with the ACI BGP AS. This is typically used when performing Transit Routing from an eBGP L3Out to another eBGP L3Out with the same AS number. Otherwise, an eBGP peer device may not accept the route from ACI because of AS_PATH loop prevention. When this option is enabled, Disable Peer AS Check also needs to be enabled. This option is valid only for eBGP peers.

Dis abl e Pe er AS Ch ec k	Allows ACI to advertise a route to the eBGP peer even if the most recent AS in the <code>AS_PATH</code> of the route is the same as the remote AS for the eBGP peer. This option is valid only for eBGP peers.
Ne xt- ho p Sel f	Allows ACI to update the next-hop address when advertising a route from an eBGP peer to an iBGP peer. By default, route advertisement between iBGP peers keeps the original next-hop address of the route, and the one between eBGP peers always updates the next-hop address with a self IP address.
Se nd Co m m uni ty	When enabled, allows ACI L3Out to advertise routes with a BGP Community attribute, such as <code>AS2:NN</code> format. Otherwise, the BGP Community attribute is stripped when routes are advertised to the outside.

Send Extended Community	<p>When enabled, allows ACI L3Out to advertise routes along with the BGP Extended Community attribute, such as RT:AS2:NN, RT:AS4:NN, and so on. Otherwise, the BGP Extended Community attribute is stripped when routes are advertised to the outside.</p>
Configure Password	<p>When configured, allows the BGP peering to use MD5 authentication on the BGP TCP session. The password can be reset by right-clicking the BGP peer connectivity profile and selecting Reset Password.</p>

All ow ed Sel f AS Co un t	Sets the maximum count for the Allow Self AS option under BGP controls.
Bid ire cti on al For wa rdi ng De tec tio n	Enables BFD on the BGP neighbor.

Dis abl e Co nn ect ed Ch ec k	<p>Provides an alternative to increasing the eBGP multihop TTL in cases where there is a security concern about increasing TTL unnecessarily. For eBGP peering, BGP checks whether the neighbor IP is on the same subnet as any of its local interfaces to see if the neighbor IP is directly connected. If it is not, BGP automatically assumes that the TTL needs to be larger than 1. Hence, when BGP is peering via loopbacks with directly connected routers, the BGP peering is rejected without the eBGP Multihop TTL being set to 2 or larger, even though TTL 1 is technically enough.</p>
Weigh for ro ut es fro m thi s nei ghbo r	<p>Sets the default value of a Cisco proprietary BGP path attribute weight on all the routes learned from the border leaf by the configured peer.</p>

Remove Private AS	In outgoing eBGP route updates to this neighbor, removes all private AS numbers from the AS_PATH when the AS_PATH has only private AS numbers. This option is not applied if the neighbor remote AS is in the AS_PATH.
Remove All Private AS	In outgoing eBGP route updates to this neighbor, removes all private AS numbers from the AS_PATH, regardless of whether a public AS number is included in the AS_PATH. This feature does not apply if the neighbor remote AS is in the AS_PATH. To enable this option, Remove Private AS needs to be enabled.
Replace All Private AS with Local AS	In outgoing eBGP route updates to this neighbor, replaces all private AS numbers in the AS_PATH with ACI local AS, regardless of whether a public AS or the neighbor remote AS is included in the AS_PATH. To enable this option, Remove All Private AS needs to be enabled.

BG P Pe Pre fix Pol icy	Defines an action to take when the number of received prefixes from this neighbor exceeds the configured maximum number. This option is activated by attaching a BGP peer prefix policy to the BGP peer connectivity profile.
Lo cal - AS Nu m be r	Disguises the ACI BGP ASN with the configured local ASN to peer with a particular neighbor. When this feature is used, it looks like there is one more ASN (local AS) between the ACI BGP AS and the external neighbor. Hence, the neighbor peers with the configured local ASN instead of the real ACI BGP ASN. In such situations, both the local ASN and the real ACI BGP ASN are added to the AS_PATH of routes advertised to the neighbor. The local ASN is also prepended to routes learned from the neighbor.

Local AS Number Configuration	<p>Allows granular control over how the local ASN and the fabric ASN appear in the AS_PATHs of routes advertised to external routers or received by the fabric.</p> <p>The no-prepend option prevents ACI from prepending the local ASN in the AS_PATHs of routes learned from this neighbor.</p> <p>The no-prepend, replace-as option allows ACI to add only a local ASN, instead of both a local ASN and a real ACI BGP ASN, to the AS_PATHs of routes advertised to this neighbor on top of the no-prepend option effect.</p> <p>The no-prepend, replace-as, dual-as option allows the neighbor to peer with both a local ASN and a real ACI BGP ASN on top of the no-prepend and replace-as option effect.</p>
Administrative State	Enables a BGP session with a peer to be turned off or on.

Route Control Profile	Allows application of a route profile to a specific BGP neighbor.
-----------------------	---

## Implementing BFD on a BGP L3Out

One of the most common tweaks in BGP peer connectivity profiles is to implement BFD with neighbors. BFD is not supported on loopback interfaces since there is no support for BFD multihop in ACI at the time of writing. BFD is also not supported for dynamic neighbors (prefix peers).

To enable BFD on a BGP L3Out, navigate to the desired BGP peer connectivity profile and enable the Bidirectional Forwarding Detection checkbox, which is visible in the Peer Controls section of [Figure 9-59](#), shown earlier.

Customization of BFD timers on a BGP L3Out requires application of a custom BFD policy and BFD interface profile. To apply a previously created BFD interface policy or to create a new one, right-click the desired logical interface profile under the L3Out and select Create BFD Interface Profile. Then select or create a BFD interface policy.

## Implementing BGP Customizations at the VRF Level

Aside from assigning a BGP timer policy at the VRF level, you can create a custom BGP address family context policy for VRF-wide application to the IPv4 unicast address family or the IPv6 unicast address family. [Figure 9-60](#) shows options that can be tweaked for a BGP address family context policy.

Create BGP Address Family Context Policy

Name: BGP-Policy-for-IPv4

Description: optional

eBGP Distance: 20

iBGP Distance: 200

Local Distance: 220

eBGP Max ECMP: 16

iBGP Max ECMP: 16

Enable Host Route Leak:

Cancel Submit

The dialog box is titled "Create BGP Address Family Context Policy". It contains several input fields: "Name" set to "BGP-Policy-for-IPv4", "Description" set to "optional", and numerical spinners for "eBGP Distance" (20), "iBGP Distance" (200), "Local Distance" (220), "eBGP Max ECMP" (16), and "iBGP Max ECMP" (16). Below these is a checkbox labeled "Enable Host Route Leak" which is unchecked. At the bottom right are two buttons: "Cancel" and "Submit", with "Submit" being highlighted in blue.

**Figure 9-60** Creating a BGP Address Family Context Policy

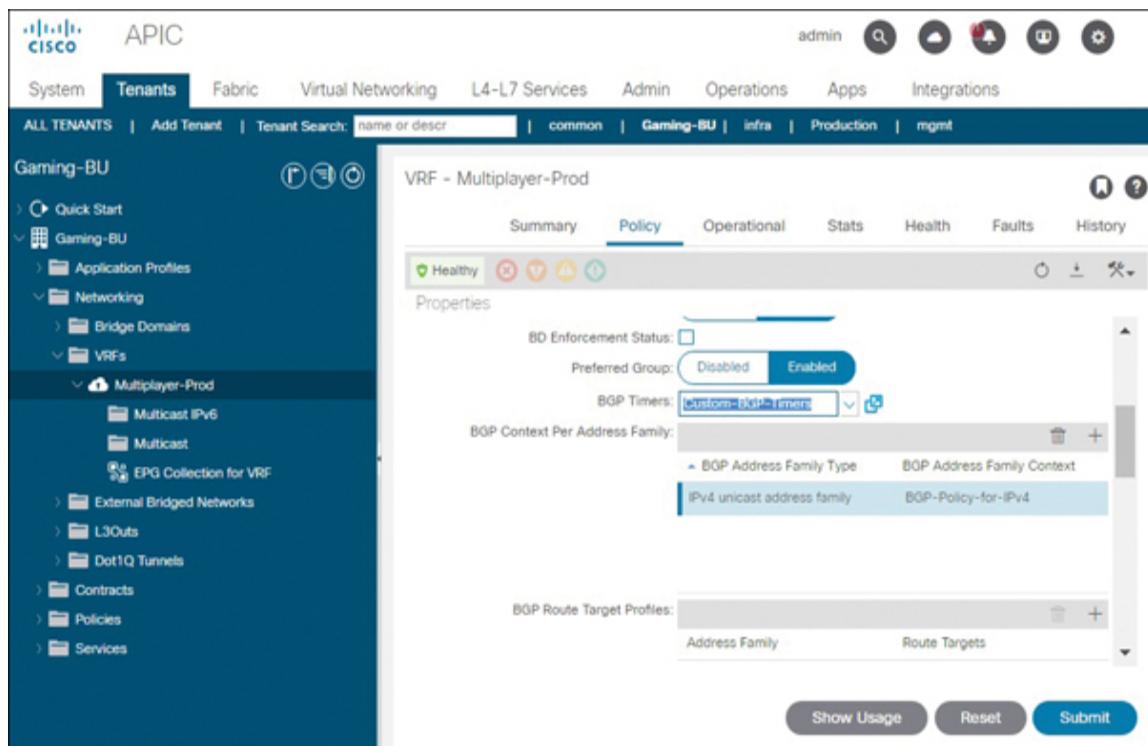
[Table 9-6](#) explains these configuration options.

**Table 9-6** Configuration Parameters for BGP Timer Policies

Config Description	
Parameter	Description
eBGP Distance	Specifies the administrative distance for eBGP-learned routes. The default AD for such routes is 20.
iBGP Distance	Specifies the administrative distance for eBGP-learned routes. The default AD for such routes is 200.
Local Distance	Is used for aggregate discard routes. The default AD for such routes is 220.
eBGP/iBGP Max ECMP	Configures the maximum number of equal-cost paths a switch adds to the routing table for eBGP-learned and iBGP-learned routes. The default value in ACI for this setting is 16.

Enable Host Route Leak	Is used only for the GOLF feature and is therefore beyond the scope of the DCACI 300-620 exam.
------------------------	--

Once it is configured, a BGP address family context policy needs to be applied to an address family for the policy to take effect. [Figure 9-61](#) shows a BGP timer policy being applied VRF-wide to Multiplayer-Prod and a BGP address family context policy being applied to the IPv4 unicast address family for the VRF.



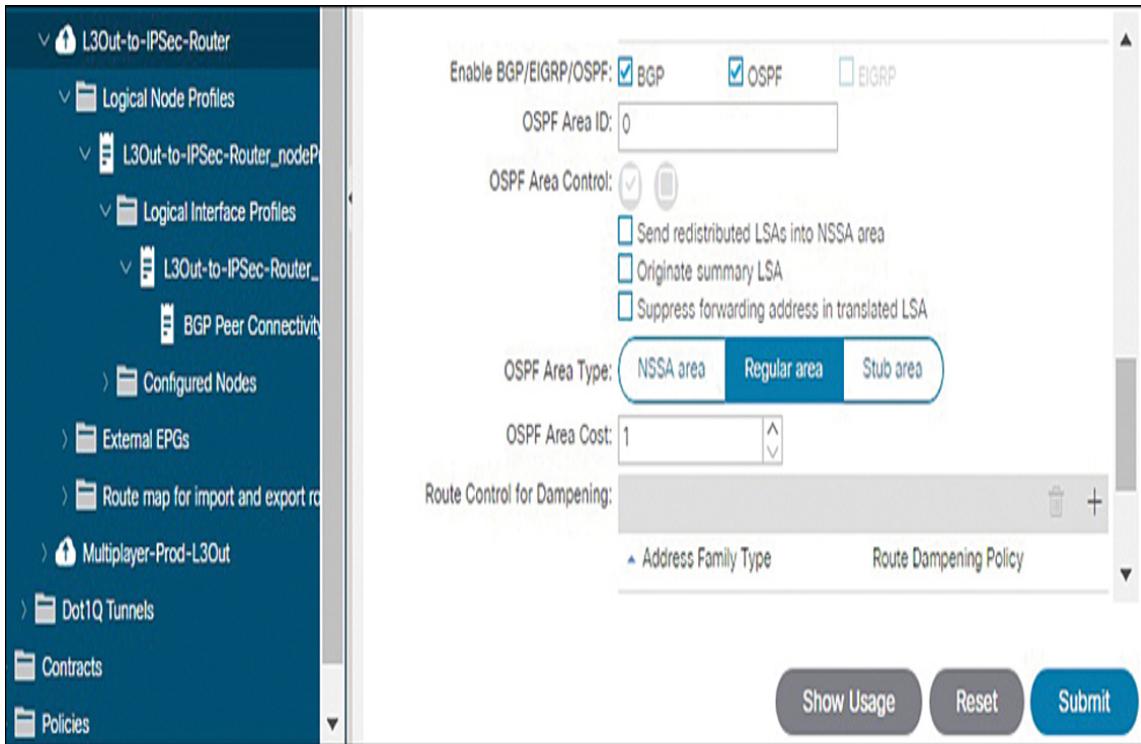
**Figure 9-61** Applying BGP Policies at the VRF Level

### Note

Although it is not impossible, it *is* improbable that DCACI candidates will need to know the name of a setting that modifies the administrative distance of routes learned by a specific routing protocol. DCACI 300-620 exam candidates should, however, know that enforcement of policies around protocol timers and administrative distances often take place at the VRF level.

## Implementing OSPF for IP Reachability on a BGP L3Out

To use OSPF for dynamic IP reachability advertisement on a BGP L3Out, select the root of the L3Out, click the Policy menu, and navigate to the Main submenu. Toward the bottom of the page, you see options for enabling OSPF. The available configuration options, as indicated in [Figure 9-62](#), are exactly the same as those available for OSPF L3Outs. But when OSPF and BGP are enabled side-by-side in the same L3Out, OSPF is programmed only to advertise its L3Out loopback and interfaces IP addresses.



**Figure 9-62 Enabling OSPF to Advertise IP Reachability for BGP Sessions**

## Implementing Hot Standby Router Protocol (HSRP)

ACI supports HSRP on L3Out routed interfaces and routed subinterfaces. When deploying HSRP, external devices must provide Layer 2 connectivity between the ACI border leaf switches that run HSRP. This allows the exchange of Hello messages over external Layer 2 connections. HSRP Hello messages do not pass through spine switches.

To configure HSRP in ACI, navigate to the logical interface profile of interest and select the Create HSRP Interface Profile checkbox. The process of configuring an HSRP interface policy should be straightforward for those familiar with HSRP. Note that the HSRP virtual IP address must be in the same subnet as the interface IP address.

Because SVIs on L3Outs can leverage secondary addresses, use of HSRP in ACI L3Outs for connectivity to appliances that only support static routing is not common. In such cases, static routes on appliances should point to the secondary addresses configured in the L3Out to enable next-hop high availability.

## **IPv6 and OSPFv3 Support**

IPv6 is not discussed in this book because ACI attempts to make the transition to IPv6 natural by not delineating between IPv6 and IPv4.

To enable OSPFv3 on an L3Out, enable the OSPF checkbox on the L3Out Main page and assign an IPv6 address to the L3Out and witness the border leaf bring up the OSPFv3 process.

## **Implementing Route Control**

Sometimes it may be necessary to filter certain routes or make changes to metrics and other attributes of routes advertised out of a fabric or learned by a fabric. Where granular route control is a requirement, ACI route profiles can help.

## **Route Profile Basics**

As you have seen, ACI creates route maps behind the scenes to accomplish certain tasks. For example, when an administrator associates a bridge domain with an L3Out and updates the scope of a BD subnet to Advertised Externally, ACI creates a route map to redistribute the subnet out the specified L3Out.

ACI uses route maps internally for quite a few different purposes, such as infra MP-BGP route distribution, BD subnet advertisement to the outside world, and transit routing.

**Route profiles** give users the ability to add user-defined match rules or set rules for route filtering or route manipulation. Route profiles can alternatively be referred to as *route control profiles*. The term *route map* is also sometimes used interchangeably with *route profile*, but in terms of implementation, each route profile is more a collection of implicit and explicit route maps.

Some of the use cases for this feature that are inside the scope of the DCACI 300-620 exam are as follows:

- A route profile to allow advertisement (export) of BD subnets to the outside world via L3Outs
- A route profile to limit learning (importing) of external routes from the outside world via L3Outs
- A route profile to set certain attributes on routes exiting or entering the fabric if they meet a certain criterion

A route profile can be associated with any of the following objects/components:

- A bridge domain
- A bridge domain subnet
- An EPG on an L3Out
- A subnet of an EPG on an L3Out
- A route peering neighbor
- An entire L3Out

**Table 9-7** defines the various components of a route profile in ACI.



**Table 9-7** Components of a Route Profile

C o m p o n e n t	D e s c r i p t i o n
Type	The route profile type is specific to ACI. There are two route profile types. One type, Match Prefix AND Routing Policy, combines prefixes from the component that the route profile is associated with and the match criteria configured in the route profile. Components that route profiles can be associated to include bridge domains, bridge domain subnets, L3Outs, L3Out EPGs, and L3Out EPG subnets. The other type, Match Routing Policy Only, only matches routes based on criteria configured in the route profile and ignores prefixes from the components with which the route profile is associated.

C In a sense, each entry in a route profile includes two context options: Order and Action. Order is equivalent to a sequence number in a normal route map with the caveat that some route profiles merge internal route maps of components with statements explicitly entered by administrators, changing the actual applicable sequence of rules. Action consists of permit or deny and is equivalent in function to permit or deny in a normal route map.

M Route profile match rules are similar to match clauses in route maps. Clauses ACI can match against include prefixes, community attributes, and regular expressions.

S Set rules are equivalent to set clauses in a route map. ACI can set parameters such as community attributes, weight, OSPF types, and AS\_PATH.

The next few subsections provide examples of route profile implementation.

### Note

The coverage of route profiles in this book is just the tip of the iceberg when it comes to route control in ACI.

The content and examples in this chapter reflect the scope of the DCACI 300-620 exam. Where deploying these solutions in environments with route leaking or transit routing, additional caveats may apply.

## Modifying Route Attributes to All Peers Behind an L3Out

One of the most prevalent use cases for route maps in traditional routers and Layer 3 switches is to manipulate route metrics or to add attributes to routes. Let's take a look at an example of how route profiles address this use case. Say that a fabric has multiple L3Outs, and an IT team wants to assign different BGP communities to routes advertised out of each L3Out. The idea is that the BGP communities could then be used by the external devices for traffic engineering or any type of policy enforcement.

To address any route profile requirement, the implementation team needs to fully understand the required match rules and set rules. Since the idea is that the interesting prefixes are any ACI routes egressing a specific L3Out, and routes advertised have already been marked with the scope Advertised Externally and have been associated with the relevant L3Out, there should be an easy way to match such prefixes. It turns out there is. Application of a route profile at the default export level of an L3Out

automatically satisfies this requirement without the need for Match statements.

The next part of the equation is configuration of set rules. To configure a set rule, go to the tenant in question, double-click Policies, double-click Protocol, right-click Set Rules, and select Create Set Rules for a Route Map. [Figure 9-63](#) shows a set rule that assigns BGP community 65000:100 to any matched routes. Note also the other potential options for set rules.

Create Set Rules for a Route Map

STEP 1 > Select

Name: Set-BGP-Community-65000:100

Description: optional

Set Community:

Criteria: Append community

Community: regular:as2-nn2:65000:100

e.g., regular:as2-nn2:4:15  
e.g., extended:as4-nn2:5:16  
e.g., no-export  
e.g., no-advertise

Set Route Tag:

Set Dampening:

Set Weight:

Set Next Hop:

Set Preference:

Set Metric:

Set Metric Type:

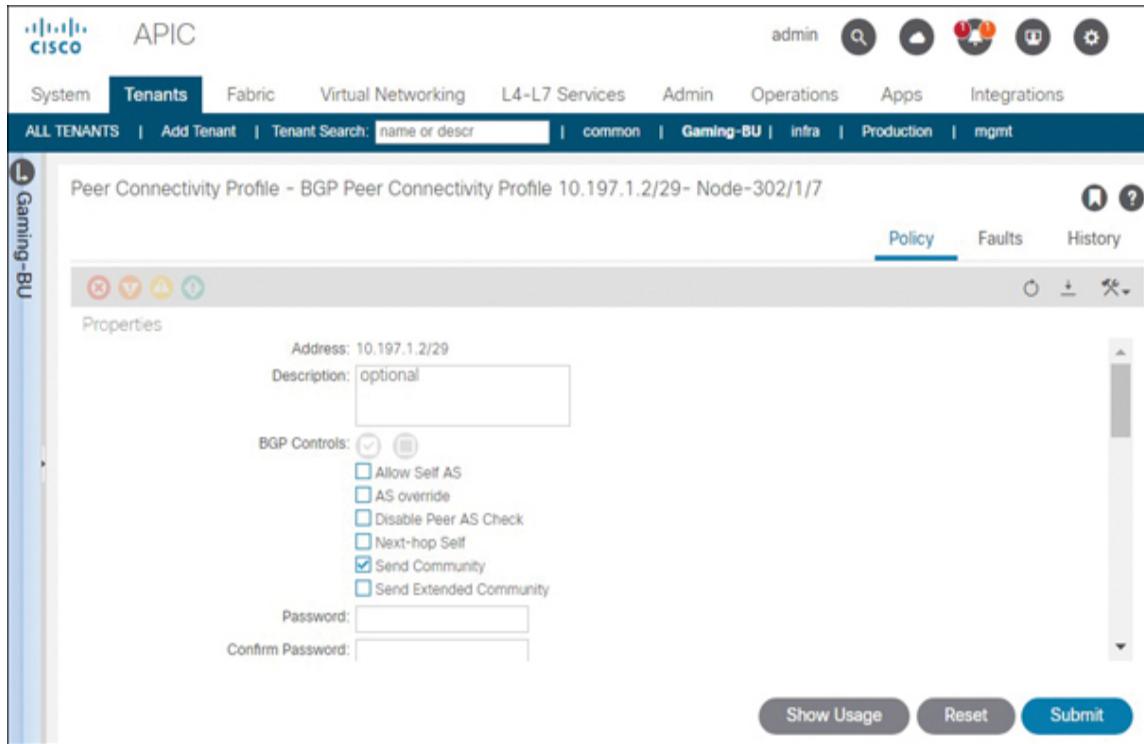
Additional Communities:

Set AS Path:

1. Select

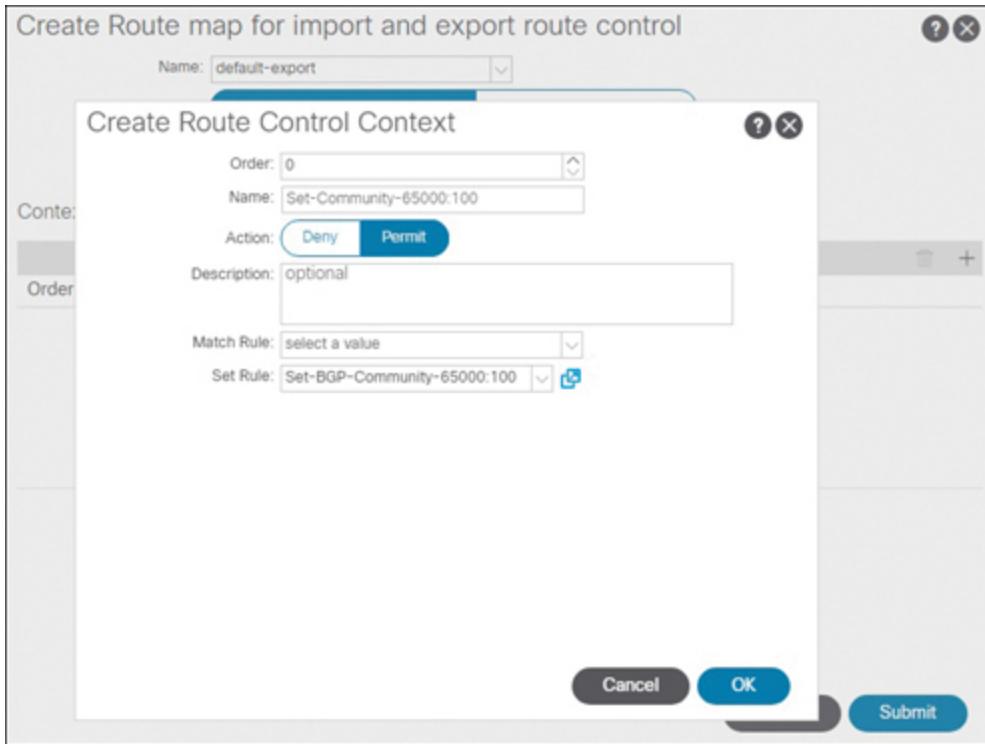
**Figure 9-63** Creating Set Rules for a Route Profile

By default, ACI strips BGP communities in route advertisements to external peers. To address this, as shown in [Figure 9-64](#), you can enable the Send Community checkbox in the BGP peer connectivity profile for the single BGP neighbor currently attached to the L3Out. If advertising extended BGP communities, you need to enable the Send Extended Community.



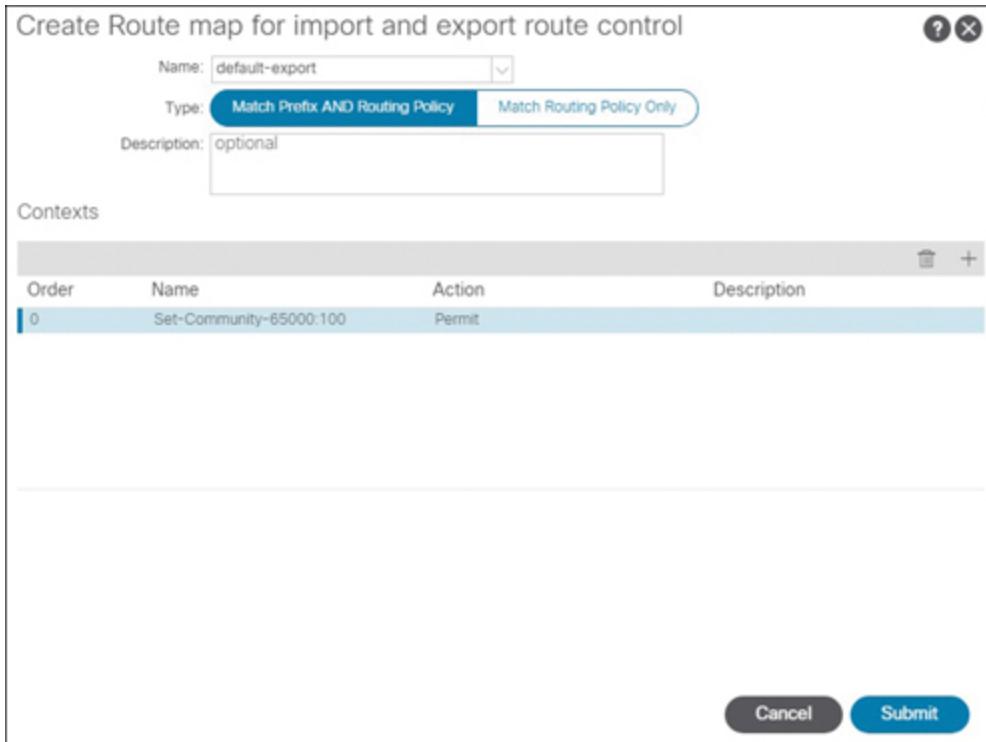
**Figure 9-64** Enabling Community Value Advertisements to a Neighboring Router

The next step is to actually configure the route profile on the L3Out. At a minimum, each route profile requires at least one context. Each context is similar to a sequence number in a traditional route map. As indicated in [Figure 9-65](#), creation of a context requires entry of a numeric value indicating the priority or order of the context, whether to permit (advertise) or deny (drop) the route, any match criteria (Match Rule drop-down) to determine prefixes of interest, and any modifications (Set Rule drop-down) ACI should apply to matched routes.



**Figure 9-65** Creating a Context Within a Route Profile

Note in [Figure 9-65](#) that a match rule, in this instance, was not required. This is specifically because the route profile type Match Prefix AND Routing Policy has been selected, as indicated in [Figure 9-66](#). The term Match Prefix Policy can be understood as a directive for ACI to consider any prefixes associated with the object(s) to which the route profile is associated to be implicit match rules. In this case, the default-export route profile is associated with the entire L3Out. Therefore, Match Prefix Policy implicitly matches all BD subnets associated with the L3Out with the scope Advertised Externally, even though there is no explicit match statement in the route profile. You can read the term Match Routing Policy to refer to the type of explicit match statements engineers are used to adding to route maps. The phrase Match Prefix AND Routing Policy, therefore, is a merging of the implicit route map(s) and match rules explicitly configured in the route profile.



**Figure 9-66** Setting Route Profile Type to Match Prefix AND Routing Policy

Once these settings are implemented, all routes out of the specified L3Out should be advertised out to all peers behind the L3Out using the community 65000:100, as shown in Example 9-7.

### Example 9-7 Validating Community Values on a Route on Nexus Platforms

[Click here to view code image](#)

```
Router# show ip bgp 10.233.58.0/24 vrf LAB-BGP
(...output truncated for brevity...)
BGP routing table information for VRF LAB-BGP, address family
IPv4 Unicast
BGP routing table entry for 10.233.58.0/24, version 14
Paths: (1 available, best #1)
AS-Path: 65000 , path sourced external to AS
```

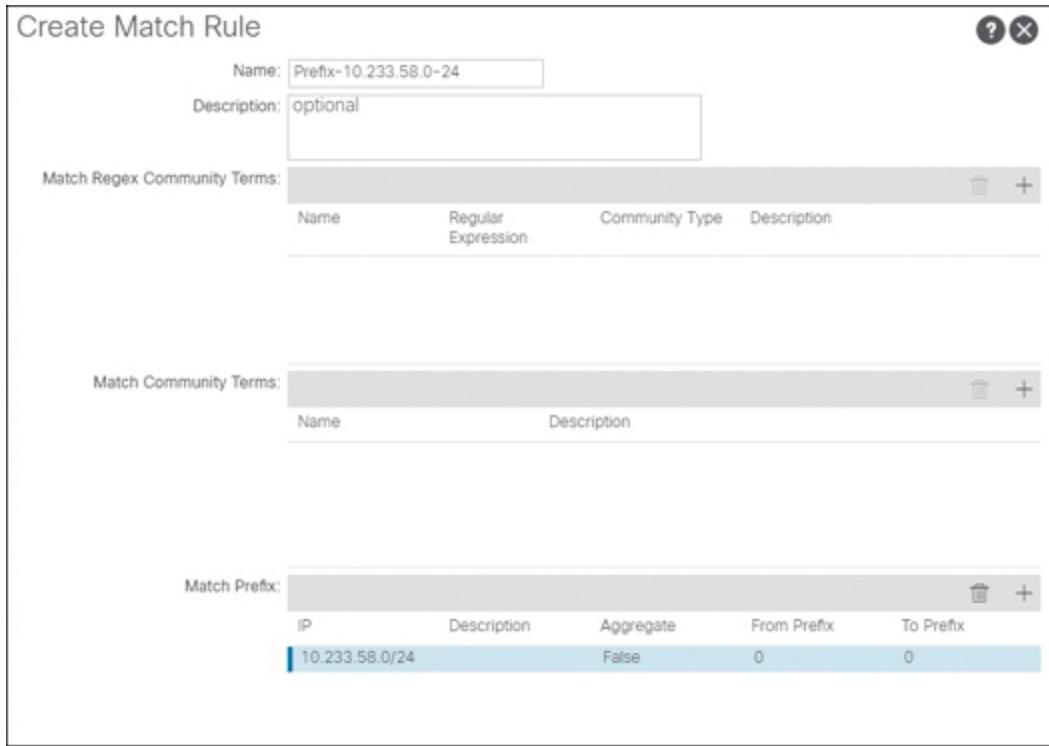
```
10.197.1.1 (metric 0) from 10.197.1.1 (10.233.75.170)
  Origin incomplete, MED 0, localpref 100, weight 0
  Community: 65000:100
Path-id 1 not advertised to any peer
```

Earlier in this example configuration, the text indicated that the Deny action drops matched routes and does not advertise them out the L3Out. This is not always true, especially when using route profiles of the type Match Prefix AND Routing Policy. Because this particular type of route profile merges multiple route maps together, it can sometimes be difficult to determine whether an implicit route map has already permitted advertisement of the prefix.

## Modifying Route Attributes to a Specific Peer Behind an L3Out

Now let's say that the objective is a little different from the objective we've been considering. Assume that multiple BGP peers have been deployed behind an L3Out, and different community values need to be applied to outbound routes to each peer behind the L3Out. In such a scenario, the default-export route profile would be of little help. Instead, route profiles would need to be applied to the BGP peer connectivity profiles. But, in addition, explicit match rules would be needed to specify the prefixes of interest. This is because BD subnets do not have a direct association with BGP peers.

[Figure 9-67](#) shows an explicit match statement for the subnet assigned to BD-Production. Note that the Aggregate flag has been set to False, indicating that this rule matches only 10.233.58.0/24 and not host routes within that range.

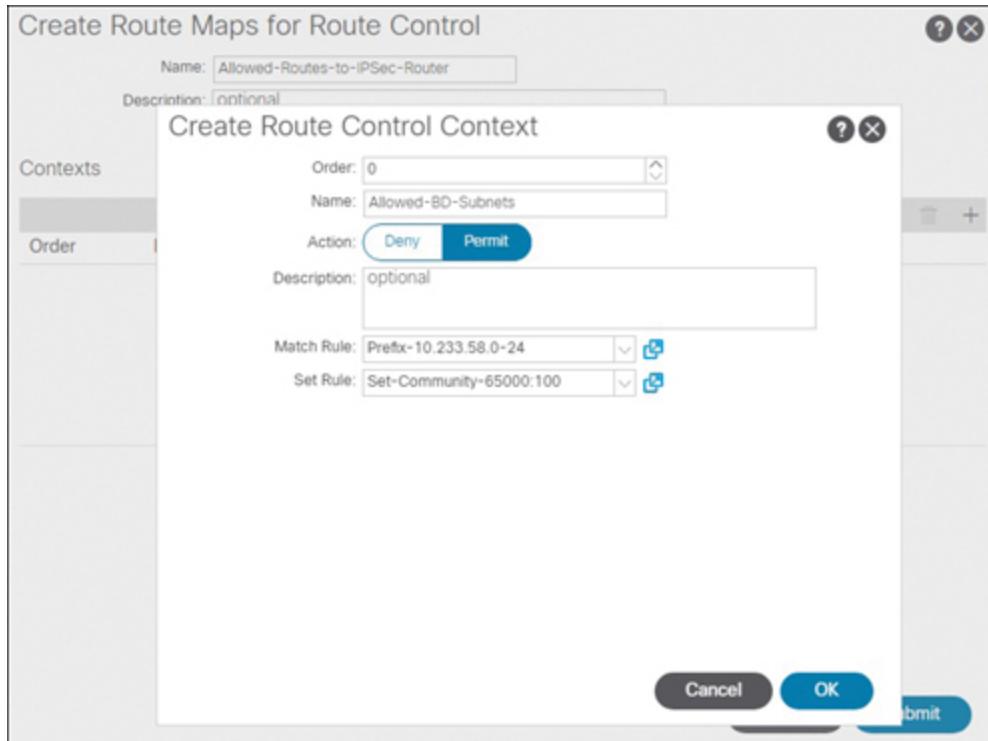


**Figure 9-67 An Example of a Match Rule Determining an Explicit Prefix as the Only Match**

If the expectation is that several conditions need to be met for a prefix to be matched, all the conditions can be specified under a single Match Rule object. For example, if the requirement for a match is for a prefix to be in the 10.5.0.0/16 range AND have a community value of 65600:100, these two conditions should both be included in a single match rule. On the other hand, if the goal is to say that either prefixes under 10.5.0.0/16 OR community value 65600:100 matches the criteria for advertisement with a particular set rule, separate route map sequences (separate context policies) should be configured for these two requirements, each referencing a different match rule.

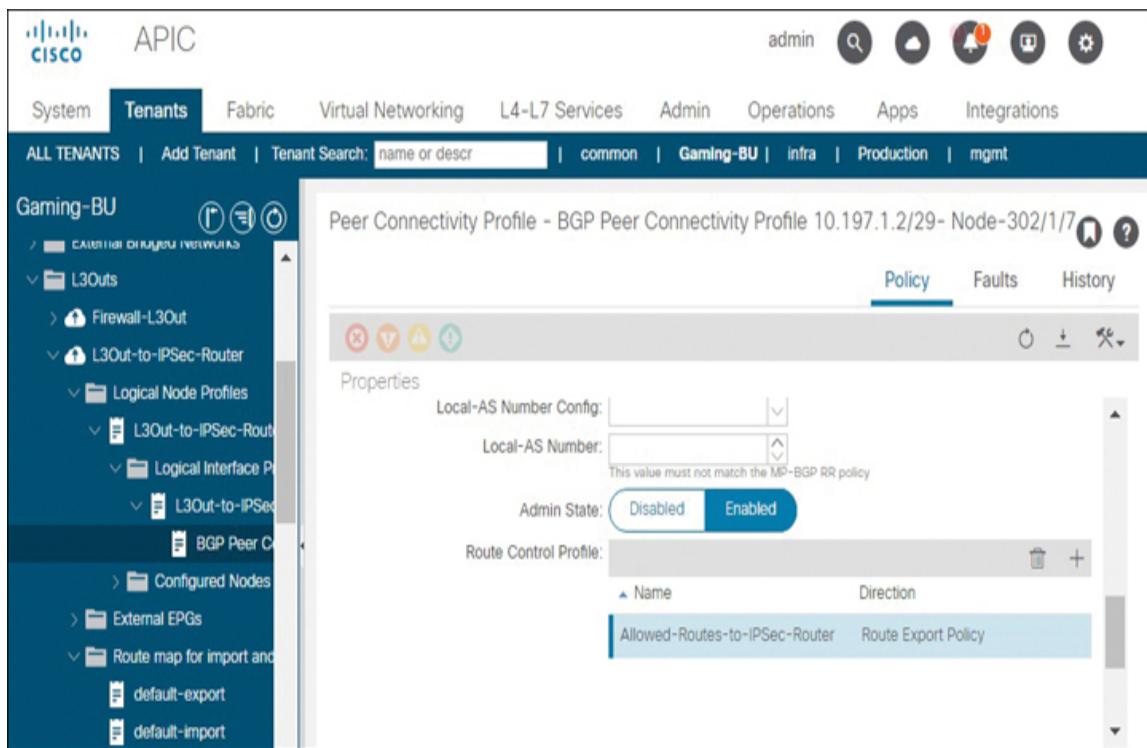
Once the match rule has been defined, a route profile needs to be created under **Tenants > the tenant in question > Policies > Protocol > Route Maps for Route Control**. As indicated in the background in [Figure 9-68](#), tenant-level

route profiles do not display the route profile type because route profiles defined here are for use cases such as MP-BGP interleak and BGP route dampening, which do not benefit from Match Prefix AND Routing Policy. Notice in [Figure 9-68](#) that both a match rule and a set rule have been assigned to the route profile context.



**Figure 9-68** Creating a Route Profile with an Explicit Prefix List Assigned

After the route profile has been created, it needs to be assigned to the desired BGP peer. [Figure 9-69](#) shows that the route has been assigned to a BGP peer connectivity profile in the export direction. In traditional route map terms, this should be understood as the *out* direction.



**Figure 9-69** Applying a Custom Route Profile in the Outbound (Export) Direction

With this change, the defined community values should only be applied to the one BGP peer.

One thing that might not be apparent from the configuration shown here is that the act of matching 10.233.58.0/24 in a custom outbound route profile or default-export route map is an alternative method to redistribute BD subnets outside the fabric. In other words, as long as BD subnet 10.233.58.0/24 has been marked Advertised Externally, this configuration would work to advertise the subnet out of the fabric, even without a BD association with the L3Out.

### Note

At the time of writing, there is no mechanism to assign a route profile on a per-neighbor basis for OSPF and EIGRP.

## Assigning Different Policies to Routes at the L3Out Level

Let's say there is a need to be more granular and assign different community values to routes as they leave the fabric on an L3Out. [Figure 9-70](#) shows that a new match rule and a new set rule have been associated with one another under a route profile context. Notice that the order 1 has been assigned, given that multiple requirements exist.

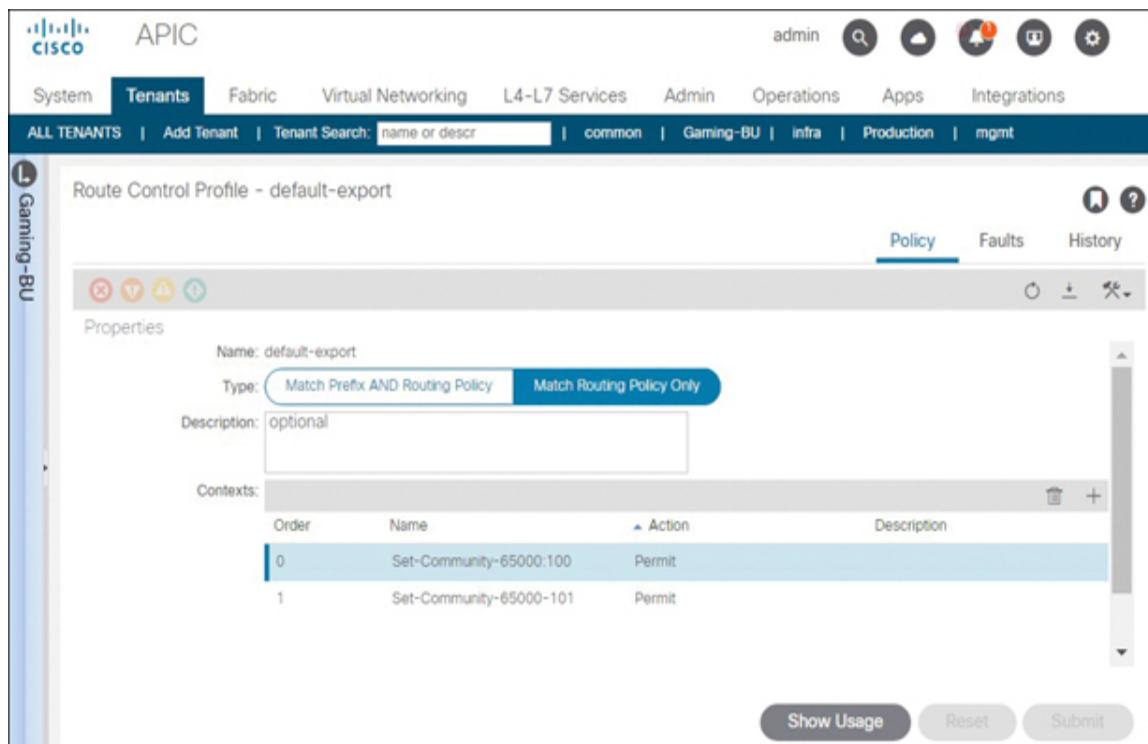
The screenshot shows a configuration interface for a 'Route Control Context'. The title bar says 'Create Route Control Context'. The form fields include:

- Order:** A dropdown menu set to '1'.
- Name:** A text input field containing 'Set-Community-65000-101'.
- Action:** A radio button group with 'Deny' and 'Permit' options; 'Permit' is selected.
- Description:** A text input field containing 'optional'.
- Set Rule:** A dropdown menu showing 'Set-Community-65000:101'.
- Associated Matched Rules:** A section with a 'Rule Name' dropdown menu showing 'Match-10.233.60.0-23'.
- Buttons:** 'Update' and 'Cancel' buttons at the bottom left, and 'Cancel' and 'Submit' buttons at the bottom right.

**Figure 9-70** Configuring Multiple Route Profile Contexts

[Figure 9-71](#) shows how contexts can be used together to match different subnets with community values or other set rules. One very important question here is whether it makes sense to use Match Prefix AND Routing Policy in instances like this. The problem with using this particular route profile type is that the merging of the explicit route profile rules

with implicit route maps may place implicit matches into the route map sequence specified with the order 0. This, in turn, could assign 65000:100 to all BD subnets, even though this might not be the intention. In any instance, where the intention is to only apply explicit policy, the route profile type can be toggled to Match Routing Policy Only. In this example, subnets in the 10.233.58.0/23 supernet range are assigned the BGP community 65000:100, while subnets in the 10.233.60.0/23 supernet range are assigned to 65000:101. Any subnets not in the ranges specified by match rules hit an implicit deny rule and are not advertised.



**Figure 9-71 Reliance on Explicit Match Rules Using Match Routing Policy Only**

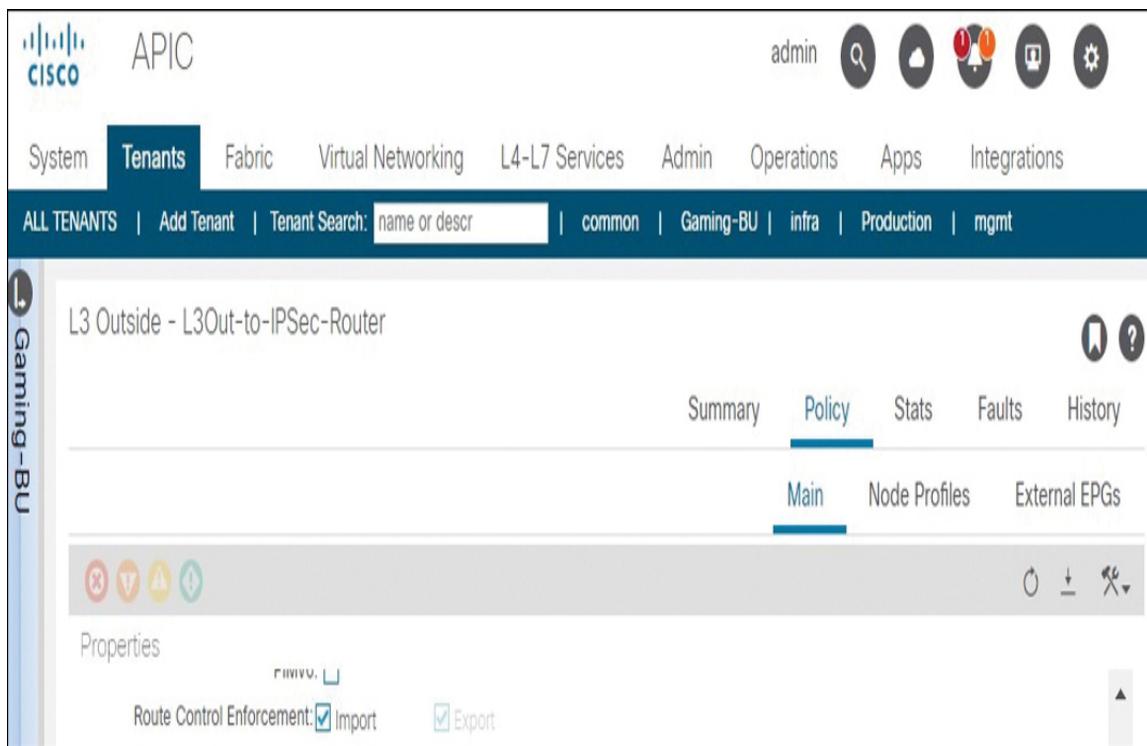
### Note

Although not apparent in this case, the policy type depicted here blocks exit of any routes not matched

with an explicit Permit action. Can this be useful if a border leaf needs to deploy OSPF and EIGRP L3Outs side-by-side within a VRF? If there is a requirement to advertise different subnets out of each L3Out and sidestep the issue discussed earlier in this chapter, the answer is yes. Use of a route profile with Match Routing Policy Only on one L3Out and leveraging regular BD subnet advertisement mechanisms for the second L3Out is a valid (but probably not recommended) way to overcome limitations related to OSPF and EIGRP L3Outs in the same VRF on the same border leaf advertising the same subnets due to application of common implicit route maps.

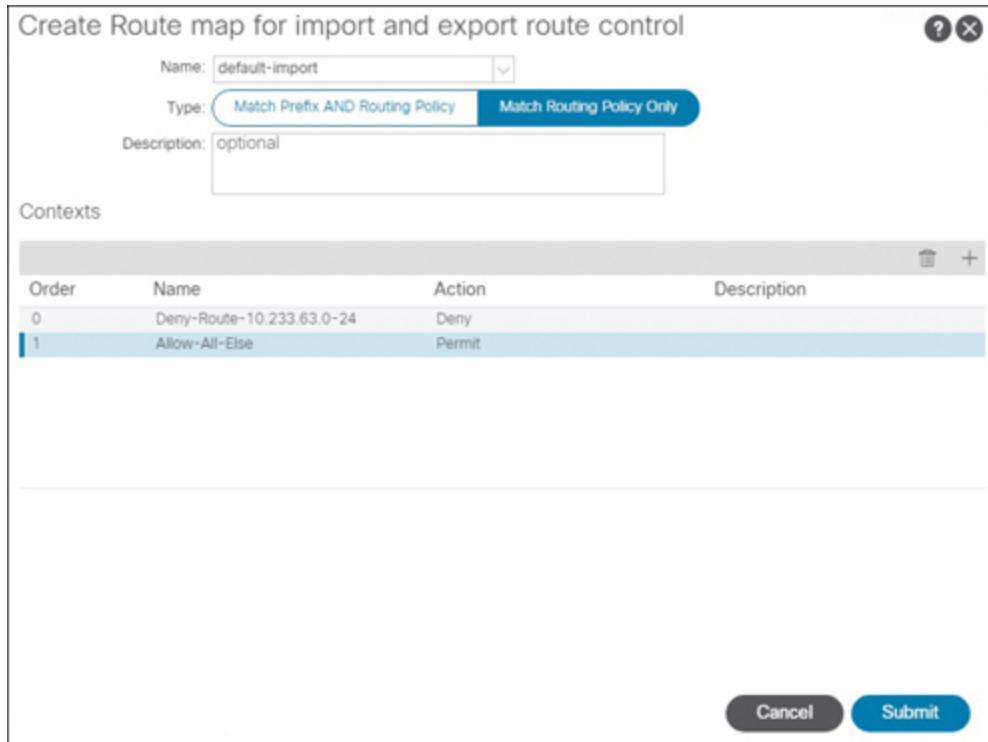
## Configuring Inbound Route Filtering in ACI

ACI learns all routes inbound by default. If this behavior is not desired, navigate to **Policy > Main** for an L3Out and enable the Route Control Enforcement Import checkbox, as shown in [Figure 9-72](#). This setting enables administrators to enforce inbound route profiles.



**Figure 9-72** Enabling Route Control in the Inbound Direction on an L3Out

As shown in [Figure 9-73](#), the default-import route profile on the L3Out can be used in conjunction with explicit prefix lists and match rules to determine what routes can be imported into the fabric.



**Figure 9-73** Creating a default-import Route Profile to Filter Specific Inbound Routes

### Note

The goal of this section on route profiles was not to provide step-by-step guidance on configuring route profiles but to demonstrate route profiles and route profile types in context. The coverage is by no means complete but shows that what is within the scope of DCACI is very limited.

## Exam Preparation Tasks

As mentioned in the section “How to Use This Book” in the Introduction, you have a couple of choices for exam preparation: [Chapter 17, “Final Preparation,”](#) and the exam

simulation questions in the Pearson Test Prep Software Online.

## Review All Key Topics

Review the most important topics in this chapter, noted with the Key Topic icon in the outer margin of the page. [Table 9-8](#) lists a reference of these key topics and the page numbers on which each is found.



**Table 9-8** Key Topics for [Chapter 9](#)

Key Topic Element	Description	Page Number
List	Details the five critical functions of an L3Out	<a href="#">293</a>
List	Describes the anatomy and important components of a typical L3Out	<a href="#">294</a>
List	Lists the types of interfaces that can be configured under an L3Out logical interface profile	<a href="#">296</a>

Key Topic	Description	Page Number
Element		
Paragraph	Describes why a port or port aggregation with an EPG mapping can no longer function as an L3Out routed interface or routed subinterface	296
Figure 9-5	Details the impact in terms of route peerings when using the same SVI encapsulation across interfaces than when different encapsulations are used	297
Paragraph	Describes ACI behavior when an SVI with a particular encapsulation is deployed on a second L3Out on a different border leaf switch	298
Paragraph	Describes the significance of the SVI Encap Scope setting	299
Figure 9-7	Compares use of SVI Encap Scope of a VRF versus Local on L3Outs on a particular leaf switch	299

Key Topic Element	Description	Page Number
Paragraph	Describes default ACI behavior with SVI Auto State set to Disabled	299
Figure 9-8	Illustrates the impact on static routes during failover with ACI Auto State set to Disabled	300
Figure 9-9	Illustrates the impact on static routes during failover with ACI Auto State set to Enabled	300
Paragraph	Details what happens if no BGP route reflector has been configured in an ACI fabric	304
List	Lists the two configuration parameters needed for implementing BGP route reflection in ACI	304
Figure 9-14	Shows configuration of BGP route reflection under the BGP route reflector policy object	304

Key Topic	Description	Page Number
Element		
Figure 9-17	Shows enablement of EIGRP and entry of autonomous system	307
Figure 9-18	Illustrates entry of node and interface information using routed interfaces	308
Figure 9-20	Shows the external EPG creation page of the L3Out wizard	309
Paragraph	Describes the significance of the External Subnets for External EPG subnet scope	311
Figure 9-25	Shows the General tab in an external EPG, which provides a quick view into subnets and scopes	312

Key Topic Element	Description	Page Number
Figure 9-26	Shows how to mark a BD as a candidate for subnet redistribution	314
Figure 9-27	Shows toggling a BD subnet with Advertised Externally	315
Figure 9-29	Illustrates a filter for communication over an L3Out	316
Figure 9-30	Illustrates a contract subject for filter	316
Paragraph	Explains contract directionality in the context of external EPGs	317

Key Topic Element	Description	Page Number
Paragraph	Describes the Advertise Host Routes setting	321
Paragraph	Describes implementation of BFD on an EIGRP L3Out	321
Table 9-2	Details customizable settings for EIGRP applied at the VRF level	324
Figure 9-44	Shows OSPF configuration options and area types supported in ACI	325
Paragraph	Describes a use case for secondary IP addresses on L3Out SVIs	325
Figure 9-45	Illustrates configuration of L3Out SVIs with secondary IP addresses	326

Key Topic Element	Description	Page Number
Paragraph	Describes some scope implications for external EPGs that classify traffic	327
Paragraph	Describes a common problem related to deployment of OSPF and EIGRP L3Outs side-by-side within a VRF on the same leaf switches	328
List	Calls out the most common and recommended solutions where OSPF and EIGRP need to be deployed on the same border leaf while advertising different subnets	328
Table 9-3	Details customizable parameters under OSPF timer policies	329
Paragraph	Explains that static routing L3Outs do not need to deploy any dynamic routing protocols	330

Key Topic Element	Description	Page Number
Paragraph	Explains the Preference field for static routes	330
Figure 9-49	Demonstrates adding a static route to an L3Out	330
List	Describes the process for implementing IP SLA tracking for static routes	330
Paragraph	Describes the various fields in the IP SLA configuration wizard	331
Paragraph	Describes the process of creating a track member	331
Paragraph	Describes the significance of configuration options for track lists	332

Key Topic Element	Description	Page Number
Paragraph	Explains the difference between deploying BGP configurations at the node profile level versus the interface profile level	336
Figure 9-58	Shows the custom BGP timer policy page	338
Paragraph	Explains that BGP timer policies can be applied not just at the node level but at the VRF level	339
Table 9-5	Defines configuration parameters in BGP peer connectivity profiles	339
Table 9-7	Describes the various configuration components that make up a route profile	345

## Complete Tables and Lists from Memory

Print a copy of [Appendix C, “Memory Tables”](#) (found on the companion website), or at least the section for this chapter, and complete the tables and lists from memory. [Appendix D, “Memory Tables Answer Key”](#) (also on the companion website), includes completed tables and lists you can use to check your work.

## Define Key Terms

Define the following key terms from this chapter and check your answers in the glossary:

- logical node profile
- logical interface profile
- floating SVI
- L3Out bridge domain (BD)
- route reflector
- interleak
- secondary IP address
- route profile

# Chapter 10

## Extending Layer 2 Outside ACI

This chapter covers the following topics:

**Understanding Network Migrations into ACI:** This section describes the network-centric approach to ACI and settings significant to network migrations.

**Implementing Layer 2 Connectivity to Non-ACI Switches:** This section covers the implementation of bridge domain and EPG extensions out of an ACI fabric.

**Understanding ACI Interaction with Spanning Tree Protocol:** This section addresses how ACI reacts when it receives Spanning Tree Protocol BPDUs from a traditional network.

This chapter covers the following exam topics:

- 1.6 Implement ACI logical constructs
  - 1.6.d bridge domain (unicast routing, Layer 2 unknown hardware proxy, ARP flooding)
  - 1.6.e endpoint groups (EPG)
  - 1.6.f contracts (filter, provider, consumer, reverse port filter, VRF enforced)
- 3.1 Implement Layer 2 out (STP/MCP basics)

In the current world of routed Clos fabrics, Layer 2 connectivity to traditional switches is often intended to be an interim state adopted either to enable workload migrations into the fabric or to prolong the life of non-supported hardware.

Sometimes Layer 2 connectivity to traditional switches is used to keep a specific device outside a fabric indefinitely. This is seldom an

approach engineers use to deploy high-availability appliances whose failover procedures or selected settings conflict with ACI endpoint learning and where there is no desire to mitigate the issue through IP learning customizations.

Because migrations into ACI fabrics are the most prominent use case for Layer 2 extension to non-ACI switches, this chapter first addresses bridge domain, EPG, and contract configuration settings significant to network migrations. Next, it details the implementation of the two current flavors of Layer 2 connectivity to traditional switches. Finally, it covers ACI interaction with Spanning Tree Protocol.

## “Do I Know This Already?” Quiz

The “Do I Know This Already?” quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the “Exam Preparation Tasks” section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. [Table 10-1](#) lists the major headings in this chapter and their corresponding “Do I Know This Already?” quiz questions. You can find the answers in [Appendix A, “Answers to the ‘Do I Know This Already?’ Questions.”](#)

**Table 10-1** “Do I Know This Already?” Section-to-Question Mapping

Foundation Topics Section	Questions
Understanding Network Migrations into ACI	1-4
Implementing Layer 2 Connectivity to Non-ACI Switches	5-8
Understanding ACI Interaction with Spanning Tree Protocol	9, 10

## Caution

The goal of self-assessment is to gauge your mastery of the topics in this chapter. If you do not know the answer to a question or are only partially sure of the answer, you should mark that question as wrong for purposes of the self-assessment. Giving yourself credit for an answer you correctly guess skews your self-assessment results and might provide you with a false sense of security.

- 1.** True or false: When trunking a VLAN to ACI and the default gateway is outside the fabric, it is best to set L2 Unknown Unicast to Hardware Proxy.
  - a.** True
  - b.** False
- 2.** An any-to-any contract that allows open communication between a large number of EPGs for the purpose of migration into ACI can heavily constrain which of the following resources?
  - a.** Contract database
  - b.** VLAN encapsulations
  - c.** Endpoint table scalability
  - d.** Policy CAM
- 3.** Which of the following solutions can best optimize hardware resources on leaf switches if the only whitelisting requirement is to allow SSH access to all endpoints in a VRF instance?
  - a.** Standard contracts provided and consumed by all EPGs
  - b.** Preferred group member
  - c.** vzAny
  - d.** Policy Control Enforcement Preference set to Unenforced
- 4.** Which of the following BD configuration knobs governs how Layer 2 multicast traffic is forwarded in an ACI fabric?
  - a.** Multi Destination Flooding
  - b.** ARP Flooding

- c. GARP Based Detection
  - d. L3 Unknown Multicast Flooding
- 5. Which of the following statements about implementation of bridge domain extension is accurate?
  - a. ACI runs all variants of Spanning Tree Protocol.
  - b. No more than one Layer 2 EPG can be associated with a given bridge domain extension.
  - c. Bridge domain extension requires use of a physical domain.
  - d. The same VLAN ID used to extend a bridge domain out a border leaf can be reused for border leaf downstream connectivity to servers via EPG extension.
- 6. Which of the following are valid forms of Layer 2 extension to outside switches in ACI? (Choose all that apply.)
  - a. Remote Leaf Layer 2 domain extension
  - b. AAEP extension
  - c. BD extension
  - d. EPG extension
- 7. Using EPG extension, an engineer has moved all endpoints in a VLAN into an ACI fabric. When he moves the default gateway from traditional switches into the fabric, he suddenly loses all connectivity to the endpoints from outside the fabric. Which of the following are possible reasons this has taken place? (Choose all that apply.)
  - a. The Layer 2 connection between ACI switches and non-ACI switches has been disconnected.
  - b. The bridge domain does not have an associated L3Out configured.
  - c. The subnet Scope parameter on the BD needs to be set to Advertised Externally.
  - d. No contracts have been associated with the EPG.
- 8. A customer has deployed a fabric using two ACI switches. Two overlapping VLANs exist in the customer environment in the DMZ and in the Inside network zone. Both need to be moved into the

fabric using a vPC to the two switches. Is this possible? If so, what feature enables this capability? If not, why?

- a.** Yes. Use the VLAN scope setting Port Local Scope.
  - b.** No. ACI rejects use of an encapsulation for more than a single EPG on a switch.
  - c.** Yes. Use a feature called Global Scope.
  - d.** No. Spanning Tree Protocol in ACI disables any ports to which overlapping VLAN IDs are deployed.
- 9.** An ACI deployment was functioning perfectly when an EPG was extended to two different pairs of external switches. Then the ACI administrator extended the EPG to another external switch, and the performance of all endpoints in the bridge domain degraded to a crawl. By investigating event logs, the administrator finds that ACI never blocked any ports to external switches. What is a possible reason for the performance degradation?
- a.** The ACI administrator enabled MCP on all ports, and MCP blocked redundant connections.
  - b.** ACI does not support connectivity to switches with Rapid PVST+.
  - c.** The ACI administrator forgot to enable Rapid PVST+ on the leaf interfaces facing external switches.
  - d.** External switches connect to ACI with a point-to-point Spanning Tree Protocol port, which has led to premature Spanning Tree Protocol convergence causing a Layer 2 loop.
- 10.** Which one of the following statements is correct?
- a.** Without special configuration, ACI may drop MST BPDUs on ingress.
  - b.** ACI runs Spanning Tree Protocol and participates in the Spanning Tree Protocol topology.
  - c.** Cisco ACI drops Spanning Tree Protocol BPDUs arriving in EPGs associated with a bridge domain if unicast routing has been enabled at the bridge domain level and a default gateway has been deployed to the bridge domain.
  - d.** It is important to enable BPDU filtering on ACI leaf ports facing all external switches to ensure that ACI no longer receives Spanning Tree Protocol BPDUs and becomes loop free.

## Foundation Topics

### Understanding Network Migrations into ACI

On paper, the topic of network migrations is beyond the scope of the Implementing Cisco Application Centric Infrastructure DCACI 300-620 exam and is addressed by the DCACIA 300-630 exam. However, effectively, all of the configuration knobs required for network migrations have actually been included in the DCACI 300-620 exam blueprint. Therefore, this chapter goes a bit beyond the scope of the DCACI 300-620 to provide the additional theoretical coverage necessary for successful Layer 2 extension and basic migration into ACI.

You do not need to memorize every single detail included in this chapter, but you should try to understand the logic behind and use cases for the most important settings related to bridge domains. You should also try to understand the information called out with Key Topic icons for the exam. You should also definitely master the configuration steps needed for EPG and bridge domain extension.

The following section dives into the network-centric approach to ACI, which is the basis for most network migrations into ACI.

### Understanding Network-Centric Deployments

This book has so far emphasized application centricity in the sense that previous chapters divide each sample bridge domain (or the majority of bridge domains) into multiple EPGs and focus on whitelisting of traffic flows via contracts.

However, the assumption that traffic flows can be whitelisted at the moment they are moved into ACI is often unrealistic. The most significant reason for this is that companies often lack a detailed understanding of their application traffic flows and interdependencies. This is *not* meant to suggest that ACI cannot be used for zero-trust security enforcement at the moment endpoints are moved into the fabric. Indeed, ACI *can* be used this way.

The polar opposite of the application-centric model described in earlier chapters is the network-centric approach to ACI. The *network-centric approach* is more of a mindset than any specific ACI feature. It allows network teams to move endpoints into ACI without necessarily changing the network architecture. This ensures that they can achieve some of the benefits of ACI while familiarizing themselves with the platform.

In essence, you can think of the network-centric approach as a way to dumb down ACI to the level of traditional networks, whose security relies mostly on VLANs, VRF instances, and rudimentary access list enforcement.

The most fundamental aspect of the network-centric approach is that each VLAN in traditional networking needs to be mapped to a single bridge domain and a single EPG. In other words:



**Network-centric approach: Each VLAN = 1 bridge domain = 1 EPG = 1 subnet**

Adoption of the network-centric approach does not necessarily mean that security mechanisms such as contracts are not used. However, it does imply that there will be minimal security enforcement within subnet boundaries. For the most part, the network-centric mode, when seen in the context of how traditional data centers are built, assumes that the ACI fabric needs to be configured to perform blacklisting. Therefore, network-centric deployment often starts out with all traffic flows being allowed unless denied through explicit security enforcement. That said, network-centric deployment is often seen as a stepping stone toward application-centric deployment.

## **Understanding Full-Mesh Network-Centric Contracts**

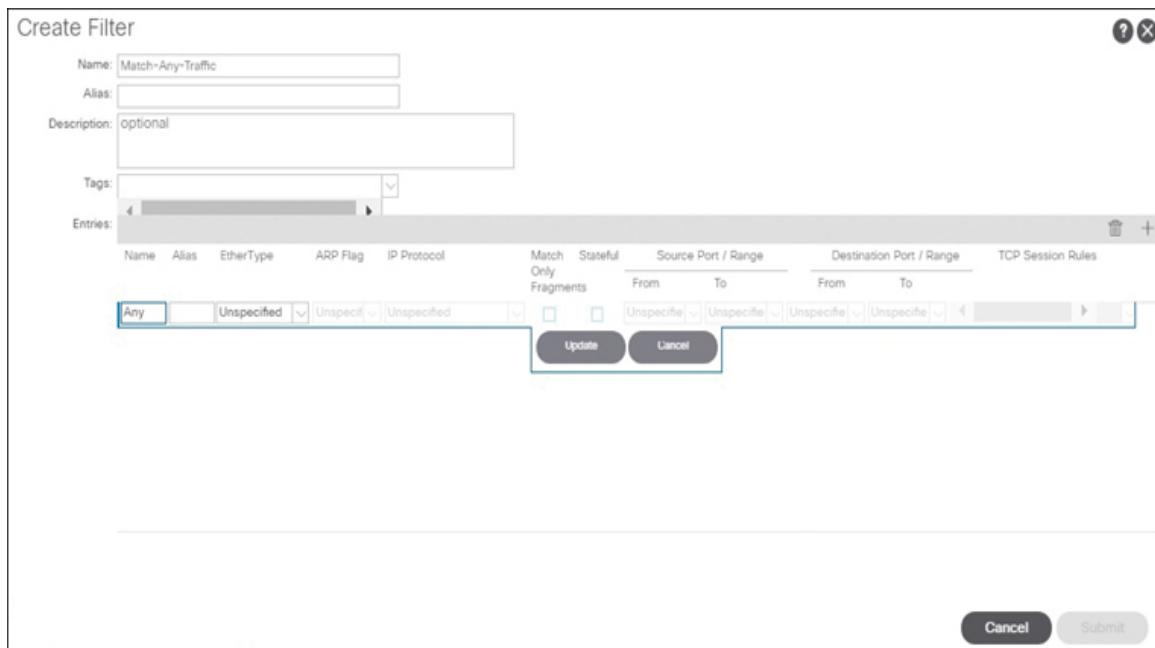
One way for ACI to mimic a traditional network is through assignment of a contract allowing any-to-any communication between all EPGs. Because this any-to-any contract needs to be applied to all EPGs, it can be understood as a sort of full mesh of contract relationships.

The only real drawback to this approach is that with any full mesh, the number of relationships between participants grows exponentially. Translated into ACI terminology, any-to-any communication using contracts can consume a large amount of policy content-addressable memory (CAM) resources.

The thought process with full-mesh any-to-any contracts is that VLANs will be migrated into ACI in network-centric mode, and IT will later move endpoints into application-centric EPGs dedicated to endpoint functions. Once the original EPGs have the endpoints removed, they can be deleted. The resulting deployment is application-centric.

If an eventual evolution from a network-centric design to a more application-centric approach is not one of the drivers for migration to an ACI fabric, use of a full-mesh network-centric contract may not be the most ideal migration approach.

Let's examine what an any-to-any contract might look like. [Figure 10-1](#) shows the creation of a filter that matches all traffic. The single entry in the filter has the EtherType value Unspecified. No other tweaks have been made from the default entry settings.



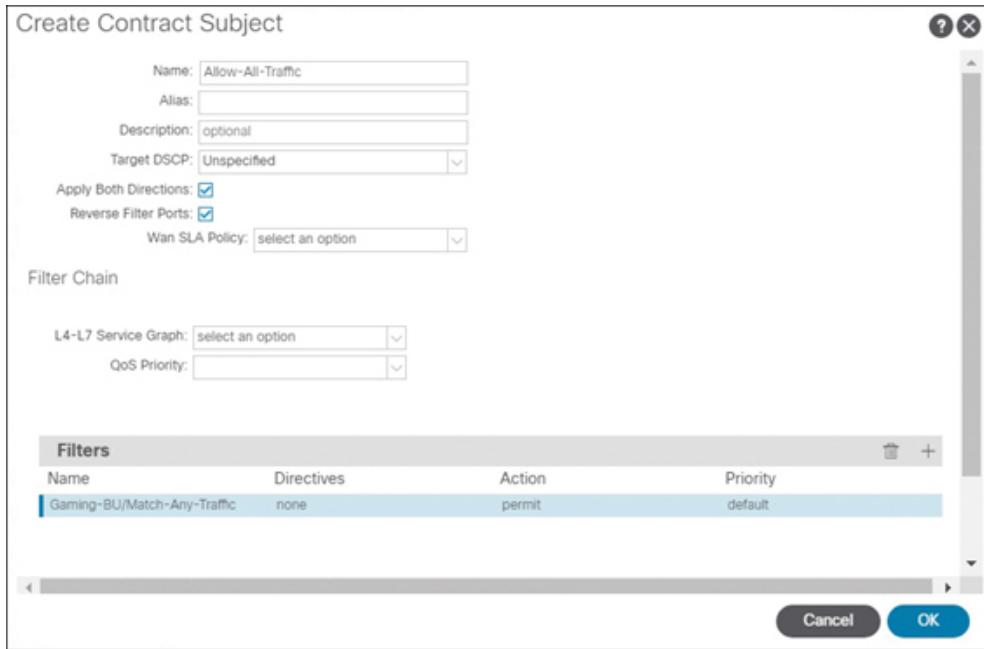
**Figure 10-1** Creating a Filter Matching All Traffic

This contract is pretty basic. [Figure 10-2](#) shows creation of a contract named Permit-Any with Scope set to VRF. To create a subject for the contract, you click the + sign next to Subjects.

The screenshot shows the 'Create Contract' dialog box. At the top, there are fields for Name (set to 'Permit-Any'), Alias, Scope (set to 'VRF'), QoS Class (Unspecified), Target DSCP (Unspecified), and Description (optional). Below these are fields for Tags (enter tags separated by comma) and Subjects. The Subjects section contains a table with columns 'Name' and 'Description', which is currently empty. At the bottom right are 'Cancel' and 'Submit' buttons.

**Figure 10-2** Creating an Any-to-Any Contract

Finally, you can associate the filter with the contract by adding it to the subject and enabling the Apply Both Directions and Reverse Filter Ports settings (see [Figure 10-3](#)). Click OK and then click Submit to complete the process.



**Figure 10-3** *Associating a Filter Matching All Traffic to a Contract Subject*

The contract should then be allocated as both consumed and provided on all network-centric EPGs as well as any external EPGs that should be able to access the EPGs.

### Note

A contract with no filter applied also enables the any-to-any communication necessary for open network-centric communication.

## Understanding Any EPG

### Key Topic

**Any EPG**, more commonly called **vzAny**, provides a convenient way of associating all endpoint groups (EPGs) in a VRF instance with one or more contracts.

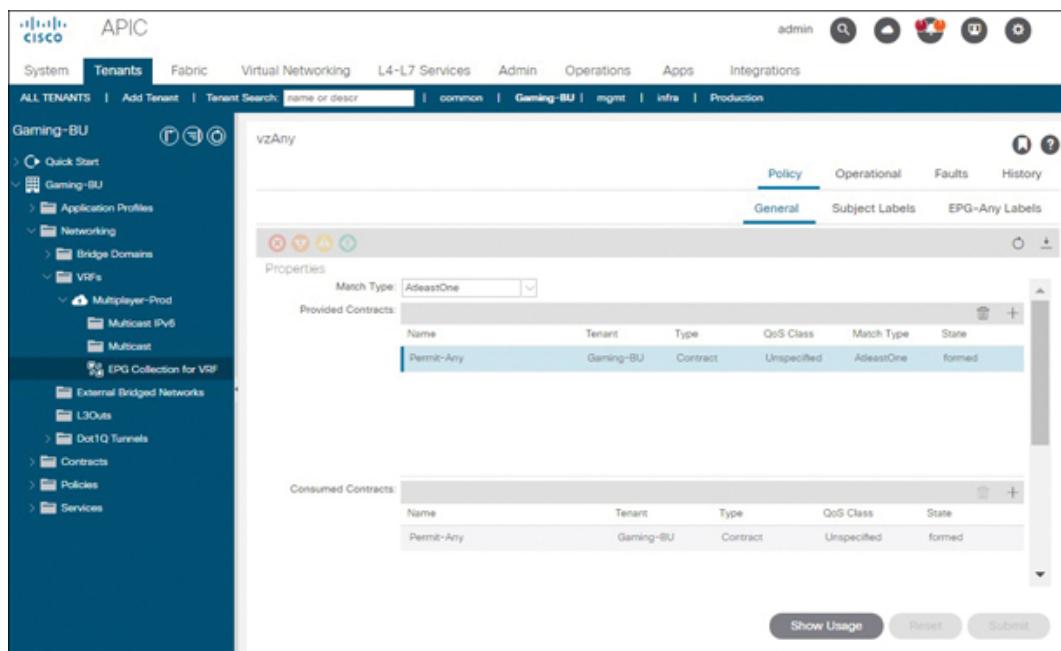
Whereas a contract applied bidirectionally to tens of EPGs forms a many-to-many relationship, vzAny creates a one-to-all contract

relationship and is thus considered an optimization of policy CAM space more than anything else.

The ideal use case for vzAny is permitting common services. For example, if all endpoints in all EPGs within a specific VRF instance should be able to respond to **ping**, a contract allowing ICMP traffic can be associated with vzAny in the given VRF instance in the provided direction. All EPGs that need to ping these endpoints would then need to consume the contract.

Likewise, if all endpoints within a VRF should be able to query a set of DNS servers, a contract allowing DNS queries can be configured and consumed by vzAny within the VRF. The same contract would then be allocated to the DNS server EPG as a provided contract.

**Figure 10-4** shows how vzAny can be used to enable open communication among all EPGs within a VRF when the any-to-any contract defined previously is instead associated with vzAny in both the provided and consumed directions.



**Figure 10-4** Using vzAny as the Contract Enforcement Point for Open Communication

As shown in the figure, you can associate contracts with vzAny by navigating to the VRF in question and opening the EPG Collection for VRF subfolder.

### Note

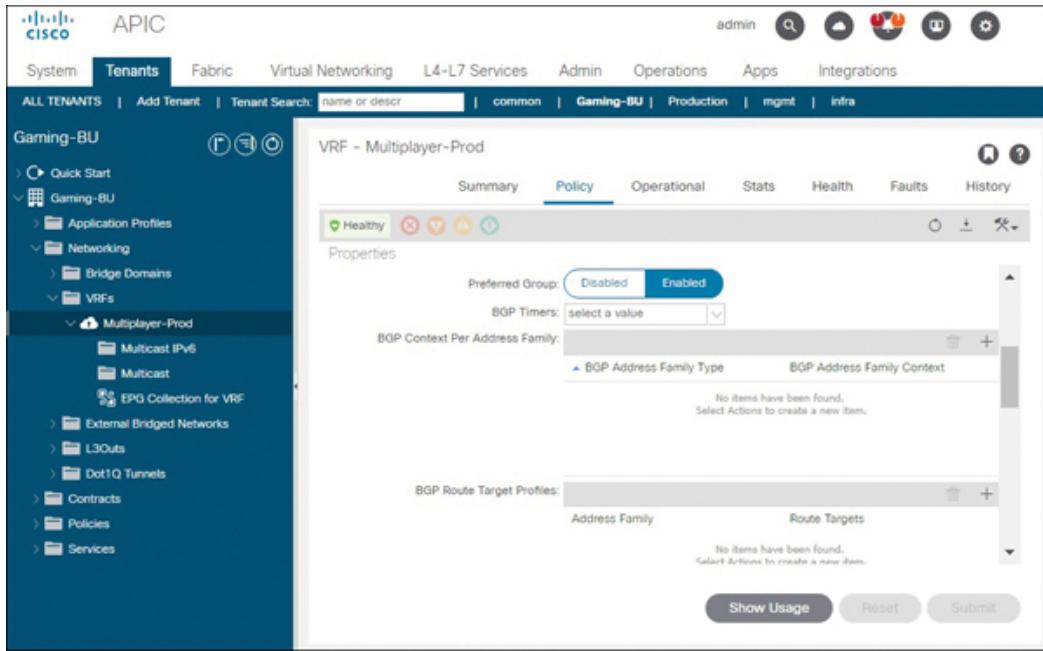
The advent of preferred group members has dramatically reduced the utility of vzAny as a mechanism for establishing open communication among EPGs.

## Understanding Preferred Group Members

Whereas vzAny is great for opening specific ports and services to all endpoints in a VRF instance, it poses a new challenge when used as a network-centric implementation tool. If open communication is enforced between all EPGs in a VRF, how can companies transition to a whitelisting model? This is exactly the question that preferred group members address.

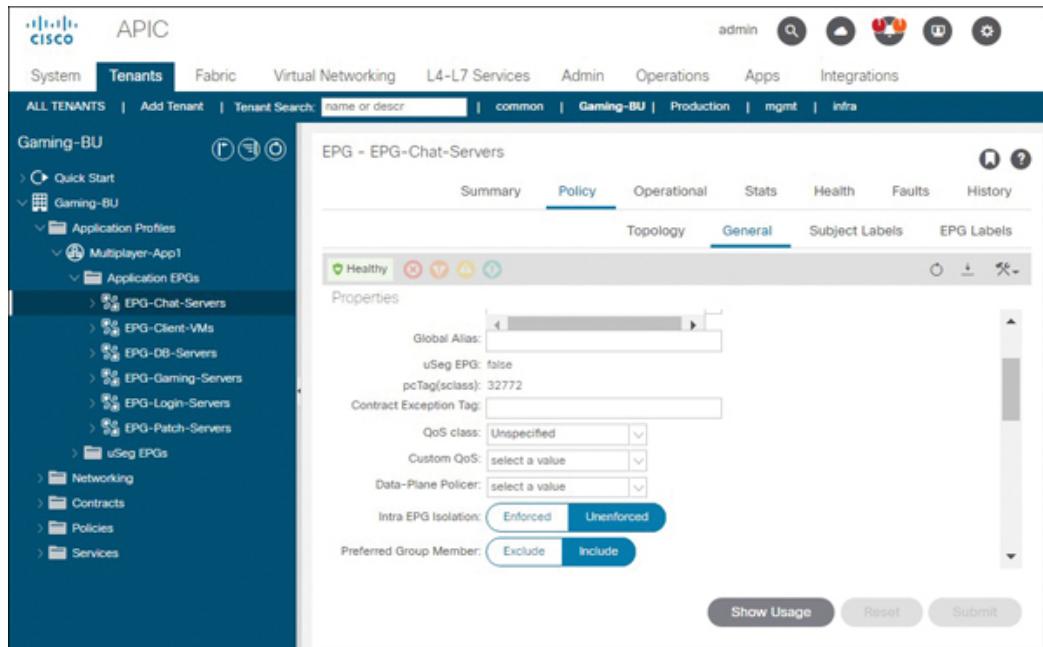
Select EPGs within a VRF instance, called **preferred group members**, can be afforded open communication, while others can be locked down with contracts. Basically, this enables all EPGs to be moved into a fabric with open communication, and once contracts are fully defined for an EPG, the EPG can then be excluded as a preferred group member to allow for full contract enforcement.

[Figure 10-5](#) shows that the Preferred Group setting needs to be first toggled to Enabled at the VRF level to confirm that the VRF is a candidate for open communication among EPGs.



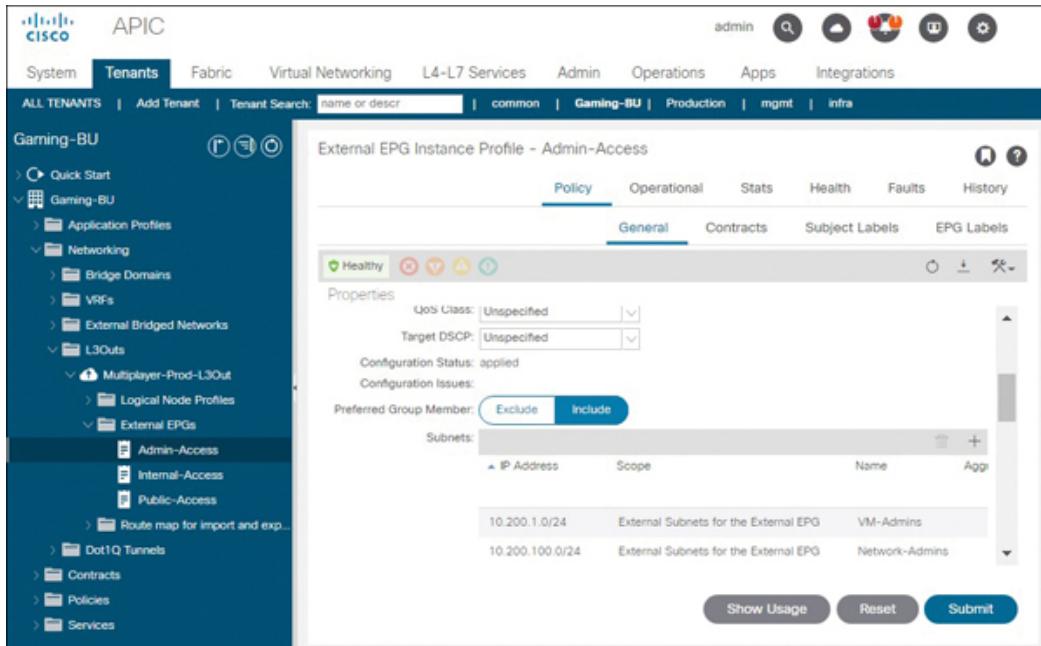
**Figure 10-5 Enabling Preferred Group at the VRF Level**

Once Preferred Group is enabled at the VRF level, you can navigate to each individual EPG that is a candidate for open communication and toggle the Preferred Group Member setting to Include under **Policy > General**, as shown in [Figure 10-6](#).



**Figure 10-6 Including an EPG as a Preferred Group Member**

If outside endpoints also need open communication with all EPGs configured as preferred group members within the VRF instance, you need to also enable the Include setting for the external EPG (see [Figure 10-7](#)).



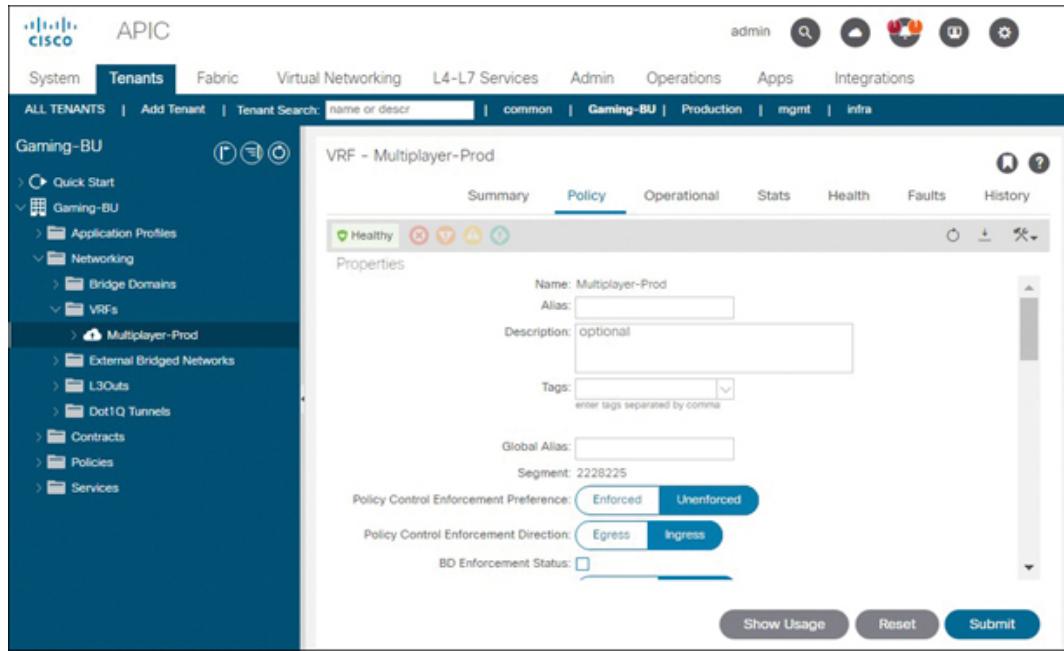
**Figure 10-7** Including an External EPG as a Preferred Group Member

If at any point contracts are defined for an EPG that is part of the preferred group, the Preferred Group Member setting on the EPG can be toggled back to Exclude, which is the default value.

## Disabling Contract Enforcement at the VRF Instance Level

Although it is very uncommon to deploy new fabrics with contract enforcement disabled, use of the Policy Control Enforcement Preference setting is feasible if the deployment is not expected to ever make use of contracts. It is also a useful feature for troubleshooting whether the loss of expected connectivity is a result of a contract misconfiguration.

[Figure 10-8](#) shows how contract enforcement can be disabled for a VRF instance by setting Policy Control Enforcement Preference to Unenforced under the VRF instance.



**Figure 10-8** Disabling Contract Enforcement Altogether for an Individual VRF Instance

## Flooding Requirements for L2 Extension to Outside Switches

Figure 10-9 shows the best practice bridge domain configurations around forwarding that must be in place for proper Layer 2 extension between ACI and external switches. The use of these settings is not mandatory in every setup. However, these settings help ensure that ACI behaves as much like traditional switches as possible, without data plane learning being disabled, thereby resolving some forwarding issues that may be experienced when migrating endpoints and subnets into ACI. Note that use of these settings is especially important when the default gateway for a subnet is outside ACI.

Create Bridge Domain

STEP 1 > Main

1. Main    2. L3 Configurations    3. Advanced/Troubleshooting

Name: BD-VLAN27 |

Alias:

Description: optional

Tags:  enter tags separated by comma

Type:  fc  regular

Advertise Host Routes:

VRF: Multiplayer-Prod

Forwarding: Custom

L2 Unknown Unicast: Flood

L3 Unknown Multicast Flooding: Flood

Multi Destination Flooding: Flood in BD

ARP Flooding:  Enabled

This screenshot shows the 'Create Bridge Domain' configuration page. The 'Main' tab is selected. The 'Name' field contains 'BD-VLAN27'. The 'Type' dropdown is set to 'regular'. Under 'VRF', 'Multiplayer-Prod' is selected. 'Forwarding' is set to 'Custom'. For 'Unknown Unicast', 'Multicast', and 'Destination Flooding', the option 'Flood' is chosen. 'ARP Flooding' is enabled. Other fields like 'Alias' and 'Description' are optional.

**Figure 10-9** Recommended BD Settings When Endpoints Attach to Non-ACI Switches

Table 10-2 explains what these settings do and the logic behind their use.



**Table 10-2** Bridge Domain Settings for ACI Layer 2 Extension to Non-ACI Switches

## B Required Setting for Property and Justification

D

P  
r  
o  
p  
e  
r  
t  
y

F The Forwarding field only appears when a bridge domain is first configured. Its default value, Optimized, automatically sets the Unicast and ARP parameters. To enable customization of forwarding settings to values that enable Layer 2 extension, select a the Custom option.

r  
d  
i  
n  
g

**L**This field applies to unicast traffic destined to an endpoint whose  
**2**MAC address cannot be found in the ACI endpoint table.

**U**  
**n**The forwarding options available for the L2 Unknown Unicast  
**k**parameter are Flood and Hardware Proxy. When endpoints directly  
**n**attach to leaf switches and ACI is the default gateway for the BD,  
**o**hardware proxy forwarding is preferred because it allows for a  
**w**reduction in flooding within the fabric. However, when some  
**n**endpoints associated with a bridge domain as well as the default  
**U**gateway for the BD subnet(s) reside outside the fabric, the ACI  
**n**spine proxy forwarding behavior can lead to suboptimal learning  
**i**on non-ACI switches outside the fabric. For this reason, the L2  
**c**Unknown Unicast setting needs to be set to Flood to accommodate  
**a**any endpoints behind the Layer 2 extension until default gateways  
**s**are moved into the fabric and unicast routing is enabled on  
**t**the BD.

**L**By default, IGMP snooping is enabled on bridge domains. The IGMP  
**3**snooping feature snoops the IGMP membership reports and leave  
**U**messages and forwards them to the IGMP router function only  
**n**when necessary. When a leaf receives traffic for a multicast group  
**k**that is unknown, this traffic is considered unknown Layer 3  
**n**multicast, and the L3 Unknown Multicast Flooding setting  
**o**determines how the traffic is forwarded. The two options for this  
**w**setting are Flood and Optimized Flood. When Flood is selected,  
**n**traffic destined to unknown multicast groups is flooded on the  
**M**ingress switch and any border leafs on which the BD is active.  
**u**When Optimized Flood is selected, traffic for the unknown  
**I**multicast group is forwarded to the multicast router ports only.

**t**  
**i**  
**c**  
**a**  
**s**  
**t**  
**F**  
**I**  
**o**  
**o**  
**d**  
**i**  
**n**  
**g**

- A When the ARP Flooding parameter is enabled, ARP requests with a broadcast destination MAC address are flooded in the bridge domain. If this option is disabled and the fabric has already learned the destination endpoint, it unicasts the ARP request to the destination. If this option is disabled and the fabric has *not* learned the destination endpoint, it uses ARP gleaning to identify the destination endpoint. When unicast routing is disabled, ARP traffic is always flooded, even if the ARP Flooding parameter has been disabled on the BD.
- n
- g Enabling ARP Flooding ensures that ACI behaves much like traditional networks and allows non-ACI switches behind a Layer 2 extension to proactively learn endpoints residing in the fabric. This, by itself, should be sufficient justification for its use during migrations into ACI. There is one other compelling use case for enabling the ARP Flooding parameter that relates to silent hosts. Remember from [Chapter 8, “Implementing Tenant Policies,”](#) that ARP gleaning detects silent hosts by prodding them into communicating on the network, but in the rare case that the silent host moves elsewhere without sending a GARP packet into the network, ACI continues to think that the endpoint details it learned prior to the endpoint move are accurate. In this case, if ARP Flooding has been disabled, the ACI leaf continues to unicast ARP requests that are destined to the silent host to the old location until the IP endpoint ages out. On the other hand, with ARP Flooding enabled, ACI floods all ARP requests with broadcast destination MAC addresses. When the silent host receives the ARP request, it responds to the ARP request, prompting ACI nodes to update the endpoint table accordingly. Even though the issue of silent hosts is not specifically related to Layer 2 extension, this example should help illuminate why the ARP Flooding parameter can help alleviate some corner-case endpoint learning issues.

**M**This parameter primarily addresses forwarding of traffic types not covered by the other settings mentioned in this table, such as broadcast, L2 multicast, and link-local traffic. There are three configuration options for the Multi Destination Flooding property:

**i  
D  
e  
s  
t.  
i  
n  
a  
t  
i  
o  
n**

**Flood in BD:** Sends a packet to all ports in the same bridge domain.

**n  
F  
I  
o  
o  
d  
i  
n  
g**

**Drop:** Drops a packet and never sends it to any other ports.

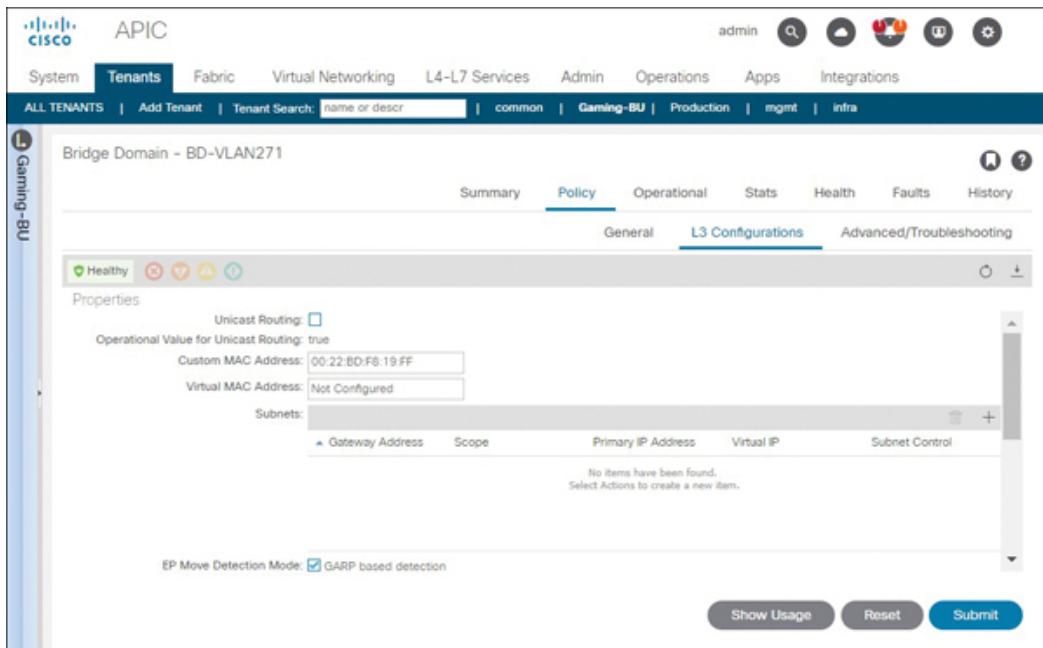
**Flood in Encapsulation:** Sends a packet to all ports in the same VLAN encapsulation. If there is a one-to-one relationship between encapsulations and EPGs, this setting effectively limits flooding to each EPG.

Note that while Flood in Encapsulation does enable Layer 2 extension and is an option in some deployments, there are more caveats that require careful consideration when using this option for migrations. The Flood in BD option, which is the default setting for the L3 Unknown Multicast Flooding bridge domain property, remains the most ideal setting for Layer 2 extension.

# Understanding GARP-Based Detection

GARP-based detection helps in a variety of scenarios in both first- and second-generation Cisco ACI leaf switches.

Although ACI can detect MAC and IP address movement between leaf switch ports, leaf switches, bridge domains, and EPGs, first-generation leaf switches cannot detect the movement of an IP address to a new MAC address if the new MAC address resides behind the same switch interface and the same EPG as the old MAC address. Enabling GARP-based detection addresses this caveat related to first-generation switches as long as the ARP Flooding parameter has also been enabled on the relevant bridge domain (see [Figure 10-10](#)).



**Figure 10-10** GARP-Based Detection Enabled in the L3 Configurations Subtab of a BD

When might an IP address reasonably move to a new MAC address, and what is the probability that the IP address will remain behind a single interface, port channel, or vPC? A common example of an IP address moving to a different MAC address is a high-availability cluster failover, as with some load balancer and firewall cluster setups. When the failover takes place with non-ACI switches behind a Layer 2 extension, the failover in essence stays behind a single interface, port channel, or vPC. If unicast routing has been enabled

on the BD and the switch connecting to the non-ACI switch(es) is a first-generation leaf, ACI communication with the cluster IP is, in effect, black-holed.

Note that GARP-based detection is also a significant configuration item in second-generation leaf switches. If there is ever a need to disable ACI data plane learning for a bridge domain or VRF, GARP-based detection along with ARP flooding enable the network to perform endpoint learning using traditional control plane-oriented methods.

That said, you should never disable data plane learning without first consulting the latest Cisco documentation! The only valid use case for disabling data plane learning is to do so in conjunction with service graphs. If there is a valid technical driver for disabling data plane learning outside of service graphs, it should only be done at the VRF level.

### Note

You may have noticed in [Figure 10-10](#) that the Unicast Routing checkbox has been disabled for the BD. It is a common misconception that this feature can be enabled on bridge domains whose default gateways are outside the fabric to force ACI to learn endpoint IP addresses. The problem with this approach, as well intentioned as it may be, is that it may also cause ACI to directly forward traffic to other endpoints in the Layer 3 domain if ACI also happens to have learned the destination endpoint IP address. This behavior can lead to asynchronous routing and can inadvertently change traffic flows in the data center. If the default gateway for a subnet is outside ACI and the fabric should not be performing routing for the bridge domain, you should save yourself the headache and disable the Unicast Routing checkbox for the bridge domain.

## Understanding Legacy Mode

One feature you can use to increase the number of VLAN IDs available for encapsulating traffic out of leaf switch ports is the Legacy mode bridge domain subconfiguration. Use of this feature makes sense only for network-centric bridge domains in

environments in which thousands of VLANs need to be deployed to a given switch. This feature locks the bridge domain and its corresponding EPG into a single encapsulation, freeing up the encapsulation that would have been used by the bridge domain.

Because there are quite a few caveats associated with Legacy mode and due to the fact that it reduces the ease of migrating to an application-centric model, it is not used very often. But you need to be aware of this feature and why it exists.

You can find the Legacy Mode checkbox to enable this feature by navigating to the desired bridge domain, selecting the Policy tab, and then selecting the General subtab.

## **Endpoint Learning Considerations for Layer 2 Extension**

An important consideration for Layer 2 extension is the number of endpoints that an ACI leaf may be expected to learn from external switches.

You can review the Verified Scalability Guide for your target ACI code release to understand the number of endpoints each leaf platform can learn based on its ASICs.

In larger environments, it helps to distribute Layer 2 extensions among multiple leafs and to trunk VLANs to different leafs to reduce the likelihood of filling up the local station table on the leafs.

## **Preparing for Network-Centric Migrations**

There are certainly tenant design and endpoint placement considerations that you need to think about that are beyond the scope of this book. But if all endpoints moving into the fabric are expected to be placed in a single user tenant and VRF instance, you may be able to simply create a basic table. This table would need to include each VLAN that needs to be migrated and its equivalent network-centric bridge domain, EPG, and subnet. It should also include ideal target state settings for the bridge domain, including whether hardware proxy, ARP flooding, and/or GARP-based detection should be enabled following the move of the subnet default gateway into the fabric.

If there is a need for multiple Layer 2 or Layer 3 connections between an ACI fabric and non-ACI switches, this prep work should also detail the Layer 2 connections through which a bridge domain or EPG needs to be extended or the L3Out(s) through which bridge domain subnets need to be advertised after the migration.

## Implementing Layer 2 Connectivity to Non-ACI Switches

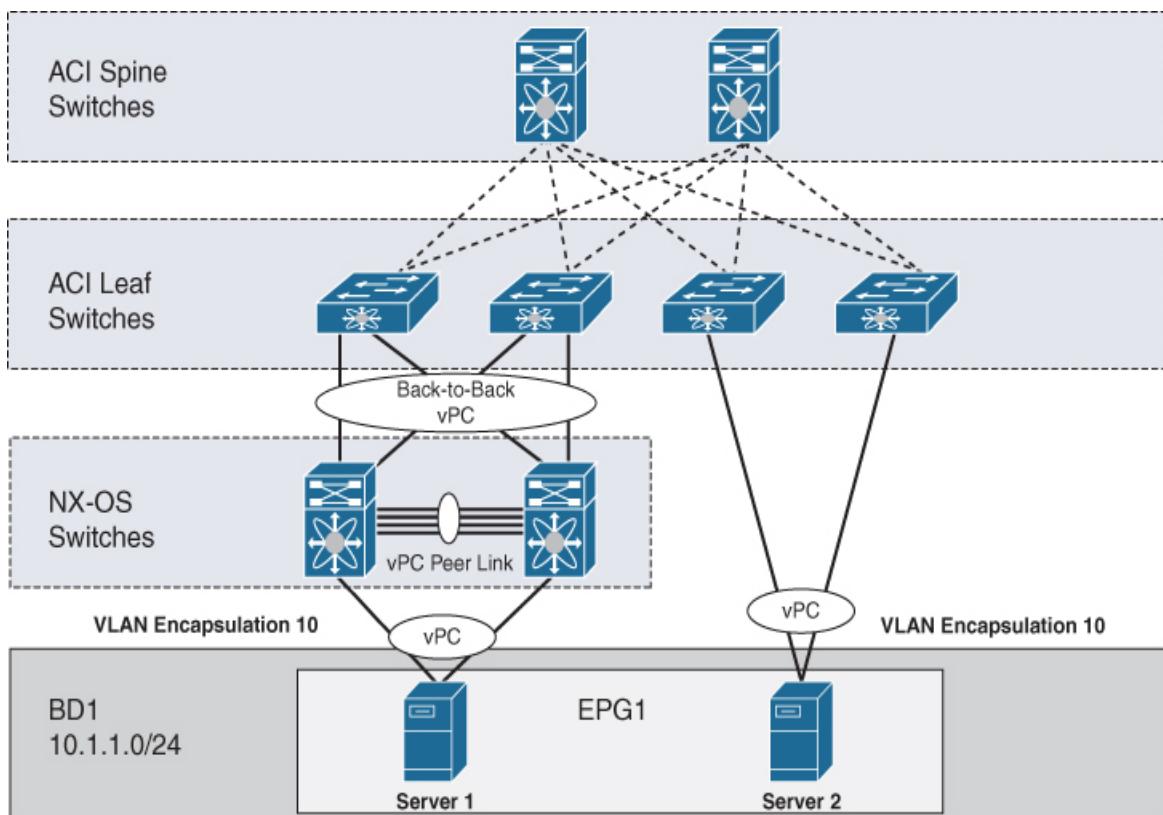
There are three primary methods for extending Layer 2 outside an ACI fabric:

- **Extend the EPG out the ACI fabric:** In this method, an ACI administrator extends an EPG out an ACI fabric by statically mapping the EPG to a VLAN ID on a given port, port channel, or vPC. Extending an EPG outside a fabric allows endpoints both within and outside ACI to be classified into the same EPG.
- **Extend the bridge domain out the ACI fabric:** Technically, the term *L2Out* refers to bridge domain extension even though the term may be used colloquially for both bridge domain and EPG extensions. In the ACI GUI, bridge domain extensions are also referred to as ***external bridged networks***. When extending a bridge domain, ACI classifies endpoints residing in a VLAN outside the fabric into a Layer 2 EPG (an external EPG used in a bridge domain extension). Administrators can create additional EPGs and associate them with the bridge domain to enable policy enforcement between the external Layer 2 EPG and any other EPGs in the fabric. This can essentially enable enforcement of a zero-trust architecture at the moment endpoints are fully moved into a fabric, but it still requires that traffic flows between endpoints be well understood.
- **Extend the Layer 2 domain with remote VTEP:** The remote VTEP feature can be used to implement either EPG extension or bridge domain extension. Remote VTEP is beyond the scope of the DCACI 300-620 exam and is not discussed further in this book.

## Understanding EPG Extensions

If you have followed along and read this book chapter by chapter, EPG extension should not be new to you. Statically mapping an EPG to a VLAN on a switch port connecting to a server *is* EPG extension.

[Figure 10-11](#) illustrates the extension of an EPG to non-ACI switches. Note three significant points regarding EPG extension in this figure. First, EPG extension does not necessitate use of a new VLAN ID for server connections. Second, endpoints within the EPG, regardless of location (inside or outside the fabric), are considered part of the same EPG. This means there is no need for contracts to allow these endpoints to communicate. Finally, for EPG extension with the recommended flooding settings described in the previous section to work seamlessly, no more than one EPG associated with each bridge domain should ever be extended.



**Figure 10-11** Extending an EPG out a Fabric

### Note

Quite a few engineers have asked whether multiple EPGs associated with a single bridge domain can be extended to non-

ACI switches outside a fabric. The answer is yes. Among the options for Multi Destination Flooding, administrators can choose Flood in Encapsulation at the bridge domain level to isolate flooding to each associated EPG.

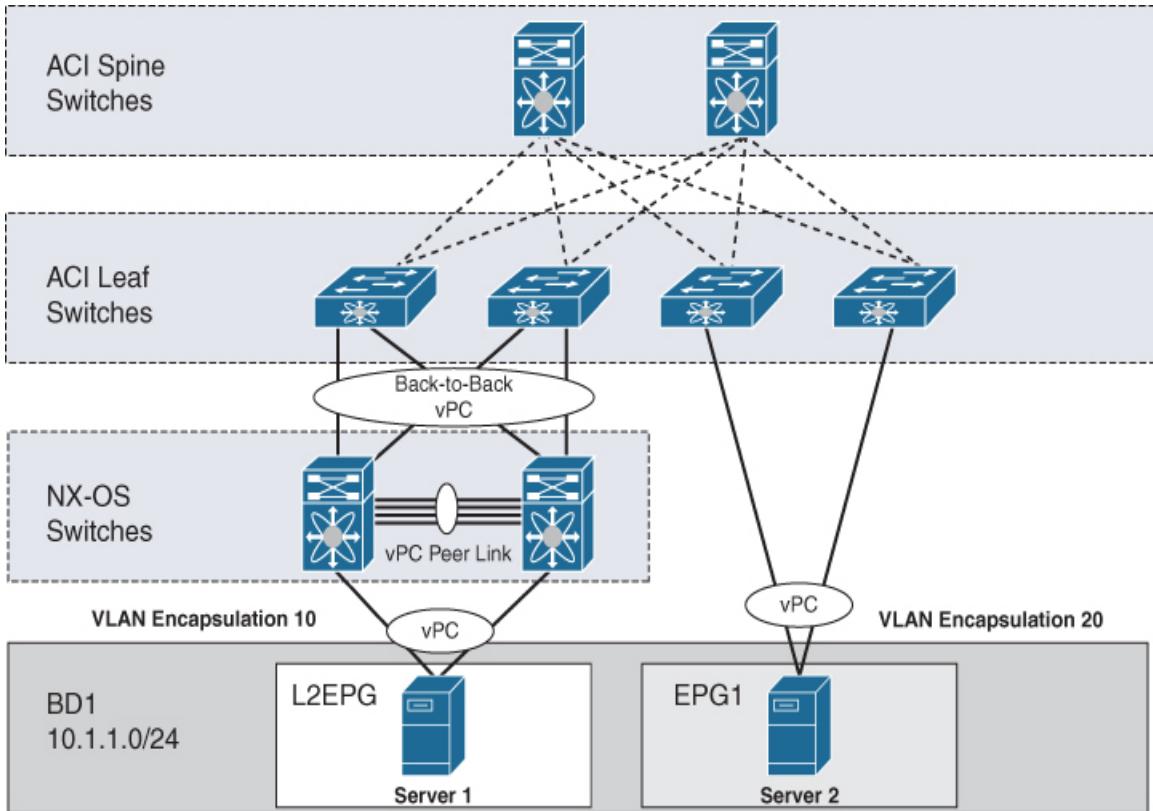
In the context of migrations, the use case many proponents of this feature have in mind is to consolidate multiple VLANs and subnets into a small number of bridge domains. This way, when migrating to an application-centric model, endpoints dedicated to a particular function that live across several subnets but are bound to a common bridge domain can be placed into a single EPG as opposed to having the number of EPGs aligned with the number of network-centric bridge domains.

The challenge with this feature is that not all hardware supports it, and it is riddled with caveats. For instance, ACI Multi-Site does not currently support the stretching of objects that have this feature enabled. Therefore, use of this feature is not suitable in all environments. If you decide to use this feature to migrate multiple VLANs into a single bridge domain, make sure you understand all the caveats.

## Understanding Bridge Domain Extensions

Bridge domain extensions are dramatically different from EPG extensions. [Figure 10-12](#) demonstrates two of the main differences:

- Bridge domain extensions necessitate use of a separate encapsulation for EPGs inside the fabric compared to the Layer 2 EPG encapsulation used to bridge the traffic to non-ACI switches.
- Endpoints in a subnet extended between ACI and non-ACI switches are in separate EPGs, so no communication is allowed between internal and external endpoints until contracts or Preferred Group Member settings are put in place or contract enforcement is disabled.



**Figure 10-12** Extending a Bridge Domain out a Fabric

## Comparing EPG Extensions and BD Extensions

Table 10-3 recaps the points already made and compares EPG extensions with BD extensions based on a number of criteria.



**Table 10-3** Comparison Between Bridge Domain Extension and EPG Extension

Comparison of Extend EPG and Extend Bridge Domain		
	Extend EPG	Extend Bridge Domain
Comparison Criteria	Co mp ari son Crit eri a	Extend Bridge Domain
Use cases	Extend EPG beyond an ACI fabric; migrate VLANs into ACI in network-centric mode with Flood in BD; consolidate multiple VLANs and subnets into a single bridge domain at the time of migration to ACI by using Flood in Encapsulation	Extend a bridge domain out the fabric or extend a tenant subnet of the bridge domain out the fabric; migrate VLANs into ACI with intra-VLAN policy enforcement applied at the time of migration
Configuration	Statically assign a port to an EPG (static binding under EPG or direct assignment to an AAEP)	Create external bridged networks (L2Out) in a tenant where a bridge domain resides
Domain type applicable	Physical domain	External bridged domain

External endpoint placement	Endpoints connected to non-ACI switches placed in the same EPG (VLAN) as directly attached endpoints	Endpoints connected to non-ACI switches in a different EPG (VLAN) but the same bridge domain as directly attached endpoints
Policy model	External endpoints are seen as an internal EPG, and the same principles apply.	An external endpoint is placed under an external EPG (Layer 2 EPG). Policy is applied between internal EPGs and a Layer 2 EPG.
Endpoint learning	ACI learns both MAC and IP addresses. (IP addresses are only learned if unicast routing is enabled at the BD level.)	ACI learns both MAC and IP addresses. (IP addresses are only learned if unicast routing is enabled at the BD level.)

### Note

EPG extensions are by far the most popular method for migrating VLANs into ACI.

## Implementing EPG Extensions

Just as with any other type of switch port configuration in ACI, the first thing you need to do when implementing an EPG extension is to configure access policies. [Figure 10-13](#) shows the configuration of a vPC interface policy group toward a non-ACI switch. Note that in this

case, a Spanning Tree Protocol interface policy called Switch-Facing-Interface is created to ensure that BPDU Filter and BPDU Guard are not enabled on this particular vPC interface policy group. In the Attached Entity Profile field, a newly created AAEP called L2-to-Legacy-Network-AAEP is selected. The figure does not show it, but this AAEP has a physical domain named GamingBU-Physical as its domain association, enabling use of a range of static VLAN IDs assigned to the VLAN pool associated with GamingBU-Physical as potential encapsulations for the EPG extension.

The screenshot shows a configuration dialog titled 'Create VPC Interface Policy Group'. The 'Name' field is set to 'EPG-Extension-vPC-PolGrp'. The 'Attached Entity Profile' dropdown is set to 'L2-to-Legacy-Network-AAEP'. Other fields like 'Link Level Policy', 'CDP Policy', and 'LLDP Policy' are set to their default values. The 'Port Channel Policy' is set to 'LACP-ACTIVE'. The 'Monitoring Policy' dropdown is also visible.

**Figure 10-13** Configuring an Interface Policy Group as a Prelude to EPG Extension

Next, you need to map the interface policy group to switch ports. [Figure 10-14](#) shows the vPC interface policy group being mapped to port 1/6 on two switches that have been bound to a single interface profile.

Create Access Port Selector

Name: L2-vPC-to-Legacy-Switches

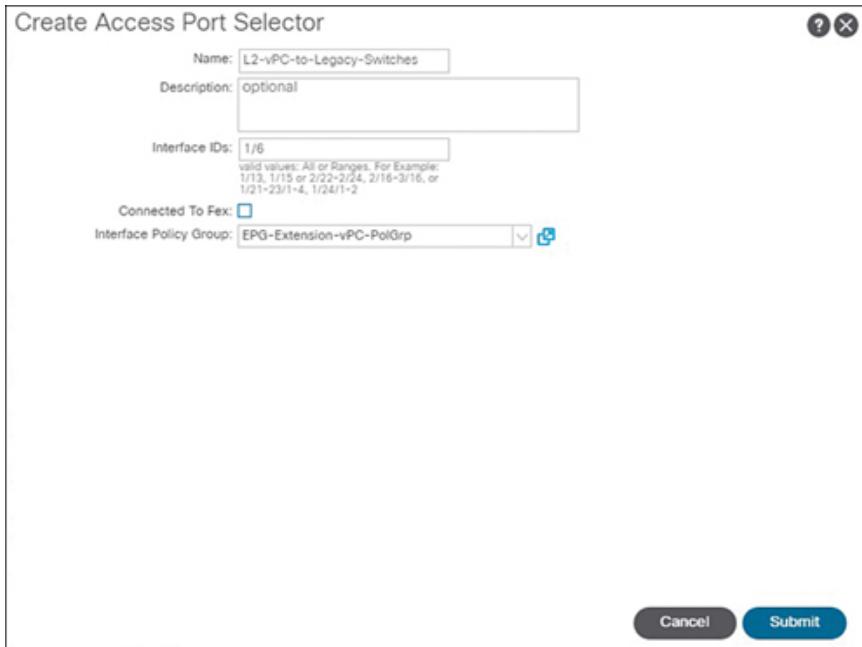
Description: optional

Interface IDs: 1/6  
Valid values: All or Ranges. For Example:  
1/13, 1/15 or 2/22-2/54, 2/16-3/16, or  
1/21-23/1-4, 1/24/1-2

Connected To Flex:

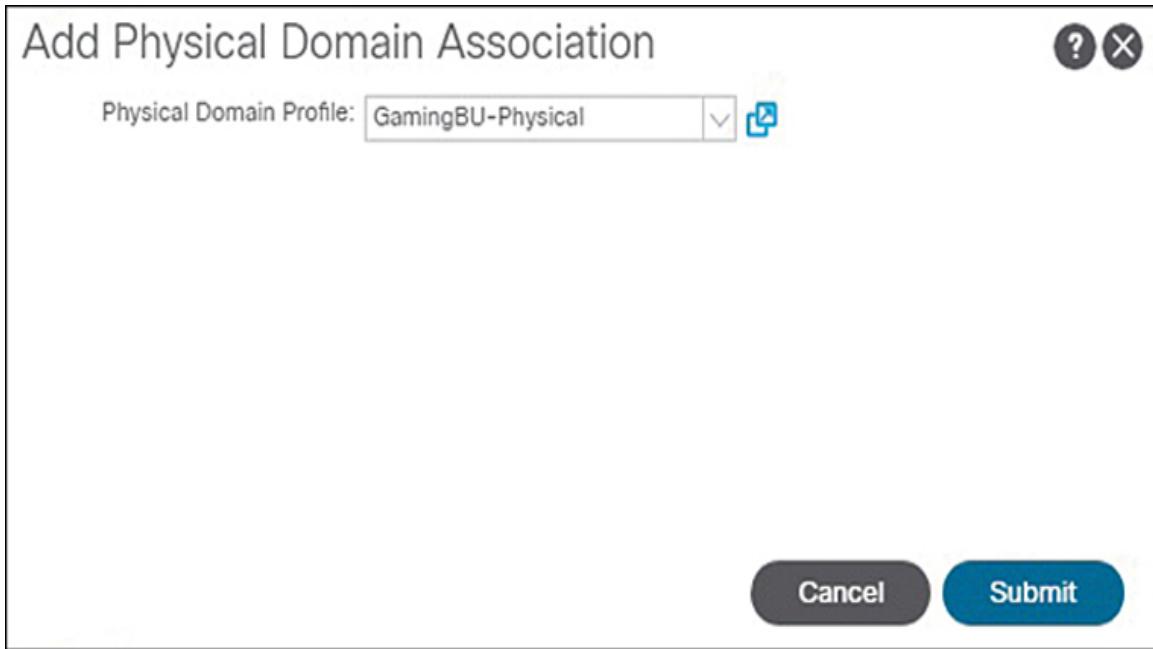
Interface Policy Group: EPG-Extension-vPC-PolGrp

Cancel Submit



**Figure 10-14** Implementing Access Policies as a Prerequisite to EPG Extension

After verifying that the vPC has come online, you navigate to the Tenants view to extend the desired EPGs out the fabric. [Figure 10-15](#) shows a new EPG called EPG-VLAN271 being associated with the physical domain GamingBU-Physical, so that VLAN encapsulations allowed by the physical domain and port assignments associated with the relevant AAEP can be used to extend the given EPG.



**Figure 10-15** *Associating a Physical Domain with an EPG*

With the physical domain assignment in place, the stage has been set to map the EPG to an allowed encapsulation on the vPC. To do this mapping, you navigate to the desired EPG and expand its subfolders, right-click the Static Ports subfolder, and select Deploy Static EPG on PC, VPC, or Interface. [Figure 10-16](#) shows the resulting wizard. In this wizard, you enter the vPC interface policy group created earlier as the path and enter the VLAN ID used to encapsulate the traffic in the Port Encap field. Notice the options in the Mode field and that the port does not necessarily need to be configured as a trunk. Finally, review the Deployment Immediacy setting. (Deployment Immediacy settings are covered in [Chapter 8](#). They are further covered in [Chapter 12, “Implementing Service Graphs.”](#)) The Deployment Immediacy setting Immediate is often the best choice for EPG extensions to non-ACI switches because it commits the configuration (and any relevant contracts, if applicable) to hardware immediately.