The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

Notwithstanding any other warranty herein, all document files and software of these suppliers are provided “as is” with all faults. Cisco and the above-named suppliers disclaim all warranties, expressed or implied, including, without limitation, those of merchantability, fitness for a particular purpose and noninfringement or arising from a course of dealing, usage, or trade practice.

In no event shall Cisco or its suppliers be liable for any indirect, special, consequential, or incidental damages, including, without limitation, lost profits or loss or damage to data arising out of the use or inability to use this manual, even if Cisco or its suppliers have been advised of the possibility of such damages.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2017–2020 Cisco Systems, Inc. All rights reserved.
CONTENTS

PREFACE

Preface xv
  Audience xv
  New and Changed Information xv
  Document Conventions xvi
  Related Documentation xvii
  Documentation Feedback xviii

CHAPTER 1

Cisco ACI Forwarding 1

  Forwarding Within the Fabric 1
    ACI Fabric Optimizes Modern Data Center Traffic Flows 1
    VXLAN in ACI 2
      Layer 3 VNIDs Facilitate Transporting Inter-subnet Tenant Traffic 4
      WAN and Other External Networks 5
      Networking Domains 5
      Configuring Route Reflectors 6
      Router Peering and Route Distribution 6
      Route Import and Export, Route Summarization, and Route Community Match 7
      ACI Route Redistribution 11
      Route Distribution Within the ACI Fabric 11
      External Layer 3 Outside Connection Types 12
      About the Modes of Configuring Layer 3 External Connectivity 14
      Controls Enabled for Subnets Configured under the L3Out Network Instance Profile 16
      ACI Layer 3 Outside Network Workflows 17

CHAPTER 2

Prerequisites for Configuring Layer 3 Networks 19

  Layer 3 Prerequisites 19
CHAPTER 3

Routed Connectivity to External Networks 21

About Routed Connectivity to Outside Networks 21
Layer 3 Out for Routed Connectivity to External Networks 21
Guidelines for Routed Connectivity to Outside Networks 24
Configuring Layer 3 Outside for Tenant Networks 27
  Configuring a Tenant Layer 3 Outside Network Connection Overview 27
  Configuring Layer 3 Outside for Tenant Networks Using the REST API 29
    REST API Example: L3Out Prerequisites 31
    REST API Example: L3Out 32
  Configuring a Layer 3 Outside for Tenant Networks Using the NX-OS Style CLI 34
    NX-OS Style CLI Example: L3Out Prerequisites 37
    NX-OS Style CLI Example: L3Out 37
  Configuring a Layer 3 Outside for Tenant Networks Using the GUI 39

CHAPTER 4

Layer 3 Routed and Sub-Interface Port Channels 45

About Layer 3 Port Channels 45
Configuring Port Channels Using the GUI 46
Configuring a Layer 3 Routed Port-Channel Using the GUI 47
Configuring a Layer 3 Sub-Interface Port-Channel Using the GUI 49
Configuring a Layer 3 Routed Port-Channel Using the NX-OS CLI 51
Configuring a Layer 3 Sub-Interface Port-Channel Using the NX-OS CLI 53
Adding Ports to the Layer 3 Port-Channel Using the NX-OS CLI 56
Configuring Port Channels Using the REST API 57
Configuring a Layer 3 Routed Port Channel Using the REST API 58
Configuring a Layer 3 Sub-Interface Port Channel Using the REST API 59

CHAPTER 5

QoS for L3Outs 61

L3Outs QoS 61
L3Outs QoS Guidelines and Limitations 61
Configuring QoS Directly on L3Out Using GUI 62
Configuring QoS Directly on L3Out Using CLI 63
Configuring QoS Directly on L3Out Using REST API 64
Troubleshooting Inconsistency and Faults 87
Configuring BFD Support 88
  Bidirectional Forwarding Detection 88
Configuring BFD Globally on Leaf Switch Using the GUI 89
Configuring BFD Globally on Spine Switch Using the GUI 90
Configuring BFD Globally on Leaf Switch Using the NX-OS Style CLI 91
Configuring BFD Globally on Spine Switch Using the NX-OS Style CLI 92
Configuring BFD Globally Using the REST API 93
Configuring BFD Interface Override Using the GUI 93
Configuring BFD Interface Override Using the NX-OS Style CLI 94
Configuring BFD Interface Override Using the REST API 96
Configuring BFD Consumer Protocols Using the GUI 96
Configuring BFD Consumer Protocols Using the NX-OS Style CLI 98
Configuring BFD Consumer Protocols Using the REST API 99

OSPF External Routed Networks 102
  OSPF Layer 3 Outside Connections 102
Creating an OSPF External Routed Network for Management Tenant Using the GUI 104
Creating an OSPF External Routed Network for a Tenant Using the NX-OS CLI 105
Creating OSPF External Routed Network for Management Tenant Using REST API 108

EIGRP External Routed Networks 108
  About EIGRP Layer 3 Outside Connections 109
  EIGRP Protocol Support 109
  Guidelines and Limitations When Configuring EIGRP 111
Configuring EIGRP Using the GUI 111
Configuring EIGRP Using the NX-OS-Style CLI 112
Configuring EIGRP Using the REST API 116

Route Summarization 119
  Route Summarization 119
Configuring Route Summarization for BGP, OSPF, and EIGRP Using the REST API 119
Configuring Route Summarization for BGP, OSPF, and EIGRP Using the NX-OS Style CLI 121
Configuring Route Summarization for BGP, OSPF, and EIGRP Using the GUI 122

Route Control 125
Route Maps/Profiles with Explicit Prefix Lists 125

About Route Map/Profile 125

About Explicit Prefix List Support for Route Maps/Profile 126

Aggregation Support for Explicit Prefix List 128

Guidelines and Limitations 132

Configuring a Route Map/Profile with Explicit Prefix List Using the GUI 132

Configuring Route Map/Profile with Explicit Prefix List Using NX-OS Style CLI 134

Configuring Route Map/Profile with Explicit Prefix List Using REST API 136

Routing Control Protocols 138

About Configuring a Routing Control Protocol Using Import and Export Controls 138

Configuring a Route Control Protocol to Use Import and Export Controls, With the GUI 138

Configuring a Route Control Protocol to Use Import and Export Controls, With the NX-OS Style CLI 140

Configuring a Route Control Protocol to Use Import and Export Controls, With the REST API 141

CHAPTER 9

Common Pervasive Gateway 143

Overview 143

Configuring Common Pervasive Gateway Using the GUI 144

Configuring Common Pervasive Gateway Using the NX-OS Style CLI 146

Configuring Common Pervasive Gateway Using the REST API 146

CHAPTER 10

Static Route on a Bridge Domain 149

About Static Routes in Bridge Domains 149

Configuring a Static Route on a Bridge Domain Using the GUI 149

Configuring a Static Route on a Bridge Domain Using the NX-OS Style CLI 150

Configuring a Static Route on a Bridge Domain Using the REST API 151

CHAPTER 11

MP-BGP Route Reflectors 153

BGP Protocol Peering to External BGP Speakers 153

Configuring an MP-BGP Route Reflector Using the GUI 155

Configuring an MP-BGP Route Reflector for the ACI Fabric 155

Configuring an MP-BGP Route Reflector Using the REST API 156

Verifying the MP-BGP Route Reflector Configuration 156
## CHAPTER 12 - Switch Virtual Interface

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVI External Encapsulation Scope</td>
<td>159</td>
</tr>
<tr>
<td>About SVI External Encapsulation Scope</td>
<td>159</td>
</tr>
<tr>
<td>Encapsulation Scope Syntax</td>
<td>161</td>
</tr>
<tr>
<td>Guidelines for SVI External Encapsulation Scope</td>
<td>161</td>
</tr>
<tr>
<td>Configuring SVI External Encapsulation Scope Using the GUI</td>
<td>162</td>
</tr>
<tr>
<td>Configuring SVI Interface Encapsulation Scope Using NX-OS Style CLI</td>
<td>162</td>
</tr>
<tr>
<td>Configuring SVI Interface Encapsulation Scope Using the REST API</td>
<td>163</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVI Auto State</td>
<td>164</td>
</tr>
<tr>
<td>About SVI Auto State</td>
<td>164</td>
</tr>
<tr>
<td>Guidelines and Limitations for SVI Auto State Behavior</td>
<td>164</td>
</tr>
<tr>
<td>Configuring SVI Auto State Using the GUI</td>
<td>165</td>
</tr>
<tr>
<td>Configuring SVI Auto State Using NX-OS Style CLI</td>
<td>165</td>
</tr>
<tr>
<td>Configuring SVI Auto State Using the REST API</td>
<td>166</td>
</tr>
</tbody>
</table>

## CHAPTER 13 - Shared Services

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Layer 3 Out</td>
<td>167</td>
</tr>
<tr>
<td>Layer 3 Out to Layer 3 Out Inter-VRF Leaking</td>
<td>170</td>
</tr>
<tr>
<td>Configuring Two Shared Layer 3 Outs in Two VRFs Using REST API</td>
<td>171</td>
</tr>
<tr>
<td>Configuring Shared Layer 3 Out Inter-VRF Leaking Using the NX-OS Style CLI - Named Example</td>
<td>172</td>
</tr>
<tr>
<td>Configuring Shared Layer 3 Out Inter-VRF Leaking Using the NX-OS Style CLI - Implicit Example</td>
<td>174</td>
</tr>
<tr>
<td>Configuring Shared Layer 3 Out Inter-VRF Leaking Using the Advanced GUI</td>
<td>175</td>
</tr>
</tbody>
</table>

## CHAPTER 14 - Interleak of External Routes

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>179</td>
</tr>
<tr>
<td>Configuring Interleak of External Routes Using the GUI</td>
<td>179</td>
</tr>
<tr>
<td>Configuring Interleak External Routes Using the NX-OS Style CLI</td>
<td>180</td>
</tr>
<tr>
<td>Configuring Interleak of External Routes Using the REST API</td>
<td>181</td>
</tr>
</tbody>
</table>

## CHAPTER 15 - Dataplane IP Learning per VRF

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>183</td>
</tr>
</tbody>
</table>
CHAPTER 16  
IP Aging 187
Overview 187
Configuring the IP Aging Policy Using the GUI 187
Configuring the IP Aging Policy Using the NX-OS-Style CLI 188
Configuring IP Aging Using the REST API 188

CHAPTER 17  
IPv6 Neighbor Discovery 191
Neighbor Discovery 191
Configuring IPv6 Neighbor Discovery on a Bridge Domain 192
Creating the Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the REST API 192
Configuring a Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the NX-OS Style CLI 193
Creating the Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the GUI 194
Configuring IPv6 Neighbor Discovery on a Layer 3 Interface 195
Guidelines and Limitations 195
Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the GUI 195
Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the REST API 196
Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the NX-OS Style CLI 197
Configuring IPv6 Neighbor Discovery Duplicate Address Detection 200
About Neighbor Discovery Duplicate Address Detection 200
Configuring Neighbor Discovery Duplicate Address Detection Using the REST API 200
Configuring Neighbor Discovery Duplicate Address Detection Using the GUI 201

CHAPTER 18  
IP Multicast 203
Layer 3 Multicast 203
Contents

About the Fabric Interface 204
Enabling Multicast Routing 205
Allocating VRF GiPo 205
Multiple Border Leaf Switches as Designated Forwarder 206
PIM Designated Router Election 207
Non-Border Leaf Switch Behavior 207
Active Border Leaf Switch List 208
Overload Behavior On Bootup 208
First-Hop Functionality 208
The Last-Hop 208
Fast-Convergence Mode 208
About Rendezvous Points 209
About Inter-VRF Multicast 209
  Inter-VRF Multicast Requirements 210
Guidelines and Restrictions for Configuring Layer 3 Multicast 210
Configuring Layer 3 Multicast Using the GUI 212
Configuring Layer 3 Multicast Using the NX-OS Style CLI 214
Configuring Layer 3 Multicast Using REST API 216

CHAPTER 19

IGMP Snooping 219
About Cisco APIC and IGMP Snooping 219
  How IGMP Snooping is Implemented in the ACI Fabric 219
  Virtualization Support 221
  The APIC IGMP Snooping Function, IGMPv1, IGMPv2, and the Fast Leave Feature 221
  The APIC IGMP Snooping Function and IGMPv3 221
  Cisco APIC and the IGMP Snooping Querier Function 222
  Guidelines and Limitations for the APIC IGMP Snooping Function 222
Configuring and Assigning an IGMP Snooping Policy 223
  Configuring and Assigning an IGMP Snooping Policy to a Bridge Domain in the Advanced GUI 223
  Configuring an IGMP Snooping Policy Using the GUI 223
  Assigning an IGMP Snooping Policy to a Bridge Domain Using the GUI 224
  Configuring and Assigning an IGMP Snooping Policy to a Bridge Domain using the NX-OS Style CLI 225
Contents

Supported Transit Combination Matrix 307
Transit Routing Guidelines 309
  Guidelines for Transit Routing 309
  Transit Route Control 314
Scope and Aggregate Controls for Subnets 315
Route Control Profile Policies 317
Security Import Policies 318
Configuring Transit Routing 319
  Transit Routing Overview 319
  Configuring Transit Routing Using the REST API 321
    REST API Example: Transit Routing 324
  Configure Transit Routing Using the NX-OS Style CLI 326
    Example: Transit Routing 330
  Configure Transit Routing Using the GUI 332
Preface

This preface includes the following sections:

- Audience, on page xv
- New and Changed Information, on page xv
- Document Conventions, on page xvi
- Related Documentation, on page xvii
- Documentation Feedback, on page xviii

Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Virtual machine installation and administration
- Server administration
- Switch and network administration
- Cloud administration

New and Changed Information

The following table provides an overview of the significant changes to the organization and features in this guide up to this current release. The table does not provide an exhaustive list of all changes made to the guide or of the new features up to this release.

**Table 1: New Features and Changed Behavior in Cisco APIC Release 4.0(1)**

<table>
<thead>
<tr>
<th>Feature or Change</th>
<th>Description</th>
<th>Where Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertise Host Route configuration</td>
<td>Support for advertising host route configuration on a border leaf</td>
<td>Routed Connectivity to External Networks, on page 21</td>
</tr>
<tr>
<td>Disable dataplane IP learning per VRF</td>
<td>Allows IP learning through the dataplane to be disabled on a VRF.</td>
<td>Dataplane IP Learning per VRF, on page 183</td>
</tr>
</tbody>
</table>
Document Conventions

Command descriptions use the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bold</td>
<td>Bold text indicates the commands and keywords that you enter literally as shown.</td>
</tr>
<tr>
<td>Italic</td>
<td>Italic text indicates arguments for which the user supplies the values.</td>
</tr>
<tr>
<td>[x]</td>
<td>Square brackets enclose an optional element (keyword or argument).</td>
</tr>
<tr>
<td>[x</td>
<td>y]</td>
</tr>
<tr>
<td>{x</td>
<td>y}</td>
</tr>
<tr>
<td>[x {y</td>
<td>z} ]</td>
</tr>
<tr>
<td>variable</td>
<td>Indicates a variable for which you supply values, in context where italics cannot be used.</td>
</tr>
</tbody>
</table>
### Convention

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.</td>
</tr>
</tbody>
</table>

Examples use the following conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>screen font</td>
<td>Terminal sessions and information the switch displays are in screen font.</td>
</tr>
<tr>
<td>boldface screen font</td>
<td>Information you must enter is in boldface screen font.</td>
</tr>
<tr>
<td>italic screen font</td>
<td>Arguments for which you supply values are in italic screen font.</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Nonprinting characters, such as passwords, are in angle brackets.</td>
</tr>
<tr>
<td>[]</td>
<td>Default responses to system prompts are in square brackets.</td>
</tr>
<tr>
<td>!, #</td>
<td>An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.</td>
</tr>
</tbody>
</table>

This document uses the following conventions:

- **Note**
  - Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

- **Caution**
  - Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

- **Warning**
  - IMPORTANT SAFETY INSTRUCTIONS
    - This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.
    - SAVE THESE INSTRUCTIONS

### Related Documentation

**Application Policy Infrastructure Controller (APIC) Documentation**

The following companion guides provide documentation for APIC:

- *Cisco APIC Getting Started Guide*
- *Cisco APIC Basic Configuration Guide*
Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to apic-docfeedback@cisco.com. We appreciate your feedback.
This chapter contains the following sections:

- Forwarding Within the Fabric, on page 1
- WAN and Other External Networks, on page 5

Forwarding Within the Fabric

ACI Fabric Optimizes Modern Data Center Traffic Flows

The Cisco ACI architecture addresses the limitations of traditional data center design, and provides support for the increased east-west traffic demands of modern data centers.

Today, application design drives east-west traffic from server to server through the data center access layer. Applications driving this shift include big data distributed processing designs like Hadoop, live virtual machine or workload migration as with VMware vMotion, server clustering, and multi-tier applications.

North-south traffic drives traditional data center design with core, aggregation, and access layers, or collapsed core and access layers. Client data comes in from the WAN or Internet, a server processes it, and then it exits the data center, which permits data center hardware oversubscription due to WAN or Internet bandwidth constraints. However, Spanning Tree Protocol is required to block loops. This limits available bandwidth due to blocked links, and potentially forces traffic to take a suboptimal path.

In traditional data center designs, IEEE 802.1Q VLANs provide logical segmentation of Layer 2 boundaries or broadcast domains. However, VLAN use of network links is inefficient, requirements for device placements in the data center network can be rigid, and the VLAN maximum of 4094 VLANs can be a limitation. As IT departments and cloud providers build large multi-tenant data centers, VLAN limitations become problematic.

A spine-leaf architecture addresses these limitations. The ACI fabric appears as a single switch to the outside world, capable of bridging and routing. Moving Layer 3 routing to the access layer would limit the Layer 2 reachability that modern applications require. Applications like virtual machine workload mobility and some clustering software require Layer 2 adjacency between source and destination servers. By routing at the access layer, only servers connected to the same access switch with the same VLANs trunked down would be Layer 2-adjacent. In ACI, VXLAN solves this dilemma by decoupling Layer 2 domains from the underlying Layer 3 network infrastructure.
As traffic enters the fabric, ACI encapsulates and applies policy to it, forwards it as needed across the fabric through a spine switch (maximum two-hops), and de-encapsulates it upon exiting the fabric. Within the fabric, ACI uses Intermediate System-to-Intermediate System Protocol (IS-IS) and Council of Oracle Protocol (COOP) for all forwarding of endpoint to endpoint communications. This enables all ACI links to be active, equal cost multipath (ECMP) forwarding in the fabric, and fast-reconverging. For propagating routing information between software defined networks within the fabric and routers external to the fabric, ACI uses the Multiprotocol Border Gateway Protocol (MP-BGP).

**VXLAN in ACI**

VXLAN is an industry-standard protocol that extends Layer 2 segments over Layer 3 infrastructure to build Layer 2 overlay logical networks. The ACI infrastructure Layer 2 domains reside in the overlay, with isolated broadcast and failure bridge domains. This approach allows the data center network to grow without the risk of creating too large a failure domain.

All traffic in the ACI fabric is normalized as VXLAN packets. At ingress, ACI encapsulates external VLAN, VXLAN, and NVGRE packets in a VXLAN packet. The following figure shows ACI encapsulation normalization.
Forwarding in the ACI fabric is not limited to or constrained by the encapsulation type or encapsulation overlay network. An ACI bridge domain forwarding policy can be defined to provide standard VLAN behavior where required.

Because every packet in the fabric carries ACI policy attributes, ACI can consistently enforce policy in a fully distributed manner. ACI decouples application policy EPG identity from forwarding. The following illustration shows how the ACI VXLAN header identifies application policy within the fabric.

The ACI VXLAN packet contains both Layer 2 MAC address and Layer 3 IP address source and destination fields, which enables efficient and scalable forwarding within the fabric. The ACI VXLAN packet header source group field identifies the application policy endpoint group (EPG) to which the packet belongs. The VXLAN Instance ID (VNID) enables forwarding of the packet through tenant virtual routing and forwarding (VRF) domains within the fabric. The 24-bit VNID field in the VXLAN header provides an expanded address space for up to 16 million unique Layer 2 segments in the same network. This expanded address space gives IT departments and cloud providers greater flexibility as they build large multitenant data centers.

VXLAN enables ACI to deploy Layer 2 virtual networks at scale across the fabric underlay Layer 3 infrastructure. Application endpoint hosts can be flexibly placed in the data center network without concern for the Layer 3 boundary of the underlay infrastructure, while maintaining Layer 2 adjacency in a VXLAN overlay network.
Layer 3 VNIDs Facilitate Transporting Inter-subnet Tenant Traffic

The ACI fabric provides tenant default gateway functionality that routes between the ACI fabric VXLAN networks. For each tenant, the fabric provides a virtual default gateway that spans all of the leaf switches assigned to the tenant. It does this at the ingress interface of the first leaf switch connected to the endpoint. Each ingress interface supports the default gateway interface. All of the ingress interfaces across the fabric share the same router IP address and MAC address for a given tenant subnet.

The ACI fabric decouples the tenant endpoint address, its identifier, from the location of the endpoint that is defined by its locator or VXLAN tunnel endpoint (VTEP) address. Forwarding within the fabric is between VTEPs. The following figure shows decoupled identity and location in ACI.

Figure 4: ACI Decouples Identity and Location

VXLAN uses VTEP devices to map tenant end devices to VXLAN segments and to perform VXLAN encapsulation and de-encapsulation. Each VTEP function has two interfaces:

- A switch interface on the local LAN segment to support local endpoint communication through bridging
- An IP interface to the transport IP network

The IP interface has a unique IP address that identifies the VTEP device on the transport IP network known as the infrastructure VLAN. The VTEP device uses this IP address to encapsulate Ethernet frames and transmit the encapsulated packets to the transport network through the IP interface. A VTEP device also discovers the remote VTEPs for its VXLAN segments and learns remote MAC Address-to-VTEP mappings through its IP interface.

The VTEP in ACI maps the internal tenant MAC or IP address to a location using a distributed mapping database. After the VTEP completes a lookup, the VTEP sends the original data packet encapsulated in VXLAN with the destination address of the VTEP on the destination leaf switch. The destination leaf switch de-encapsulates the packet and sends it to the receiving host. With this model, ACI uses a full mesh, single hop, loop-free topology without the need to use the spanning-tree protocol to prevent loops.

The VXLAN segments are independent of the underlying network topology; conversely, the underlying IP network between VTEPs is independent of the VXLAN overlay. It routes the encapsulated packets based on the outer IP address header, which has the initiating VTEP as the source IP address and the terminating VTEP as the destination IP address.

The following figure shows how routing within the tenant is done.
For each tenant VRF in the fabric, ACI assigns a single L3 VNID. ACI transports traffic across the fabric according to the L3 VNID. At the egress leaf switch, ACI routes the packet from the L3 VNID to the VNID of the egress subnet.

Traffic arriving at the fabric ingress that is sent to the ACI fabric default gateway is routed into the Layer 3 VNID. This provides very efficient forwarding in the fabric for traffic routed within the tenant. For example, with this model, traffic between 2 VMs belonging to the same tenant, on the same physical host, but on different subnets, only needs to travel to the ingress switch interface before being routed (using the minimal path cost) to the correct destination.

To distribute external routes within the fabric, ACI route reflectors use multiprotocol BGP (MP-BGP). The fabric administrator provides the autonomous system (AS) number and specifies the spine switches that become route reflectors.

**WAN and Other External Networks**

**Networking Domains**

A fabric administrator creates domain policies that configure ports, protocols, VLAN pools, and encapsulation. These policies can be used exclusively by a single tenant, or shared. Once a fabric administrator configures domains in the ACI fabric, tenant administrators can associate tenant endpoint groups (EPGs) to domains.

The following networking domain profiles can be configured:

- VMM domain profiles (`vmmDomP`) are required for virtual machine hypervisor integration.
- Physical domain profiles (`physDomP`) are typically used for bare metal server attachment and management access.
- Bridged outside network domain profiles (`l2extDomP`) are typically used to connect a bridged external network trunk switch to a leaf switch in the ACI fabric.
- Routed outside network domain profiles (`l3extDomP`) are used to connect a router to a leaf switch in the ACI fabric.
• Fibre Channel domain profiles (fcDomP) are used to connect Fibre Channel VLANs and VSANs.

A domain is configured to be associated with a VLAN pool. EPGs are then configured to use the VLANs associated with a domain.

---

**Note**

EPG port and VLAN configurations must match those specified in the domain infrastructure configuration with which the EPG associates. If not, the APIC will raise a fault. When such a fault occurs, verify that the domain infrastructure configuration matches the EPG port and VLAN configurations.

---

### Configuring Route Reflectors

The ACI fabric route reflectors use multiprotocol BGP (MP-BGP) to distribute external routes within the fabric. To enable route reflectors in the ACI fabric, the fabric administrator must select the spine switches that will be the route reflectors, and provide the autonomous system (AS) number. Once route reflectors are enabled in the ACI fabric, administrators can configure connectivity to external networks as described in the following sections.

To connect external routers to the ACI fabric, the fabric infrastructure administrator configures spine nodes as Border Gateway Protocol (BGP) route reflectors. For redundancy purposes, more than one spine is configured as a router reflector node (one primary and one secondary reflector).

When a tenant needs to attach a WAN router to the ACI fabric, the infrastructure administrator configures the leaf node (as described below) to which the WAN router is being connected as WAN top of rack (ToR) and pairs this WAN ToR with one of the route reflector nodes as a BGP peer. When route reflectors are configured on the WAN ToR, they are able to advertise the tenant routes into the fabric.

Each leaf node can store up to 4000 routes. If a WAN router has to advertise more than 4000 routes, it should peer with multiple leaf nodes. The infrastructure administrator configures each of the paired leaf nodes with the routes (or route prefixes) that it can advertise.

The infrastructure administrator must configure an external WAN router connected to the fabric as follows:

1. Configure up to two spine nodes as route reflectors. For redundancy, configure primary and secondary route reflectors.
2. On WAN ToRs, configure the primary and secondary route reflector nodes.
3. On WAN ToRs, configure the routes that the ToR is responsible for advertising. This is optional and needs to be done only when the tenant router is known to advertise more than 4000 routes.

### Router Peering and Route Distribution

As shown in the figure below, when the routing peer model is used, the leaf switch interface is statically configured to peer with the external router’s routing protocol.
The routes that are learned through peering are sent to the spine switches. The spine switches act as route reflectors and distribute the external routes to all of the leaf switches that have interfaces that belong to the same tenant. These routes are longest prefix match (LPM) summarized addresses and are placed in the leaf switch's forwarding table with the VTEP IP address of the remote leaf switch where the external router is connected. WAN routes have no forwarding proxy. If the WAN routes do not fit in the leaf switch's forwarding table, the traffic is dropped. Because the external router is not the default gateway, packets from the tenant endpoints (EPs) are sent to the default gateway in the ACI fabric.

**Route Import and Export, Route Summarization, and Route Community Match**

Subnet route export or import configuration options can be specified according to the scope and aggregation options described below.

For routed subnets, the following scope options are available:

- **Export Route Control Subnet**—Controls the export route direction.
- **Import Route Control Subnet**—Controls the import route direction.

| Note | Import route control is supported for BGP and OSPF, but not EIGRP. |
• External Subnets for the External EPG (Security Import Subnet)—Specifies which external subnets have contracts applied as part of a specific External Network Instance Profile (l3extInstP). For a subnet under the l3extInstP to be classified as an External EPG, the scope on the subnet should be set to "import-security". Subnets of this scope determine which IP addresses are associated with the l3extInstP. Once this is determined, contracts determine with which other EPGs that external subnet is allowed to communicate. For example, when traffic enters the ACI switch on the Layer 3 External Outside Network (L3extOut), a lookup occurs to determine which source IP addresses are associated with the l3extInstP. This action is performed based on Longest Prefix Match (LPM) so that more specific subnets take precedence over more general subnets.

• Shared Route Control Subnet—In a shared service configuration, only subnets that have this property enabled will be imported into the consumer EPG Virtual Routing and Forwarding (VRF). It controls the route direction for shared services between VRFs.

• Shared Security Import Subnet—Applies shared contracts to imported subnets. The default specification is External Subnets for the External EPG.

Routed subnets can be aggregated. When aggregation is not set, the subnets are matched exactly. For example, if 11.1.0.0/16 is the subnet, then the policy will not apply to a 11.1.1.0/24 route, but it will apply only if the route is 11.1.0.0/16. However, to avoid a tedious and error prone task of defining all the subnets one by one, a set of subnets can be aggregated into one export, import or shared routes policy. At this time, only 0/0 subnets can be aggregated. When 0/0 is specified with aggregation, all the routes are imported, exported, or shared with a different VRF, based on the selection option below:

• Aggregate Export—Exports all transit routes of a VRF (0/0 subnets).

• Aggregate Import—Imports all incoming routes of given L3 peers (0/0 subnets).

Note

Aggregate import route control is supported for BGP and OSPF, but not for EIGRP.

• Aggregate Shared Routes—If a route is learned in one VRF but needs to be advertised to another VRF, the routes can be shared by matching the subnet exactly, or can be shared in an aggregate way according to a subnet mask. For aggregate shared routes, multiple subnet masks can be used to determine which specific route groups are shared between VRFs. For example, 10.1.0.0/16 and 12.1.0.0/16 can be specified to aggregate these subnets. Or, 0/0 can be used to share all subnet routes across multiple VRFs.

Note

Routes shared between VRFs function correctly on Generation 2 switches (Cisco Nexus N9K switches with "EX" or "FX" on the end of the switch model name, or later; for example, N9K-93108TC-EX). On Generation 1 switches, however, there may be dropped packets with this configuration, because the physical ternary content-addressable memory (TCAM) tables that store routes do not have enough capacity to fully support route parsing.

Route summarization simplifies route tables by replacing many specific addresses with an single address. For example, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 are replaced with 10.1.0.0/16. Route summarization policies enable routes to be shared efficiently among border leaf switches and their neighbor leaf switches. BGP, OSPF, or EIGRP route summarization policies are applied to a bridge domain or transit subnet. For OSPF, inter-area and external route summarization are supported. Summary routes are exported; they are not advertised
within the fabric. In the example above, when a route summarization policy is applied, and an EPG uses the 10.1.0.0/16 subnet, the entire range of 10.1.0.0/16 is shared with all the neighboring leaf switches.

**Note**

When two L3extOut policies are configured with OSPF on the same leaf switch, one regular and another for the backbone, a route summarization policy configured on one L3extOut is applied to both L3extOut policies because summarization applies to all areas in the VRF.

As illustrated in the figure below, route control profiles derive route maps according to prefix-based and community-based matching.

*Figure 7: Route Community Matching*

The route control profile (`rtctrtlProfile`) specifies what is allowed. The Route Control Context specifies what to match, and the scope specifies what to set. The subject profile contains the community match specifications, which can be used by multiple L3extOut instances. The subject profile (`SubjP`) can contain multiple community terms each of which contains one or more community factors (communities). This arrangement enables specifying the following boolean operations:

- Logical or among multiple community terms
- Logical and among multiple community factors

For example, a community term called northeast could have multiple communities that each include many routes. Another community term called southeast could also include many different routes. The administrator could choose to match one, or the other, or both. A community factor type can be regular or extended. Care should be taken when using extended type community factors, to ensure there are no overlaps among the specifications.
The scope portion of the route control profile references the attribute profile (rtctrlAttrP) to specify what set-action to apply, such as preference, next hop, community, and so forth. When routes are learned from an l3extOut, route attributes can be modified.

The figure above illustrates the case where an l3extOut contains a rtctrlProfile. A rtctrlProfile can also exist under the tenant. In this case, the l3extOut has an interleave relation policy (L3extRsInterleakPol) that associates it with the rtctrlProfile under the tenant. This configuration enables reusing the rtctrlProfile for multiple l3extOut connections. It also enables keeping track of the routes the fabric learns from OSPF to which it gives BGP attributes (BGP is used within the fabric). A rtctrlProfile defined under an L3extOut has a higher priority than one defined under the tenant.

The rtctrlProfile has two modes: combinable, and global. The default combinable mode combines pervasive subnets (fvSubnet) and external subnets (l3extSubnet) with the match/set mechanism to render the route map. The global mode applies to all subnets within the tenant, and overrides other policy attribute settings. A global rtctrlProfile provides permit-all behavior without defining explicit (0/0) subnets. A global rtctrlProfile is used with non-prefix based match rules where matching is done using different subnet attributes such as community, next hop, and so on. Multiple rtctrlProfile policies can be configured under a tenant.

rtctrlProfile policies enable enhanced default import and default export route control. Layer 3 Outside networks with aggregated import or export routes can have import/export policies that specify supported default-export and default-import, and supported 0/0 aggregation policies. To apply a rtctrlProfile policy on all routes (inbound or outbound), define a global default rtctrlProfile that has no match rules.

**Note**

While multiple l3extOut connections can be configured on one switch, all Layer 3 outside networks configured on a switch must use the same rtctrlProfile because a switch can have only one route map.

The protocol interleave and redistribute policy controls externally learned route sharing with ACI fabric BGP routes. Set attributes are supported. Such policies are supported per L3extOut, per node, or per VRF. An interleave policy applies to routes learned by the routing protocol in the L3extOut. Currently, interleave and redistribute policies are supported for OSPF v2 and v3. A route control policy rtctrlProfile has to be defined as global when it is consumed by an interleave policy.
ACI Route Redistribution

The routes that are learned from the OSPF process on the border leaf are redistributed into BGP for the tenant VRF and they are imported into MP-BGP on the border leaf.

- Import route control is supported for BGP and OSPF, but not for EIGRP.
- Export route control is supported for OSPF, BGP, and EIGRP.
- The routes are learned on the border leaf where the VRF is deployed. The routes are not advertised to the External Layer 3 Outside connection unless it is permitted by the export route control.

Note

When a subnet for a bridge domain/EPG is set to Advertise Externally, the subnet is programmed as a static route on a border leaf. When the static route is advertised, it is redistributed into the EPG’s Layer 3 outside network routing protocol as an external network, not injected directly into the routing protocol.

Route Distribution Within the ACI Fabric

ACI supports the following routing mechanisms:

- Static Routes
- OSPFv2 (IPv4)
- OSPFv3 (IPv6)
- iBGP
- eBGP (IPv4 and IPv6)
• EIGRP (IPv4 and IPv6) protocols

ACI supports the VRF-lite implementation when connecting to the external routers. Using sub-interfaces, the border leaf can provide Layer 3 outside connections for the multiple tenants with one physical interface. The VRF-lite implementation requires one protocol session per tenant.

Within the ACI fabric, Multiprotocol BGP (MP-BGP) is implemented between the leaf and the spine switches to propagate the external routes within the ACI fabric. The BGP route reflector technology is deployed in order to support a large number of leaf switches within a single fabric. All of the leaf and spine switches are in one single BGP Autonomous System (AS). Once the border leaf learns the external routes, it can then redistribute the external routes of a given VRF to an MP-BGP address family VPN version 4 or VPN version 6. With address family VPN version 4, MP-BGP maintains a separate BGP routing table for each VRF. Within MP-BGP, the border leaf advertises routes to a spine switch, that is a BGP route reflector. The routes are then propagated to all the leaves where the VRFs (or private network in the APIC GUI’s terminology) are instantiated.

External Layer 3 Outside Connection Types

ACI supports the following External Layer 3 Outside connection options:
• Static Routing (supported for IPv4 and IPv6)
• OSPFv2 for normal and NSSA areas (IPv4)
• OSPFv3 for normal and NSSA areas (IPv6)
• iBGP (IPv4 and IPv6)
• eBGP (IPv4 and IPv6)
• EIGRP (IPv4 and IPv6)

The External Layer 3 Outside connections are supported on the following interfaces:
• Layer 3 Routed Interface
• Subinterface with 802.1Q tagging - With subinterface, you can use the same physical interface to provide a Layer 2 outside connection for multiple private networks.
• Switched Virtual Interface (SVI) - With an SVI interface, the same physical interface that supports Layer 2 and Layer 3 and the same physical interface can be used for a Layer 2 outside connection and a Layer 3 outside connection.
The managed objects that are used for the L3Outside connections are:

- **External Layer 3 Outside (L3ext):** Routing protocol options (OSPF area type, area, EIGRP autonomous system, BGP), private network, External Physical domain.

- **Logical Node Profile:** Profile where one or more nodes are defined for the External Layer 3 Outside connections. The configurations of the router-IDs and the loopback interface are defined in the profile.

  **Note**
  
  Use the same router-ID for the same node across multiple External Layer 3 Outside connections.

  **Note**
  
  Within a single L3Out, a node can only be part of one Logical Node Profile. Configuring the node to be a part of multiple Logical Node Profiles in a single L3Out might result in unpredictable behavior, such as having a loopback address pushed from one Logical Node Profile but not from the other. Use more path bindings under the existing Logical Interface Profiles or create a new Logical Interface Profile under the existing Logical Node Profile instead.
• Logical Interface Profile: IP interface configuration for IPv4 and IPv6 interfaces. It is supported on the Route Interfaces, Routed subinterfaces, and SVIs. The SVIs can be configured on physical ports, port-channels, or vPCs.

• OSPF Interface Policy: Includes details such as OSPF Network Type and priority.

• EIGRP Interface Policy: Includes details such as Timers and split horizon.

• BGP Peer Connectivity Profile: The profile where most BGP peer settings, remote-as, local-as, and BGP peer connection options are configured. You can associate the BGP peer connectivity profile with the logical interface profile or the loopback interface under the node profile. This determines the update-source configuration for the BGP peering session.

• External Network Instance Profile (EPG) (l3extInstP): The external EPG is also referred to as the prefix-based EPG or InstP. The import and export route control policies, security import policies, and contract associations are defined in this profile. You can configure multiple external EPGs under a single L3Out. You may use multiple external EPGs when a different route or a security policy is defined on a single External Layer 3 Outside connections. An external EPG or multiple external EPGs combine into a route-map. The import/export subnets defined under the external EPG associate to the IP prefix-list match clauses in the route-map. The external EPG is also where the import security subnets and contracts are associated. This is used to permit or drop traffic for this L3out.

• Action Rules Profile: The action rules profile is used to define the route-map set clauses for the L3Out. The supported set clauses are the BGP communities (standard and extended), Tags, Preference, Metric, and Metric type.

• Route Control Profile: The route-control profile is used to reference the action rules profiles. This can be an ordered list of action rules profiles. The Route Control Profile can be referenced by a tenant BD, BD subnet, external EPG, or external EPG subnet.

There are more protocol settings for BGP, OSPF, and EIGRP L3Outs. These settings are configured per tenant in the ACI Protocol Policies section in the GUI.

---

**Note**

When configuring policy enforcement between external EPGs (transit routing case), you must configure the second external EPG (InstP) with the default prefix 0/0 for export route control, aggregate export, and external security. In addition, you must exclude the preferred group, and you must use an any contract (or desired contract) between the transit InstPs.

---

**About the Modes of Configuring Layer 3 External Connectivity**

Because APIC supports multiple user interfaces (UIs) for configuration, the potential exists for unintended interactions when you create a configuration with one UI and later modify the configuration with another UI. This section describes considerations for configuring Layer 3 external connectivity with the APIC NX-OS style CLI, when you may also be using other APIC user interfaces.

When you configure Layer 3 external connectivity with the APIC NX-OS style CLI, you have the choice of two modes:

• Implicit mode, a simpler mode, is not compatible with the APIC GUI or the REST API.

• Named (or Explicit) mode is compatible with the APIC GUI and the REST API.
In either case, the configuration should be considered read-only in the incompatible UI.

How the Modes Differ

In both modes, the configuration settings are defined within an internal container object, the "L3 Outside" (or "L3Out"), which is an instance of the \texttt{l3extOut} class in the API. The main difference between the two modes is in the naming of this container object instance:

- Implicit mode—the naming of the container is implicit and does not appear in the CLI commands. The CLI creates and maintains these objects internally.

- Named mode—the naming is provided by the user. CLI commands in the Named Mode have an additional \texttt{l3Out} field. To configure the named L3Out correctly and avoid faults, the user is expected to understand the API object model for external Layer 3 configuration.

Note

Except for the procedures in the \textit{Configuring Layer 3 External Connectivity Using the Named Mode} section, this guide describes Implicit mode procedures.

Guidelines and Restrictions

- In the same APIC instance, both modes can be used together for configuring Layer 3 external connectivity with the following restriction: The Layer 3 external connectivity configuration for a given combination of tenant, VRF, and leaf can be done only through one mode.

- For a given tenant VRF, the policy domain where the External-I3 EPG can be placed can be in either the Named mode or in the Implicit mode. The recommended configuration method is to use only one mode for a given tenant VRF combination across all the nodes where the given tenant VRF is deployed for Layer 3 external connectivity. The modes can be different across different tenants or different VRFs and no restrictions apply.

- In some cases, an incoming configuration to a Cisco APIC cluster will be validated against inconsistencies, where the validations involve externally-visible configurations (northbound traffic through the L3Outs). An Invalid Configuration error message will appear for those situations where the configuration is invalid.

- The external Layer 3 features are supported in both configuration modes, with the following exception:
  - Route-peering and Route Health Injection (RHI) with a L4-L7 Service Appliance is supported only in the Named mode. The Named mode should be used across all border leaf switches for the tenant VRF where route-peering is involved.

- Layer 3 external network objects (l3extOut) created using the Implicit mode CLI procedures are identified by names starting with “\_\_ui\_” and are marked as read-only in the GUI. The CLI partitions these external-I3 networks by function, such as interfaces, protocols, route-map, and EPG. Configuration modifications performed through the REST API can break this structure, preventing further modification through the CLI.

For the steps to remove such objects, see \textit{Troubleshooting Unwanted \_\_ui\_ Objects} in the \textit{APIC Troubleshooting Guide}.
Controls Enabled for Subnets Configured under the L3Out Network Instance Profile

The following controls can be enabled for the subnets that are configured under the L3Out Network Instance Profile.

Table 2: Route Control Options

<table>
<thead>
<tr>
<th>Route control Setting</th>
<th>Use</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export Route Control</td>
<td>Controls which external networks are advertised out of the fabric using route-maps and IP prefix lists. An IP prefix list is created on the BL switch for each subnet that is defined. The export control policy is enabled by default and is supported for BGP, EIGRP, and OSPF.</td>
<td>Specific match (prefix and prefix length).</td>
</tr>
<tr>
<td>Import Route Control</td>
<td>Controls the subnets that are allowed into the fabric. Can include set and match rules to filter routes. Supported for BGP and OSPF, but not for EIGRP. If you enable the import control policy for an unsupported protocol, it is automatically ignored. The import control policy is not enabled by default, but you can enable it on the Create Routed Outside panel. On the Identity tab, enable Route Control Enforcement: Import.</td>
<td>Specific match (prefix and prefix length).</td>
</tr>
<tr>
<td>Security Import Subnet</td>
<td>Used to permit the packets to flow between two prefix-based EPGs. Implemented with ACLs.</td>
<td>Uses the ACL match prefix or wildcard match rules.</td>
</tr>
<tr>
<td>Aggregate Export</td>
<td>Used to allow all prefixes to be advertised to the external peers. Implemented with the 0.0.0.0/le32 IP prefix-list.</td>
<td>Only supported for 0.0.0.0/0 subnet (all prefixes).</td>
</tr>
<tr>
<td>Aggregate Import</td>
<td>Used to allow all prefixes that are inbound from an external BGP peer. Implemented with the 0.0.0.0/0 le32 IP prefix-list.</td>
<td>Only supported for the 0.0.0.0/0 subnet (all prefixes).</td>
</tr>
</tbody>
</table>

You may prefer to advertise all the transit routes out of an L3Out connection. In this case, use the aggregate export option with the prefix 0.0.0.0/0. Using this aggregate export option creates an IP prefix-list entry (permit 0.0.0.0/0 le 32) that the APIC system uses as a match clause in the export route-map. Use the `show route-map <outbound route-map>` and `show ip prefix-list <match-clause>` commands to view the output.
If you enable aggregate shared routes, if a route learned in one VRF must be advertised to another VRF, the routes can be shared by matching the subnet exactly, or they can be shared by using an aggregate subnet mask. Multiple subnet masks can be used to determine which specific route groups are shared between VRFs. For example, 10.1.0.0/16 and 12.1.0.0/16 can be specified to aggregate these subnets. Or, 0/0 can be used to share all subnet routes across multiple VRFs.

Note

Routes shared between VRFs function correctly on Generation 2 switches (Cisco Nexus N9K switches with "EX" or "FX" on the end of the switch model name, or later; for example, N9K-93108TC-EX). On Generation 1 switches, however, there may be dropped packets with this configuration, because the physical ternary content-addressable memory (TCAM) tables that store routes do not have enough capacity to fully support route parsing.

ACI Layer 3 Outside Network Workflows

This workflow provides an overview of the steps required to configure a Layer 3 Outside (L3Out) network connection.

1. Prerequisites

- Ensure that you have read/write access privileges to the infra security domain.
- Ensure that the target leaf switches with the necessary interfaces are available.
Configure a Layer 3 Outside Network

Choose which of these L3Out scenarios you will use:

- For an L3Out that will be consumed within a single tenant, follow the instructions for configuring BGP or OSPF.
- For an L3Out that will be consumed (shared) among multiple tenants, follow the "Shared Layer 3 Out" guidelines.
- For an L3Out transit routing use case, follow ACI transit routing instructions.

**Note:** This feature requires APIC release 1.2(1x) or later.
Prerequisites for Configuring Layer 3 Networks

This chapter contains the following sections:

• Layer 3 Prerequisites, on page 19

Layer 3 Prerequisites

Before you begin to perform the tasks in this guide, complete the following:

• Ensure that the ACI fabric and the APIC controllers are online, and the APIC cluster is formed and healthy—For more information, see Cisco APIC Getting Started Guide, Release 2.x.

• Ensure that fabric administrator accounts for the administrators that will configure Layer 3 networks are available—For instructions, see the User Access, Authentication, and Accounting and Management chapters in Cisco APIC Basic Configuration Guide.

• Ensure that the target leaf and spine switches (with the necessary interfaces) are available—For more information, see Cisco APIC Getting Started Guide, Release 2.x.

For information about installing and registering virtual switches, see Cisco ACI Virtualization Guide.

• Configure the tenants, bridge domains, VRFs, and EPGs (with application profiles and contracts) that will consume the Layer 3 networks—For instructions, see the Basic User Tenant Configuration chapter in Cisco APIC Basic Configuration Guide.

• Configure NTP, DNS Service, and DHCP Relay policies—For instructions, see the Provisioning Core ACI Fabric Services chapter in Cisco APIC Basic Configuration Guide, Release 2.x.

⚠️ Caution

If you install 1 Gigabit Ethernet (GE) or 10GE links between the leaf and spine switches in the fabric, there is risk of packets being dropped instead of forwarded, because of inadequate bandwidth. To avoid the risk, use 40GE or 100GE links between the leaf and spine switches.

Bridge Domain Configurations

The Layer 3 Configurations tab of the bridge domain panel allows the administrator to configure the following parameters:
• **Unicast Routing:** If this setting is enabled and a subnet address is configured, the fabric provides the default gateway function and routes the traffic. Enabling unicast routing also instructs the mapping database to learn the endpoint IP-to-VTEP mapping for this bridge domain. The IP learning is not dependent upon having a subnet configured under the bridge domain.

• **Subnet Address:** This option configures the SVI IP addresses (default gateway) for the bridge domain.

• **Limit IP Learning to Subnet:** This option is similar to a unicast reverse-forwarding-path check. If this option is selected, the fabric will not learn IP addresses from a subnet other than the one configured on the bridge domain.

---

**Caution**

Enabling **Limit IP Learning to Subnet** is disruptive to the traffic in the bridge domain.
About Routed Connectivity to Outside Networks

A Layer 3 outside network configuration (L3Out) defines how traffic is forwarded outside of the fabric. Layer 3 is used to discover the addresses of other nodes, select routes, select quality of service, and forward the traffic that is entering, exiting, and transiting the fabric.

Note

For guidelines and cautions for configuring and maintaining Layer 3 outside connections, see Guidelines for Routed Connectivity to Outside Networks, on page 24.

For information about the types of L3Outs, see External Layer 3 Outside Connection Types, on page 12.

Layer 3 Out for Routed Connectivity to External Networks

Routed connectivity to external networks is enabled by associating a fabric access (infraInfra) external routed domain (l3extDomP) with a tenant Layer 3 external instance profile (l3extInstP or external EPG) of a Layer 3 external outside network (l3extOut), in the hierarchy in the following diagram:
A Layer 3 external outside network (l3extOut object) includes the routing protocol options (BGP, OSPF, or EIGRP or supported combinations) and the switch-specific and interface-specific configurations. While the l3extOut contains the routing protocol (for example, OSPF with its related Virtual Routing and Forwarding (VRF) and area ID), the Layer 3 external interface profile contains the necessary OSPF interface details. Both are needed to enable OSPF.

The l3extInstP EPG exposes the external network to tenant EPGs through a contract. For example, a tenant EPG that contains a group of web servers could communicate through a contract with the l3extInstP EPG according to the network configuration contained in the l3extOut. The outside network configuration can easily be reused for multiple nodes by associating the nodes with the L3 external node profile. Multiple nodes that use the same profile can be configured for fail-over or load balancing. Also, a node can be added to multiple l3extOuts resulting in VRFs that are associated with the l3extOuts also being deployed on that node. For scalability information, refer to the current Verified Scalability Guide for Cisco ACI.
Advertise Host Routes

Enabling Advertise Host Routes on the BD, individual host-routes (/32 and /128 prefixes) are advertised from the Border-Leaf switches (BL). The BD must be associated to the L3out or an explicit prefix list matching the host routes. The host routes must be configured to advertise host routes out of the fabric.

Border-Leaf switches along with the subnet advertise the individual end-point (EP) prefixes. The route information is advertised only if the host is connected to the local POD. If the EP is moved away from the local POD or once the EP is removed from EP database (even if the EP is attached to a remote leaf), the route advertisement is then withdrawn.

Advertise Host Route configuration guidelines and limitations are:

- The Advertise Host Routes feature is supported on Generation 2 switches or later (Cisco Nexus N9K switches with "EX", "FX", or "FX2" on the end of the switch model name or later; for example, N9K-93108TC-EX).

- Enabling PIMv4 (Protocol-Independent Multicast, version 4) and Advertise Host routes on a BD is not supported.

- Host route advertisement supports both BD to L3out Association and the explicit route map configurations. We recommend using explicit route map configuration which allows you greater control in selecting individual or a range of host routes to configure.

- EPs/Host routes in SITE-1 will not be advertised out through Border Leafs in other SITEs.

- When EPs is aged out or removed from the database, Host routes are withdrawn from the Border Leaf.

- When EP is moved across SITEs or PODs, Host routes should be withdrawn from first SITE/POD and advertised in new POD/SITE.
• EPs learned on a specific BD, under any of the BD subnets are advertised from the L3out on the border leaf in the same POD.
• EPs are advertised out as Host Routes only in the local POD through the Border Leaf.
• Host routes are not advertised out from one POD to another POD.
• In the case of Remote Leaf, if EPs are locally learned in the Remote Leaf, they are then advertised only through a L3out deployed in Remote Leaf switches in same POD.
• EPs/Host routes in a Remote Leaf are not advertised out through Border Leaf switches in main POD or another POD.
• EPs/Host routes in the main POD are not advertised through L3out in Remote Leaf switches of same POD or another POD.
• The BD subnet must have the Advertise Externally option enabled.
• The BD must be associated to an L3out or the L3out must have explicit route-map configured matching BD subnets.
• There must be a contract between the EPG in the specified BD and the External EPG for the L3out.

Note
If there is no contract between the BD/EPG and the External EPG the BD subnet and host routes will not be installed on the border leaf.

• Advertise Host Route is supported for shared services. For example: epgl/BD1 deployed is in VRF-1 and L3out in another VRF-2. By providing shared contract between EPG and L3out host routes are pulled from one VRF-1 to another VRF-2.
• When Advertise Host Route is enabled on BD custom tag cannot be set on BD Subnet using route-map.
• When Advertise Host Route is enabled on a BD and the BD is associated with an L3Out, BD subnet is marked public. If there's a rogue EP present under the BD, that EP is advertised out on L3Out.

Guidelines for Routed Connectivity to Outside Networks

Use the following guidelines when creating and maintaining Layer 3 outside connections.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Caution or Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updates through CLI</td>
<td>For Layer 3 external networks created through the API or GUI and updated through the CLI, protocols need to be enabled globally on the external network through the API or GUI, and the node profile for all the participating nodes needs to be added through the API or GUI before doing any further updates through the CLI.</td>
</tr>
<tr>
<td>Loopbacks for Layer 3 networks on same node</td>
<td>When configuring two Layer 3 external networks on the same node, the loopbacks need to be configured separately for both Layer 3 networks.</td>
</tr>
<tr>
<td>Topic</td>
<td>Caution or Guideline</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Ingress-based policy enforcement</td>
<td>Starting with Cisco APIC release 1.2(1), ingress-based policy enforcement enables defining policy enforcement for Layer 3 Outside (L3Out) traffic for both egress and ingress directions. The default is ingress. During an upgrade to release 1.2(1) or higher, existing L3Out configurations are set to egress so that the behavior is consistent with the existing configuration. You do not need any special upgrade sequence. After the upgrade, you change the global property value to ingress. When it has been changed, the system reprograms the rules and prefix entries. Rules are removed from the egress leaf and installed on the ingress leaf, if not already present. If not already configured, an Act prefix entry is installed on the ingress leaf. Direct server return (DSR), and attribute EPGs require ingress based policy enforcement. vzAny and taboo contracts ignore ingress based policy enforcement. Transit rules are applied at ingress.</td>
</tr>
<tr>
<td>Bridge Domains with L3Outs</td>
<td>A bridge domain in a tenant can contain a public subnet that is advertised through an l3extOut provisioned in the common tenant.</td>
</tr>
<tr>
<td>Bridge domain route advertisement For OSPF and EIGRP</td>
<td>When both OSPF and EIGRP are enabled on the same VRF on a node and if the bridge domain subnets are advertised out of one of the L3Outs, it will also get advertised out of the protocol enabled on the other L3Out. For OSPF and EIGRP, the bridge domain route advertisement is per VRF and not per L3Out. The same behavior is expected when multiple OSPF L3Outs (for multiple areas) are enabled on the same VRF and node. In this case, the bridge domain route will be advertised out of all the areas, if it is enabled on one of them.</td>
</tr>
<tr>
<td>BGP Maximum Prefix Limit</td>
<td>Starting with Cisco APIC release 1.2(1x), tenant policies for BGP l3extOut connections can be configured with a maximum prefix limit, that enables monitoring and restricting the number of route prefixes received from a peer. Once the maximum prefix limit has been exceeded, a log entry is recorded, and further prefixes are rejected. The connection can be restarted if the count drops below the threshold in a fixed interval, or the connection is shut down. Only one option can be used at a time. The default setting is a limit of 20,000 prefixes, after which new prefixes are rejected. When the reject option is deployed, BGP accepts one more prefix beyond the configured limit, before the APIC raises a fault.</td>
</tr>
<tr>
<td>Topic</td>
<td>Caution or Guideline</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MTU</td>
<td><strong>Note</strong> Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link. On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface. For the appropriate MTU values for each platform, see the relevant configuration guides. We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as <code>ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1</code>.</td>
</tr>
<tr>
<td>Layer 4 to Layer 7</td>
<td>When you are using a multinode service graph, you must have the two EPGs in separate VRF instances. For these functions, the system must do a Layer 3 lookup, so the EPGs must be in separate VRFs. This limitation follows legacy service insertion, based on Layer 2 and Layer 3 lookups.</td>
</tr>
<tr>
<td>QoS for L3Outs</td>
<td>To configure QoS policies for an L3Out and enable the policies to be enforced on the BL switch where the L3Out is located, use the following guidelines:</td>
</tr>
<tr>
<td></td>
<td>• The VRF Policy Control Enforcement Direction must be set to <strong>Egress</strong>.</td>
</tr>
<tr>
<td></td>
<td>• The VRF Policy Control Enforcement Preference must be set to <strong>Enabled</strong>.</td>
</tr>
<tr>
<td></td>
<td>• When configuring the contract that controls communication between the EPGs using the L3Out, include the QoS class or Target DSCP in the contract or subject of the contract.</td>
</tr>
<tr>
<td>ICMP settings</td>
<td>ICMP redirect and ICMP unreachable are disabled by default in Cisco ACI to protect the switch CPU from generating these packets.</td>
</tr>
</tbody>
</table>
Configuring Layer 3 Outside for Tenant Networks

Configuring a Tenant Layer 3 Outside Network Connection Overview

This topic provides a typical example of how to configure a Layer 3 Outside for tenant networks when using Cisco APIC.

The examples in this chapter use the following topology:

*Figure 12: Layer 3 External Connections Topology*

In this example, the Cisco ACI fabric has 3 leaf switches and two spine switches, that are controlled by an APIC cluster. The nonborder leaf switches (101 and 102) are connected to a web server and a database server. The border leaf switch (103) has an L3Out on it providing connection to a router and thus to the Internet. The goal of this example is to enable the web server to communicate through the L3Out on the border leaf switch to an endpoint (EP) on the Internet.

In this example, the tenant that is associated with the L3Out is `t1`, with VRF `v1`, and L3Out external EPG, `extnw1`.

Before configuring an L3Out, configure the node, port, functional profile, AEP, and Layer 3 domain. You must also configure the spine switches 104 and 105 as BGP route reflectors.

Configuring the L3Out includes defining the following components:
1. Tenant and VRF
2. Node and interface on leaf 103
3. Primary routing protocol (used to exchange routes between border leaf switch and external routers; in this example, BGP)
4. Connectivity routing protocol (provides reachability information for the primary protocol; in this example, OSPF)
5. External EPG
6. Route map
7. Bridge domain
8. At least one application EPG on node 101
9. Filters and contracts
10. Associate the contracts with the EPGs

The following table lists the names that are used in the examples in this chapter:

<table>
<thead>
<tr>
<th>Property</th>
<th>Node 103 (Border Leaf)</th>
<th>Node 101 (Non-Border Leaf)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenant</td>
<td>t1</td>
<td>t1</td>
</tr>
<tr>
<td>VRF</td>
<td>v1</td>
<td>v1</td>
</tr>
<tr>
<td>Layer 3 Outside</td>
<td>l3out1</td>
<td>--</td>
</tr>
<tr>
<td>Bridge domain</td>
<td>--</td>
<td>bd1 with subnet 44.44.44.1/24</td>
</tr>
<tr>
<td>Node</td>
<td>Node 103, with profile nodep1 with router ID 11.11.11.103 and path through 12.12.12.3/24</td>
<td>Node 101</td>
</tr>
<tr>
<td>Interface</td>
<td>OSPF interface ifp1 at eth/1/3 with IP address 11.11.11.1/24</td>
<td>--</td>
</tr>
<tr>
<td>BGP details</td>
<td>Peer address 15.15.15.2/24 and ASN 100</td>
<td>--</td>
</tr>
<tr>
<td>OSPF details</td>
<td>OSPF area 0.0.0.0 and type Regular</td>
<td>--</td>
</tr>
<tr>
<td>EPG</td>
<td>External EPG extnw1 at 20.20.20.0/24</td>
<td>Application app1 with epgl, with bd1</td>
</tr>
<tr>
<td>Route Control Profile</td>
<td>rp1 with a route control context ctxp1</td>
<td>--</td>
</tr>
<tr>
<td>Route map</td>
<td>map1 with rule match-rule1 with a route destination 200.3.2.0/24</td>
<td>--</td>
</tr>
<tr>
<td>Filter</td>
<td>http-filter</td>
<td>http-filter</td>
</tr>
<tr>
<td>Contract</td>
<td>httpCtrct provided by extnw1</td>
<td>httpCtrct consumed by epgl</td>
</tr>
</tbody>
</table>
Configuring Layer 3 Outside for Tenant Networks Using the REST API

The external routed network that is configured in the example can also be extended to support both IPv4 and IPv6. Both IPv4 and IPv6 routes can be advertised to and learned from the external routed network. To configure an L3Out for a tenant network, send a post with XML such as the example.

This example is broken into steps for clarity. For a merged example, see REST API Example: L3Out, on page 32.

**Before you begin**

- Configure the node, port, functional profile, AEP, and Layer 3 domain.
- Create the external routed domain and associate it to the interface for the L3Out.
- Configure a BGP route reflector policy to propagate the routes within the fabric.

For an XML example of these prerequisites, see REST API Example: L3Out Prerequisites, on page 31.

**Procedure**

**Step 1** Configure the tenant, VRF, and bridge domain.

This example configures tenant t1 with VRF v1 and bridge domain bd1. The tenant, VRF, and BD are not yet deployed.

Example:

```
<fvTenant name="t1">
  <fvCtx name="v1"/>
  <fvBD name="bd1">
    <fvRsCtx tnFvCtxName="v1"/>
    <fvSubnet ip="44.44.44.1/24" scope="public"/>
    <fvRsBDToOut tnL3extOutName="l3out1"/>
  </fvBD>
</fvTenant>
```

**Step 2** Configure an application profile and application EPG.

This example configures application profile app1 (on node 101), EPG epgl, and associates the EPG with bd1 and the contract httpCtrct, as the consumer.

Example:

```
<fvAp name="app1">
  <fvAEPg name="epgl">
    <fvRsDomAtt instrImedcy="immediate" tDn="uni/phys-dom1"/>
    <fvRsBd tnFvBDName="bd1"/>
    <fvRsPathAtt encap="vlan-2011" instrImedcy="immediate" mode="regular" tDn="topology/pod-1/paths-101/pathep-[eth1/3]"/>
    <fvRsCons tnVzBrCPName="httpCtrct"/>
  </fvAEPg>
</fvAp>
```

**Step 3** Configure the node and interface.

This example configures VRF v1 on node 103 (the border leaf switch), with the node profile, nodep1, and router ID 11.11.11.103. It also configures interface ethl/3 as a routed interface (Layer 3 port), with IP address 12.12.12.1/24 and Layer 3 domain dom1.
Step 4  Configure the routing protocol.

This example configures BGP as the primary routing protocol, with a BGP peer with the IP address, 15.15.15.2 and ASN 100.

Example:

```xml
<l3extOut name="l3out1">
  <l3extRsEctx tnVctxName="v1"/>
  <l3extLNodeP name="nodep1">
    <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-103"/>
    <l3extLIfP name="ifp1"/>
  </l3extLNodeP>
  <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port" tDn="topology/pod-1/paths-103/pathep-[eth1/3]"/>
</l3extOut>
```

Step 5  Configure the connectivity routing protocol.

This example configures OSPF as the communication protocol, with regular area ID 0.0.0.0.

Example:

```xml
<l3extOut name="l3out1">
  <ospfExtP areaId="0.0.0.0" areaType="regular"/>
  <l3extLNodeP name="nodep1">
    <l3extLIfP name="ifp1">
      <ospfIfP/>
    </l3extLIfP>
  </l3extLNodeP>
</l3extOut>
```

Step 6  Configure the external EPG.

This example configures the network 20.20.20.0/24 as external network extnw1. It also associates extnw1 with the route control profile rp1 and the contract httpCtrct, as the provider.

Example:

```xml
<l3extOut name="l3out1">
  <l3extInstP name="extnw1">
    <l3extSubnet ip="20.20.20.0/24" scope="import-security"/>
    <fvRsProv tnVzBrCPName="httpCtrct"/>
  </l3extInstP>
</l3extOut>
```

Step 7  Optional. Configure a route map.

This example configures a route map for the BGP peer in the outbound direction. The route map is applied for routes that match a destination of 200.3.2.0/24. Also, on a successful match (if the route matches this range) the route AS PATH attribute is updated to 200 and 100.

Example:
Step 8

This example creates filters and contracts to enable the EPGs to communicate. The external EPG and the application EPG are already associated with the contract `httpCtrct` as provider and consumer respectively. The scope of the contract (where it is applied) can be within the application profile, the tenant, the VRF, or it can be used globally (throughout the fabric). In this example, the scope is the VRF (context).

**Example:**

```xml
<vzFilter name="http-filter">
  <vzEntry name="http-e" etherT="ip" prot="tcp"/>
</vzFilter>
<vzBrCP name="httpCtrct" scope="context">
  <vzSubj name="subj1">
    <vzRsSubjFiltAtt tnVzFilterName="http-filter"/>
  </vzSubj>
</vzBrCP>
```

Step 9

Configure Advertise Host Routes.

**Example:**

```
"<fvBD dn="uni/tn-t1/BD-b100" hostBasedRouting="yes"/>"
```

---

**REST API Example: L3Out Prerequisites**

This example configures the node, port, functional profile, AEP, and Layer 3 domain:

```xml
<infraInfra>
  <!-- Node profile -->
  <infraNodeP name="nodeP1">
    <infraLeafS name="leafS1" type="range">
      <infraNodeBlk name="NodeBlk1" from="101" to="103"/>
    </infraLeafS>
  </infraNodeP>
  <!-- Port profile -->
  <infraAccPortP tDn="uni/infra/accportprof-PortP1"/>
</infraInfra>
```

The following example configures the required BGP route reflectors:

```xml
<!-- Spine switches 104 and 105 are configured as route reflectors -->
<?xml version="1.0" encoding="UTF8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <bgpInstPol name="default">
    <bgpAsP asn="100"/>
    <bgpRRP>
      <bgpRRNodePEp id="104"/>
      <bgpRRNodePEp id="105"/>
    </bgpRRP>
  </bgpInstPol>
</polUni>
```

**REST API Example: L3Out**

The following example provides a merged version of the steps to configure an L3Out using the REST API.

```xml
<?xml version="1.0" encoding="UTF8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
  <fvTenant name="t1">
    <fvCtx name="v1"/>
    <fvBD name="bd1">
      <fvRsCtx tnFvCtxName="v1"/>
      <fvRsL3ExtP bdName="bd1"/>
      <fvRsL3ExtP bdName="bd1"/>
    </fvBD>
  </fvTenant>
</polUni>
```
Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Configuring a Layer 3 Outside for Tenant Networks Using the NX-OS Style CLI

These steps describe how to configure a Layer 3 outside network for tenant networks. This example shows how to deploy a node and L3 port for tenant VRF external L3 connectivity using the NX-OS CLI.

This example is broken into steps for clarity. For a merged example, see NX-OS Style CLI Example: L3Out, on page 37.

**Before you begin**

- Configure the node, port, functional profile, AEP, and Layer 3 domain.
- Configure a VLAN domain using the `vlan-domain` and `vlan vlan-range` commands.
- Configure a BGP route reflector policy to propagate the routed within the fabric.

For an example using the commands for these prerequisites, see NX-OS Style CLI Example: L3Out Prerequisites, on page 37.

**Procedure**

**Step 1** Configure the tenant and VRF.

This example configures tenant `t1` with VRF `v1`. They are not yet deployed.

**Example:**

```
apicl# configure
apicl(config)# tenant t1
apicl(config-tenant)# vrf context v1
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# exit
apicl(config)#
```

**Step 2** Configure the node and interface for the L3Out.

This example configures VRF `v1` on node 103 (the border leaf switch), which is named `nodep1`, with router ID `11.11.11.103`. It also configures interface `eth1/3` as a routed interface (Layer 3 port), with IP address `12.12.12.3/24` and Layer 3 domain `dom1`.

**Example:**

```
apicl(config)# leaf 103
apicl(config-leaf)# vrf context tenant t1 vrf v1
apicl(config-leaf-vrf)# router-id 11.11.11.103
apicl(config-leaf-vrf)# exit
apicl(config-leaf)# interface ethernet 1/3
apicl(config-leaf-if)# vlan-domain member dom1
apicl(config-leaf-if)# no switchport
apicl(config-leaf-if)# vrf member tenant t1 vrf v1
apicl(config-leaf-if)# ip address 12.12.12.3/24
apicl(config-leaf-if)# exit
apicl(config-leaf)#
```

**Step 3** Configure the routing protocol.

This example configures BGP as the primary routing protocol, with a BGP peer address, `15.15.15.2` and ASN 100.

**Example:**

```
```
apic1(config)# leaf 103
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

Step 4  Optional. Configure a connectivity routing protocol.

This example configures OSPF as the communication protocol, with regular area ID 0.0.0.0, with loopback address 30.30.30.0.

Example:

apic1(config)# leaf 103
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 30.30.30.0
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit

Step 5  Configure the external EPG on node 103.

In this example, the network 20.20.20.0/24 is configured as the external network extnw1.

Example:

apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 20.20.20.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit

Step 6  Optional. Configure Advertise Host Routing.

Example:

apic1# configure
apic1(config)# tenant <Name>
apic1(config-tenant)# bridge-domain <Name>
apic1(config-tenant-bd)# advertise-host-routes
apic1(config-tenant-bd)# end

Step 7  Optional. Configure a route map.

This example configures a route map rp1 for the BGP peer in the outbound direction. The route map is applied for routes that match a destination of 200.3.2.0/24. Also, on a successful match (if the route matches this range) the route AS PATH attribute is updated to 200 and 100.

Example:

apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 200.3.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
Step 8  Add a bridge domain.

Example:
```
apic1(config)# tenant t1
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ip address 44.44.44.1/24 scope public
apic1(config-tenant-interface)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match bridge-domain bd1 tenant t1
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
```

Step 9  Create an application EPG on node 101.

Example:
```
apic1(config)# tenant t1
apic1(config-tenant)# application app1
apic1(config-tenant-app)# epg epg1
apic1(config-tenant-app-epg)# bridge-domain member bd1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 2011 tenant t1 application app1 epg epg1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)#
```

Step 10  Create filters (access-lists) and contracts.

Example:
```
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Step 11  Configure contracts and associate them with EPGs.

Example:

```
aplic1(config-tenant-contract)# subject subj1
aplic1(config-tenant-contract-subj)# access-group http-filter both
aplic1(config-tenant-contract-subj)# exit
aplic1(config-tenant-contract)# exit
```

NX-OS Style CLI Example: L3Out Prerequisites

Before you can configure an L3Out, perform the following steps:

1. Configure a VLAN domain:

```
aplic1# configure
aplic1(config)# vlan-domain dom1
aplic1(config-vlan)# vlan 1024-2048
aplic1(config-vlan)# exit
```

2. Configure BGP route reflectors:

```
aplic1(config)# bgp-fabric
aplic1(config-bgp-fabric)# asn 100
aplic1(config-bgp-fabric)# route-reflector spine 104,105
```

NX-OS Style CLI Example: L3Out

The following example provides a merged version of the steps to configure an L3Out using the NX-OS style CLI. Configure the following prerequisites before configuring the L3Out.

```
aplic1# configure
aplic1(config)# tenant t1
aplic1(config-tenant)# vrf context v1
aplic1(config-tenant-vrf)# exit
aplic1(config-tenant)# exit
aplic1(config)# leaf 103
aplic1(config-leaf)# vrf context tenant t1 vrf v1
aplic1(config-leaf-vrf)# router-id 11.11.11.103
aplic1(config-leaf-vrf)# exit
aplic1(config-leaf)# interface ethernet 1/3
aplic1(config-leaf-if)# vlan-domain member dom1
aplic1(config-leaf-if)# no switchport
aplic1(config-leaf-if)# vrf member tenant t1 vrf v1
aplic1(config-leaf-if)# ip address 12.12.12.3/24
aplic1(config-leaf-if)# exit
aplic1(config-leaf)# router bgp 100
aplic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 30.30.30.0
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit
apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 20.20.20.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 200.3.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
apic1(config)# tenant t1
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ip address 44.44.44.1/24 scope public
apic1(config-tenant-interface)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map map1
apic1(config-leaf-vrf-route-map)# match bridge-domain bd1 tenant t1
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
apic1(config)# tenant t1
apic1(config-tenant)# application app1
apic1(config-tenant-app)# epg epg1
apic1(config-tenant-app-epg)# bridge-domain member bd1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# switchport trunk allowed vlan 2011 tenant t1 application app1 epg
epgl
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# subject subj1
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-13ext-epg)# vrf member v1
apic1(config-tenant-13ext-epg)# contract provider httpCtrct
apic1(config-tenant-13ext-epg)# exit
apic1(config-tenant)# application app1
apic1(config-tenant-app)# epg epgl
apic1(config-tenant-app-epg)# contract consumer httpCtrct
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)#

Configuring a Layer 3 Outside for Tenant Networks Using the GUI

Perform the following steps to configure a Layer 3 outside (L3Out) connection for the fabric.

Before you begin

- Configure the node, port, functional profile, AEP, and Layer 3 domain.
- Create the external routed domain and associate it to the interface for the L3Out.
- Configure a BGP Route Reflector policy to propagate the routes within the fabric.

Procedure

Step 1 To create the tenant and VRF, on the menu bar, choose Tenants > Add Tenant and in the Create Tenant dialog box, perform the following tasks:
a) In the Name field, enter the tenant name.
b) In the VRF Name field, enter the VRF name.
c) Click Submit.

Step 2 To create a bridge domain, in the Navigation pane, expand Tenant and Networking and perform the following steps:
a) Right-click Bridge Domains and choose Create Bridge Domain.
b) In the Name field, enter a name for the bridge domain (BD).
c) (Optional) Click the box for Advertise Host Routes to enable advertisement to all deployed border leaves.
d) In the VRF field, from the drop-down list, choose the VRF you created (v1 in this example).
e) Click Next.
f) Click the + icon on Subnets.
g) In the Gateway IP field, enter the subnet for the BD.

h) In the Scope field, choose Advertised Externally.

Add the L3 Out for Route Profile later, after you create it.

**Note** If Advertise Host Routes is enabled, the route-map will also match all host routes.

i) Click OK.

j) Click Next and click Finish.

**Step 3**
To create an application EPG, perform the following steps:

a) Right-click Application Profiles and choose Create Application Profile.

b) Enter a name for the application.

c) Click the + icon for EPGs.

d) Enter a name for the EPG.

e) From the BD drop-down list, choose the bridge domain you previously created.

f) Click Update.

g) Click Submit.

**Step 4**
To start creating the L3Out, on the Navigation pane, expand Tenant and Networking and perform the following steps:

a) Right-click External Routed Networks and choose Create Routed Outside.

b) In the Name field, enter a name for the L3Out.

c) From the VRF drop-down list, choose the VRF.

d) From the External Routed Domain drop-down list, choose the external routed domain that you previously created.

e) In the area with the routing protocol check boxes, check the desired protocols (BGP, OSPF, or EIGRP).

For the example in this chapter, choose BGP and OSPF.

Depending on the protocols you choose, enter the properties that must be set.

f) Enter the OSPF details, if you enabled OSPF.

For the example in this chapter, use the OSPF area 0 and type Regular area.

g) Click + to expand Nodes and Interfaces Protocol Profiles.

h) In the Name field, enter a name.

i) Click + to expand Nodes.

j) From the Node ID field drop-down menu, choose the node for the L3Out.

For the topology in these examples, use node 103.

k) In the Router ID field, enter the router ID (IPv4 or IPv6 address for the router that is connected to the L3Out).

l) (Optional) You can configure another IP address for a loopback address. Uncheck Use Router ID as Loopback Address, expand Loopback Addresses, enter an IP address, and click Update.

m) In the Select Node dialog box, click OK.

**Step 5**
If you enabled BGP, click the + icon to expand BGP Peer Connectivity Profiles and perform the following steps:

a) In the Peer Address field, enter the BGP peer address.

b) In the Local-AS Number field, enter the BGP AS number.
For the example in this chapter, use the BGP peer address 15.15.15.2 and ASN number 100.

c) Click OK.

**Step 6**
Click + to expand Interface Profiles (OSPF Interface Profiles if you enabled OSPF), and perform the following actions:

a) In the Name field, enter a name for the interface profile.
b) Click Next.
c) In the Protocol Profiles dialog box, in the OSPF Policy field, choose an OSPF policy.
d) Click Next.
e) Click the + icon to expand Routed Interfaces.
f) In the Select Routed Interface dialog box, from the Node drop-down list, choose the node.
g) From the Path drop-down list, choose the interface path.
h) In the IPv4 Primary/IPv6 Preferred Address field, enter the IP address and network mask for the interface.

*Note* To configure IPv6, you must enter the link-local address in the Link-local Address field.

i) Click OK in the Select Routed Interface dialog box.
j) Click OK in the Create Interface Profile dialog box.

**Step 7**
In the Create Node Profile dialog box, click OK.

**Step 8**
In the Create Routed Outside dialog box, click Next.

**Step 9**
In the External EPG Networks tab, click Create Route Profiles.

**Step 10**
Click the + icon to expand Route Profiles and perform the following actions:

a) In the Name field, enter the route map name.
b) Choose the Type.

c) Click the + icon to expand Contexts and create a route context for the route map.
d) Enter the order and name of the profile context.
e) Choose Deny or Permit for the action to be performed in this context.
f) (Optional) In the Set Rule field, choose Create Set Rules for a Route Map.

Enter the name for the set rules, click the objects to be used in the rules, and click Finish.

g) In the Associated Matched Rules field, click + to create a match rule for the route map.
h) Enter the name for the match rules and enter the Match Regex Community Terms, Match Community Terms, or Match Prefix to match in the rule.
i) Click the rule name and click Update.
j) In the Create Match Rule dialog box, click Submit, and then click Update.
k) In the Create Route Control Context dialog box, click OK.
l) In the Create Route Map dialog box, click OK.

**Step 11**
Click the + icon to expand External EPG Networks.

**Step 12**
In the Name field, enter a name for the external network.

**Step 13**
Click the + icon to expand Subnet.

**Step 14**
In the Create Subnet dialog box, perform the following actions:

a) In the IP address field, enter the IP address and network mask for the external network.
b) In the **Scope** field, check the appropriate check boxes to control the import and export of prefixes for the L3Out.

   **Note**   For more information about the scope options, see the online help for this **Create Subnet** panel.

c) (Optional) In the **Route Summarization Policy** field, from the drop-down list, choose an existing route summarization policy or create a new one as desired. Also click the check box for **Export Route Control Subnet**.

   The type of route summarization policy depends on the routing protocols that are enabled for the L3Out.

d) Click the + icon to expand **Route Control Profile**.

e) In the **Name** field, choose the route control profile that you previously created from the drop-down list.

f) In the **Direction** field, choose **Route Export Policy**.

g) Click **Update**.

h) In the **Create Subnet** dialog box, click **OK**.

i) (Optional) Repeat to add more subnets.

j) In the **Create External Network** dialog box, click **OK**.

**Step 15**  In the **Create Routed Outside** dialog box, click **Finish**.

**Step 16**  In the **Navigation** pane, under **Tenant_name > Networking** expand **Bridge Domains**.

   **Note**   If the L3Out is static, you are not required to choose any BD settings.

**Step 17**  Choose the BD you created.

   a) In the **Work** pane, click **Policy and L3 Configurations**.
   
   b) Click the + icon to expand the **Associated L3 Outs** field, choose the previously configured L3Out, and click **Update**.

   c) In the **L3Out for Route Profile** field, choose the L3Out again.

   d) Click **Next** and **Finish**.

**Step 18**  In the **Navigation** pane, under **External Routed Networks**, expand the previously created L3Out and right-click **Route Maps/Profiles**.

   **Note**   To set attributes for BGP, OSPF, or EIGRP for received routes, create a default-import route control profile, with the appropriate set actions and no match actions.

**Step 19**  Choose **Create Route Map/Profile**, and in the **Create Route Map/Profile** dialog box, perform the following actions:

   a) From the drop-down list on the **Name** field, choose **default-import**.
   
   b) In the **Type** field, you must click **Match Routing Policy Only**. Click **Submit**.

**Step 20**  (Optional) To enable extra communities to use BGP, using the following steps:

   a) Right-click **Set Rules for Route Maps**, and click **Create Set Rules for a Route Map**.

   b) In the **Create Set Rules for a Route Map** dialog box, click the **Add Communities** field, and follow the steps to assign multiple BGP communities per route prefix.

**Step 21**  To enable communications between the EPGs consuming the L3Out, create at least one filter and contract, using the following steps:

   a) In the **Navigation** pane, under the tenant consuming the L3Out, expand **Contracts**.
   
   b) Right-click **Filters** and choose **Create Filter**.

   c) In the **Name** field, enter a filter name.
A filter is essentially an Access Control List (ACL).

d) Click the + icon to expand Entries, and add a filter entry.
e) Add the Entry details.

For example, for a simple web filter, set criteria such as the following:

- **EtherType**—IP
- **IP Protocol**—tcp
- **Destination Port Range From**—Unspecified
- **Destination Port Range To** to https

f) Click Update.
g) In the Create Filter dialog box, click Submit.

**Step 22**

To add a contract, use the following steps:

a) Under Contracts, right-click Standard and choose Create Contract.
b) Enter the name of the contract.
c) Click the + icon to expand Subjects to add a subject to the contract.
d) Enter a name for the subject.
e) Click the + icon to expand Filters and choose the filter that you previously created, from the drop-down list.
f) Click Update.
g) In the Create Contract Subject dialog box, click OK.
h) In the Create Contract dialog box, click Submit.

**Step 23**

Associate the EPGs for the L3Out with the contract, with the following steps:

In this example, the L3 external EPG (extnw1) is the provider and the application EPG (epg1) is the consumer.

a) To associate the contract to the L3 external EPG, as the provider, under the tenant, click Networking, expand External Routed Networks, and expand the L3Out.
b) Expand Networks, click the L3 external EPG, and click Contracts.
c) Click the the + icon to expand Provided Contracts.
d) In the Name field, choose the contract that you previously created from the list.
e) Click Update.
f) To associate the contract to an application EPG, as a consumer, under the tenant, navigate to Application Profiles > app-prof-name > Application EPGs > and expand the app-epg-name.
g) Right-click Contracts, and choose Add Consumed Contract.
h) On the Contract field, choose the contract that you previously created.
i) Click Submit.
Layer 3 Routed and Sub-Interface Port Channels

• About Layer 3 Port Channels, on page 45
• Configuring Port Channels Using the GUI, on page 46
• Configuring a Layer 3 Routed Port-Channel Using the GUI, on page 47
• Configuring a Layer 3 Sub-Interface Port-Channel Using the GUI, on page 49
• Configuring a Layer 3 Routed Port-Channel Using the NX-OS CLI, on page 51
• Configuring a Layer 3 Sub-Interface Port-Channel Using the NX-OS CLI, on page 53
• Adding Ports to the Layer 3 Port-Channel Using the NX-OS CLI, on page 56
• Configuring Port Channels Using the REST API, on page 57
• Configuring a Layer 3 Routed Port Channel Using the REST API, on page 58
• Configuring a Layer 3 Sub-Interface Port Channel Using the REST API, on page 59

About Layer 3 Port Channels

Previously, Cisco APIC supported only Layer 2 port channels. Starting with release 3.2(1), Cisco APIC now supports Layer 3 port channels.

Figure 13: Switch Port Channel Configuration

Layer 3 routed and sub-interface port channels on border leaf switches are supported only on new generation switches, which are switch models with "EX", "FX" or "FX2" at the end of the switch name.
Configuring Port Channels Using the GUI

You must first configure port channels using these procedures before you can configure a Layer 3 route to the port channels using the GUI in subsequent procedures.

The procedure below uses a Quick Start wizard.

**Before you begin**

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
- An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the ACI fabric and available.

**Procedure**

**Step 1** On the APIC menu bar, navigate to **Fabric > External Access Policies > Quick Start**, and click **Configure Interface, PC, and VPC**.

**Step 2** In the **Configure Interface, PC, and VPC** work area, click the large + to select switches to configure.

**Step 3** In the **Switches** section, select a switch ID from the drop-down list of available switch IDs.

**Step 4** Click the large + to configure switch interfaces.

**Step 5** In the **Interface Type** field, specify **PC** as the interface type to use.

**Step 6** In the **Interfaces** field, specify the interface IDs to use.

**Step 7** (Optional) In the **Interface Selector Name** field, enter a unique interface selector name, if desired.

**Step 8** In the Interface Policy Group area, specify the interface policies to use. For example, click the **Port Channel Policy** drop-down arrow to choose an existing port channel policy or to create a new port channel policy.
• Choosing to create a port channel policy displays the Create Port Channel Policy dialog box where you can specify the policy details and enable features such as symmetric hashing. Also note that choosing the Symmetric hashing option displays the Load Balance Hashing field, which enables you to configure hash tuple. However, only one customized hashing option can be applied on the same leaf switch.

• Symmetric hashing is not supported on the following switches:
  • Cisco Nexus 93128TX
  • Cisco Nexus 9372PX
  • Cisco Nexus 9372PX-E
  • Cisco Nexus 9372TX
  • Cisco Nexus 9372TX-E
  • Cisco Nexus 9396PX
  • Cisco Nexus 9396TX

Step 9  In the Attached Device Type field, select the External Routed Devices option.
Step 10 In the Domain field, create a domain or choose one to assign to the interface.
Step 11 If you choose to create a domain, in the VLAN field, select from existing VLAN pools or create a new VLAN range to assign to the interface.
Step 12 Click Save to update the policy details, then click Submit to submit the switch profile to the APIC. The APIC creates the switch profile, along with the interface, selector, and attached device type policies.

What to do next
Configure a Layer 3 routed port channel or a Layer 3 sub-interface port channel using the GUI.

Configuring a Layer 3 Routed Port-Channel Using the GUI

This procedure configures a Layer 3 route to the port channels that you created previously.

Before you begin

• The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.

• An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.

• The target leaf switches are registered in the ACI fabric and available.

• Port channels are configured using the procedures in "Configuring Port Channels Using the GUI".
## Procedure

**Step 1**  
On the APIC menu bar, navigate to Tenants > Tenant > Networking > External Routed Networks > L3Out > Logical Node Profiles > node > Logical Interface Profiles.

**Step 2**  
Select the interface that you want to configure. The Logical Interface Profile page for that interface opens.

**Step 3**  
Click on Routed Interfaces. The Properties page opens.

**Step 4**  
Click on the Create (+) button to configure the Layer 3 routed port-channel. The Select Routed Interface page opens.

**Step 5**  
In the Path Type field, select Direct Port Channel.

**Step 6**  
In the Path field, select the port channel that you created previously from the drop-down list. This is the path to the port channel end points for the interface profile.

**Step 7**  
In the Description field, enter a description of the routed interface.

**Step 8**  
In the IPv4 Primary / IPv6 Preferred Address field, enter the primary IP addresses of the path attached to the Layer 3 outside profile.

**Step 9**  
In the IPv6 DAD field, select disabled or enabled.  
See "Configuring IPv6 Neighbor Discovery Duplicate Address Detection" for more information for this field.

**Step 10**  
In the IPv4 Secondary / IPv6 Additional Addresses field, enter the secondary IP addresses of the path attached to the Layer 3 outside profile.  
See "Configuring IPv6 Neighbor Discovery Duplicate Address Detection" for more information for the IPv6 DAD field in the Create Secondary IP Address screen.

**Step 11**  
Check the ND RA Prefix box if you wish to enable a Neighbor Discovery Router Advertisement prefix for the interface. The ND RA Prefix Policy option appears.  
When this is enabled, the routed interface is available for auto configuration and the prefix is sent to the host for auto-configuration.  
While ND RA Interface policies are deployed under BDs and/or Layer 3 Outs, ND prefix policies are deployed for individual subnets. The ND prefix policy is on a subnet level.  
The ND RA Prefix applies only to IPv6 addresses.

**Step 12**  
If you checked the ND RA Prefix box, select the ND RA Prefix policy that you want to use. You can select the default policy or you can choose to create your own ND RA prefix policy. If you choose to create your own policy, the Create ND RA Prefix Policy screen appears:  
a) In the Name field, enter the Router Advertisement (RA) name for the prefix policy.  
b) In the Description field, enter a description of the prefix policy.  
c) In the Controller State field, check the desired check boxes for the controller administrative state. More than one can be specified. The default is Auto Configuration and On link.  
d) In the Valid Prefix Lifetime field, choose the desired value for the length of time that you want the prefix to be valid. The range is from 0 to 4294967295 milliseconds. The default is 2592000.  
e) In the Preferred Prefix Lifetime field, choose the desired value for the preferred lifetime of the prefix. The range is from 0 to 4294967295 milliseconds. The default is 604800.  
f) Click Submit.

**Step 13**  
In the MAC Address field, enter the MAC address of the path attached to the Layer 3 outside profile.

**Step 14**  
In the MTU (bytes) field, set the maximum transmit unit of the external network. The range is 576 to 9216. To inherit the value, enter inherit in the field.
**Step 15**  
In the **Target DSCP** field, select the target differentiated services code point (DSCP) of the path attached to the Layer 3 outside profile from the drop-down list.

**Step 16**  
In the **Link-local Address** field, enter an IPv6 link-local address. This is the override of the system-generated IPv6 link-local address.

**Step 17**  
Click **Submit**.

**Step 18**  
Determine if you want to configure Layer 3 Multicast for this port channel.

To configure Layer 3 Multicast for this port channel:

a) On the APIC menu bar, navigate to the Layer 3 Out that you selected for this port channel (*Tenants* > *Tenant* > *Networking* > *External Routed Networks* > *L3Out*).

b) Click on the Policy tab to access the Properties screen for the Layer 3 Out.

c) In the Properties screen for the Layer 3 Out, scroll down to the PIM field, then click the check box next to that field to enable PIM.

This enables PIM on all interfaces under the Layer 3 Out, including this port channel.

d) Configure PIM on the external router.

You have to have a PIM session from the external router to the port channel. Refer to the documentation that you received with the external router for instructions on configuring PIM on your external router.

e) Map the port channel L3 Out to a VRF that has Multicast enabled.

See [IP Multicast, on page 203](#) for those instructions. Note the following:

- You will select a specific VRF that has Multicast enabled as part of this port channel L3 Out to VRF mapping process. In the Multicast screen for that VRF, if you do not see the L3 Out for this port channel when you try to select an L3 Out in the Interfaces area, go back to the L3 Out for this port channel, go to the Policy tab, select the appropriate VRF, then click Submit and Submit Changes. The L3 Out for this port channel should now be available in the Multicast screen for that VRF.

- You have to configure a Rendezvous Point (RP) for Multicast, an IP address that is external to the fabric. You can specify static RP, auto RP, fabric RP, or bootstrap router for the RP. For example, if you choose static RP, the IP address would be present on the external router, and APIC will learn this IP address through the L3 Out. See [IP Multicast, on page 203](#) for more information.

### Configuring a Layer 3 Sub-Interface Port-Channel Using the GUI

This procedure configures a Layer 3 sub-interface route to the port channels that you created previously.

**Before you begin**

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
- An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the ACI fabric and available.
- Port channels are configured using the procedures in "Configuring Port Channels Using the GUI".
Procedure

Step 1. On the APIC menubar, navigate to Tenants > Tenant > Networking > External Routed Networks > L3Out > Logical Node Profiles > node > Logical Interface Profiles.

Step 2. Select the interface that you want to configure. The Logical Interface Profile page for that interface opens.

Step 3. Click on Routed Sub-interfaces. The Properties page opens.

Step 4. Click on the Create (+) button to configure the Layer 3 routed sub-interface port-channel. The Select Routed Sub-Interface page opens.

Step 5. In the Path Type field, select Direct Port Channel.

Step 6. In the Path field, select the port channel that you created previously from the drop-down list. This is the path to the port channel end points for the interface profile.

Step 7. In the Description field, enter a description of the routed interface.

Step 8. In the Encap field, select VLAN from the drop-down menu. This is the encapsulation of the path attached to the Layer 3 outside profile. Enter an integer value for this entry.

Step 9. In the IPv4 Primary / IPv6 Preferred Address field, enter the primary IP addresses of the path attached to the Layer 3 outside profile.

Step 10. In the IPv6 DAD field, select disabled or enabled.

Step 11. In the IPv4 Secondary / IPv6 Additional Addresses field, enter the secondary IP addresses of the path attached to the Layer 3 outside profile.

Step 12. Check the ND RA Prefix box if you wish to enable a Neighbor Discovery Router Advertisement prefix for the interface. The ND RA Prefix Policy option appears.

Step 13. If you checked the ND RA Prefix box, select the ND RA Prefix policy that you want to use. You can select the default policy or you can choose to create your own ND RA prefix policy. If you choose to create your own policy, the Create ND RA Prefix Policy screen appears:

   a) In the Name field, enter the Router Advertisement (RA) name for the prefix policy.
   b) In the Description field, enter a description of the prefix policy.
   c) In the Controller State field, check the desired check boxes for the controller administrative state. More than one can be specified. The default is Auto Configuration and On link.
   d) In the Valid Prefix Lifetime field, choose the desired value for the length of time that you want the prefix to be valid. The range is from 0 to 4294967295 milliseconds. The default is 2592000.
   e) In the Preferred Prefix Lifetime field, choose the desired value for the preferred lifetime of the prefix. The range is from 0 to 4294967295 milliseconds. The default is 604800.
   f) Click Submit.

Step 14. In the MAC Address field, enter the MAC address of the path attached to the Layer 3 outside profile.
Step 15

In the MTU (bytes) field, set the maximum transmit unit of the external network. The range is 576 to 9216. To inherit the value, enter inherit in the field.

Step 16

In the Link-local Address field, enter an IPv6 link-local address. This is the override of the system-generated IPv6 link-local address.

Verification: Use the CLI show int command on the leaf switches where the external switch is attached to verify that the vpc is configured accordingly.

Step 17

Click Submit.

---

Configuring a Layer 3 Routed Port-Channel Using the NX-OS CLI

This procedure configures a Layer 3 routed port channel.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>configure</td>
</tr>
<tr>
<td>Example:</td>
<td>apic1# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>leaf node-id</td>
</tr>
<tr>
<td>Example:</td>
<td>apic1(config)# leaf 101</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>interface port-channel channel-name</td>
</tr>
<tr>
<td>Example:</td>
<td>apic1(config-leaf)# interface port-channel pol</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>no switchport</td>
</tr>
<tr>
<td>Example:</td>
<td>apic1(config-leaf-if)# no switchport</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>vrf member vrf-name tenant tenant-name</td>
</tr>
<tr>
<td>Example:</td>
<td>apic1(config-leaf-if)# vrf member v1 tenant tl</td>
</tr>
</tbody>
</table>

- vrf-name is the VRF name. The name can be any case-sensitive, alphanumeric string up to 32 characters.
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 6</strong></td>
<td><strong>tenant-name</strong> is the tenant name. The name can be any case-sensitive, alphanumeric string up to 32 characters. <strong>Associates the port channel template with the previously configured VLAN domain.</strong></td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Sets the IP address and subnet mask for the specified interface.</td>
</tr>
</tbody>
</table>
| **Step 8** | Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface, where: 
- **sub-bits** is the subprefix bits and host bits of the address to be concatenated with the prefixes provided by the general prefix specified with the prefix-name argument. The sub-bits argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. 
- **prefix-length** is the length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value. |
| **Step 9** | Configures an IPv6 link-local address for an interface. |
| **Step 10** | Manually sets the interface MAC address. |
| **Step 11** | Sets the MTU for this class of service. |
Example
This example shows how to configure a basic Layer 3 port channel.

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# ipv6 link-local fe80::1
apic1(config-leaf-if)# mac-address 00:44:55:66:55::01
apic1(config-leaf-if)# mtu 1500
```

Configuring a Layer 3 Sub-Interface Port-Channel Using the NX-OS CLI

This procedure configures a Layer 3 sub-interface port channel.

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apic1# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>leaf node-id</td>
<td>Specifies the leaf switch or leaf switches to be configured. The node-id can be a single node ID or a range of IDs, in the form node-id1-node-id2, to which the configuration will be applied.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apic1(config)# leaf 101</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td>vrf member vrf-name tenant tenant-name</td>
<td>Associates this port channel to this virtual routing and forwarding (VRF) instance and L3 outside policy, where:</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>apic1(config-leaf-if)# vrf member v1 tenant t1</td>
<td></td>
</tr>
</tbody>
</table>

- vrf-name is the VRF name. The name can be any case-sensitive, alphanumeric string up to 32 characters.
- tenant-name is the tenant name. The name can be any case-sensitive, alphanumeric string up to 32 characters.
### Configuring a Layer 3 Sub-Interface Port-Channel Using the NX-OS CLI

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| 4     | **vlan-domain member** *vlan-domain-name*  
**Example:**  
apic1(config-leaf-if)# vlan-domain member dom1 | Associates the port channel template with the previously configured VLAN domain. |
| 5     | **ip address** *ip-address* / *subnet-mask*  
**Example:**  
apic1(config-leaf-if)# ip address 10.1.1.1/24 | Sets the IP address and subnet mask for the specified interface. |
| 6     | **ipv6 address** *sub-bits* / *prefix-length* preferred  
**Example:**  
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred | Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface, where:  
- *sub-bits* is the subprefix bits and host bits of the address to be concatenated with the prefixes provided by the general prefix specified with the prefix-name argument. The sub-bits argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.  
- *prefix-length* is the length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value. |
| 7     | **ipv6 link-local** *ipv6-link-local-address*  
**Example:**  
apic1(config-leaf-if)# ipv6 link-local fe80::1 | Configures an IPv6 link-local address for an interface. |
| 8     | **mac-address** *mac-address*  
**Example:**  
apic1(config-leaf-if)# mac-address 00:44:55:66:55:01 | Manually sets the interface MAC address. |
| 9     | **mtu** *mtu-value*  
**Example:**  
apic1(config-leaf-if)# mtu 1500 | Sets the MTU for this class of service. |
| 10    | **exit**  
**Example:**  
apic1(config-leaf-if)# exit | Returns to configure mode. |
### Layer 3 Routed and Sub-Interface Port Channels

#### Configuring a Layer 3 Sub-Interface Port-Channel Using the NX-OS CLI

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 11</strong></td>
<td><strong>interface port-channel channel-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;apic1(config-leaf)# interface port-channel po1</td>
</tr>
<tr>
<td><strong>Step 12</strong></td>
<td><strong>vlan-domain member vlan-domain-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;apic1(config-leaf-if)# vlan-domain member dom1</td>
</tr>
<tr>
<td><strong>Step 13</strong></td>
<td><strong>exit</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;apic1(config-leaf-if)# exit</td>
</tr>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>interface port-channel channel-name.number</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;apic1(config-leaf)# interface port-channel po1.2001</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td><strong>vrf member vrf-name tenant tenant-name</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;apic1(config-leaf-if)# vrf member v1 tenant t1</td>
</tr>
<tr>
<td><strong>Step 16</strong></td>
<td><strong>exit</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;apic1(config-leaf-if)# exit</td>
</tr>
</tbody>
</table>

#### Example

This example shows how to configure a basic Layer 3 sub-interface port-channel.

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 2001
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member v1 tenant t1
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# ip address 10.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# ipv6 link-local fe80::1
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Adding Ports to the Layer 3 Port-Channel Using the NX-OS CLI

This procedure adds ports to a Layer 3 port channel that you configured previously.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Specifies the leaf switch or leaf switches to be configured. The node-id can be a single node ID or a range of IDs, in the form node-id1-node-id2, to which the configuration will be applied.</td>
</tr>
<tr>
<td>leaf node-id</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1(config)# leaf 101</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Enters interface configuration mode for the interface you want to configure.</td>
</tr>
<tr>
<td>interface Ethernet slot/port</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# interface Ethernet 1/1-2</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Configures the port in a channel group.</td>
</tr>
<tr>
<td>channel-group channel-name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-if)# channel-group p01</td>
<td></td>
</tr>
</tbody>
</table>

**Example**

This example shows how to add ports to a Layer 3 port-channel.

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface Ethernet 1/1-2
apic1(config-leaf-if)# channel-group p01
```
Configuring Port Channels Using the REST API

Before you begin

The procedures in this section are meant specifically for configuring port channels as a prerequisite to the procedures for configuring a Layer 3 routed or sub-interface port channel. For general instructions on configuring leaf switch port channels, refer to the Cisco APIC Basic Configuration Guide or Cisco APIC Layer 2 Networking Configuration Guide.

• The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
• An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
• The target leaf switches are registered in the ACI fabric and available.

In the following REST API example, long single lines of text are broken up with the \ character to improve readability.

Procedure

To configure a port channel using the REST API, send a post with XML such as the following:

Example:
<polUni>
  <infraInfra dn="uni/infra">
    <infraNodeP name="test1">
      <infraLeafS name="leafs" type="range">
        <infraNodeBlk name="nbik" from="101" to="101"/>
      </infraLeafS>
      <infraRsAccPortP tDn="uni/infra/accportprof-test1"/>
    </infraNodeP>
    <infraAccPortP name="test1">
      <infraHPortS name="pselc" type="range">
        <infraPortBlk name="blk1" fromCard="1" toCard="1" fromPort="18" toPort="19"/>
      </infraHPortS>
      <infraRsAccBaseGrp tDn="uni/infra/funcprof/accbundle-po17_PolGrp"/>
    </infraAccPortP>
  </infraInfra>
</polUni>
Configuring a Layer 3 Routed Port Channel Using the REST API

Before you begin

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
- An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the ACI fabric and available.
- Port channels are configured using the procedures in "Configuring Port Channels Using the REST API".

In the following REST API example, long single lines of text are broken up with the \ character to improve readability.

Procedure

To configure a Layer 3 route to the port channels that you created previously using the REST API, send a post with XML such as the following:

Example:

```xml
<polUni>
  <fvTenant name=pep9>
    <l3extOut descr="" dn="uni/tn-pep9/out-routAccounting" enforceRtctrl="export" name="routAccounting" nameAlias="" ownerKey="" ownerTag="" targetDscp="unspecified">
      <l3extRsL3DomAtt tDn="uni/l3dom-Dom1"/>
      <l3extRsCtx tNfVCtxName="ctx9"/>
      <l3extLNodeP configIssues="" descr="" name=node101" nameAlias="" ownerKey="" ownerTag="" tag="yellow-green" targetDscp="unspecified">
        <l3extRsNodeL3OutAtt rtrId="10.1.0.101" rtrIdLoopBack="yes" tDn="topology/pod-1/node-101">
          <l3extInfraNodeP descr="" fabricExtCtrlPeering="no" fabricExtIntersiteCtrlPeering="no" name="" nameAlias="" spineRole=""/>
        </l3extRsNodeL3OutAtt>
        <l3extLIfP descr="" name=lifp17" nameAlias="" ownerKey="" ownerTag="" tag="yellow-green">
          <ospfIfP authKeyId="1" authType="none" descr="" name="" nameAlias="">
            <ospfRsIfPol tnOspfIfPolName=""/>
          </ospfIfP>
        </l3extLIfP>
      </l3extLNodeP>
    </l3extOut>
  </fvTenant>
</polUni>
```
Configuring a Layer 3 Sub-Interface Port Channel Using the REST API

Before you begin

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
- An APIC fabric administrator account is available that will enable creating the necessary fabric infrastructure configurations.
- The target leaf switches are registered in the ACI fabric and available.
- Port channels are configured using the procedures in "Configuring Port Channels Using the REST API".

Note

In the following REST API example, long single lines of text are broken up with the \ character to improve readability.

Procedure

To configure a Layer 3 sub-interface route to the port channels that you created previously using the REST API, send a post with XML such as the following:
Example:

```xml
<polUni>
  <fvTenant name=pep9>
    <l3extOut descr="" dn="uni/tn-pep9/out-routAccounting" enforceRtctrl="export" \\nname="routAccounting" nameAlias="" ownerKey="" ownerTag="" targetDscp="unspecified">
      <l3extRsL3DomAtt tDn="uni/l3dom-Dom1"/>
      <l3extRsCtx dn="ctx9"/>
      <l3extLNodeP configIssues="" descr="" name=nodes91 ownerKey="" ownerTag="" \\
tag="yellow-green" targetDscp="unspecified">
        <l3extRsNodeL3OutAtt rtrId="10.1.0.101" rtrIdLoopBack="yes" \\
tDn="topology/pod-1/node-101"/>
        <l3extInfraNodeP descr="" fabricExtCtrlPeering="no" \\
fabricExtIntersiteCtrlPeering="no" name="" spineRole=""/>
      </l3extRsNodeL3OutAtt>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>
<polUni>
```
L3Outs QoS

L3Out QoS can be configured using Contracts applied at the external EPG level. Starting with Release 4.0(1), L3Out QoS can also be configured directly on the L3Out interfaces.

Note
If you are running Cisco APIC Release 4.0(1) or later, we recommend using the custom QoS policies applied directly to the L3Out to configure QoS for L3Outs.

Packets are classified using the ingress DSCP or CoS value so it is possible to use custom QoS policies to classify the incoming traffic into Cisco ACI QoS queues. A custom QoS policy contains a table mapping the DSCP/CoS values to the user queue and to the new DSCP/CoS value (in case of marking). If there is no mapping for a specific DSCP/CoS value, the user queue is selected by the QoS priority setting of the ingress L3Out interface if configured.

L3Outs QoS Guidelines and Limitations

The following guidelines apply to configuring QoS for L3Outs:

- Custom QoS policy is not supported for Layer 3 multicast traffic sourced from outside the ACI fabric (received from L3Out).
• When configuring the QoS policy via contracts to be enforced on the border leaf where the L3Out is located, the VRF instance must be in egress mode (Policy Control Enforcement Direction must be "Egress").

Starting with Release 4.0(1), custom QoS setting can be configured directly on an L3Out and applied for the traffic coming from the border leaf, as such, the VRF does not need to be in egress mode.

• To enable the QoS policy to be enforced, the VRF Policy Control Enforcement Preference must be "Enforced."

• When configuring the Contract that controls communication between the L3Out and other EPGs, include the QoS class or target DSCP in the contract or subject.

    **Note** Only configure a QoS class or target DSCP in the contract, not in the external EPG (l3extInstP).

• When creating a contract subject, you must choose a QoS priority level. You cannot choose Unspecified.

    **Note** With the exception of Custom QoS Policies as a custom QoS Policy will set the DSCP/CoS value even if the QoS Class is set to Unspecified. When QoS level is unspecified, it by default takes as Level 3 default queue. No unspecified is supported and valid.

• Starting with Release 4.0(1), QoS supports new levels 4, 5, and 6 configured under Global policies, EPG, L3out, custom QoS, and Contracts. The following limitations apply:
  • Number of classes that can be configured with Strict priority is up to 5.
  • The 3 new classes are not supported with non-EX and non-FX switches.
  • If traffic flows between non-EX or non-FX switches and EX or FX switches, the traffic will use QoS level 3.
  • For communicating with FEX for new classes, the traffic carries a Layer 2 COS value of 0.

• Starting with Release 4.0(1), you can configure QoS Class or create a Custom QoS Policy to apply on an L3Out Interface.

## Configuring QoS Directly on L3Out Using GUI

This section describes how to configure QoS directly on an L3Out. This is the preferred way of configuring L3Out QoS starting with Cisco APIC Release 4.0(1).

### Procedure

**Step 1** From the main menu bar, select **Tenants > <tenant-name>**.
Step 2
In the left-hand navigation pane, expand Tenant <tenant-name> > Networking > External Routed Networks > <routed-network-name> > Logical Node Profiles > <node-profile-name> > Logical Interface Profiles > <interface-profile-name>.

You may need to create new network, node profile, and interface profile if none exist.

Step 3
In the main window pane, configure custom QoS for your L3Out.

You can choose to configure a standard QoS level priority using the QoS Priority dropdown menu. Alternatively, you can set an existing or create a new custom QoS policy from the Custom QoS Policy dropdown.

---

### Configuring QoS Directly on L3Out Using CLI

This section describes how to configure QoS directly on an L3Out. This is the preferred way of configuring L3Out QoS starting with Cisco APIC Release 4.0(1).

You can configure QoS for L3Out on one of the following objects:

- Switch Virtual Interface (SVI)
- Sub Interface
- Routed Outside

**Procedure**

**Step 1** Configure QoS priorities for a L3Out SVI.

**Example:**

```bash
interface vlan 19
  vrf member tenant DT vrf dt-vrf
  ip address 107.2.1.252/24
  description 'SVI19'
  service-policy type qos VrfQos006 // for custom QoS attachment
  set qos-class level6 // for set QoS priority
exit
```

**Step 2** Configure QoS priorities for a sub-interface.

**Example:**

```bash
interface ethernet 1/48.10
  vrf member tenant DT vrf inter-tentant-ctx2 l3out L4_E48_inter_tennant
  ip address 210.2.0.254/16
  service-policy type qos vrfQos002
  set qos-class level5
```

**Step 3** Configure QoS priorities for a routed outside.

**Example:**

```bash
interface ethernet 1/37
  no switchport
  vrf member tenant DT vrf dt-vrf l3out L2E37
```
ip address 30.1.1.1/24
service-policy type qos vrfQos002
set qos-class level5
exit

Configuring QoS Directly on L3Out Using REST API

This section describes how to configure QoS directly on an L3Out. This is the preferred way of configuring L3Out QoS starting with Cisco APIC Release 4.0(1).

You can configure QoS for L3Out on one of the following objects:

- Switch Virtual Interface (SVI)
- Sub Interface
- Routed Outside

Procedure

**Step 1** Configure QoS priorities for a L3Out SVI.

**Example:**
```xml
<l3extLIfP descr=""
    dn="uni/tn-DT/out-L3_4_E2_24 SVI17/lnodep-L3_4_E2_24/lifp-L3_4_E2_24_SVI_19"
    name="L3_4_E2_24_SVI_19" prio="level6" tag="yellow-green">
    <l3extRsPathL3OutAtt addr="0.0.0.0" autostate="disabled" descr="SVI19" encap="vlan-19"
        encapScope="local" ifInstT="ext-svi" ipv6Dad="enabled" llAddr="::"
        mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
        tDn="topology/pod-1/protpaths-103-104/pathep-[V_L3_l4_2-24]"
        targetDscp="unspecified">
        <l3extMember addr="107.2.1.253/24" ipv6Dad="enabled" llAddr="::" side="B"/>
        <l3extMember addr="107.2.1.252/24" ipv6Dad="enabled" llAddr="::" side="A"/>
    </l3extRsPathL3OutAtt>
    <l3extRsLIfPCustQosPol tnQosCustomPolName="VrfQos006"/>
</l3extLIfP>
```

**Step 2** Configure QoS priorities for a sub-interface.

**Example:**
```xml
<l3extLIfP dn="uni/tn-DT/out-L4E48_inter_tenant/lnodep-L4E48_inter_tenant/lifp-L4E48"
    name="L4E48" prio="level4" tag="yellow-green">
    <l3extRsPathL3OutAtt addr="210.1.0.254/16" autostate="disabled" descr="" encap="vlan-20"
        encapScope="local" ifInstT="sub-interface" ipv6Dad="enabled"
        llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
        tDn="topology/pod-1/paths-104/pathep-[eth1/48]"
        targetDscp="unspecified">
        <l3extRsNdIfPol annotation="" tnNdIfPolName=""/>
        <l3extRsLIfP CustQosPol annotation="" tnQosCustomPolName="vrfQos002"/>
    </l3extRsPathL3OutAtt>
</l3extLIfP>
```

**Step 3** Configure QoS priorities for a routed outside.
Example:

```xml
<l3extLIfP dn="uni/tn-DT/out-L2E37/lnodep-L2E37/lifp-L2E37OUT"
    name="L2E37OUT" prio="level5" tag="yellow-green">
  <l3extRsPathL3OutAtt addr="30.1.1.1/24" autostate="disabled" encap="unknown"
    encapScope="local" ifInstT="l3-port" ipv6Dad="enabled"
    llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular"
    mtu="inherit" targetDscp="unspecified"
    tDn="topology/pod-1/paths-102/pathep-[eth1/37]"/>
  <l3extRsNdIfPol annotation="" tnNdIfPolName=""/>
  <l3extRsLIfPCustQosPol tnQosCustomPolName="vrfQos002"/>
</l3extLIfP>
```

 Configuring QoS Contract for L3Out Using REST API

This section describes how to configure QoS for L3Outs using Contracts.

**Note**
Starting with Release 4.0(1), we recommend using custom QoS policies for L3Out QoS as described in Configuring QoS Directly on L3Out Using REST API, on page 64 instead.

**Procedure**

**Step 1**
When configuring the tenant, VRF, and bridge domain, configure the VRF for egress mode (pcEnfDir="egress") with policy enforcement enabled (pcEnfPref="enforced"). Send a post with XML similar to the following example:

Example:

```xml
<fvTenant name="t1">
  <fvCtx name="v1" pcEnfPref="enforced" pcEnfDir="egress"/>
  <fvBD name="bd1">
    <fvRsCtx tnFvCtxName="v1"/>
    <fvRsSubnet ip="44.44.44.1/24" scope="public"/>
    <fvRsBDToOut tnL3extOutName="l3out1"/>
  </fvBD>"/>
</fvTenant>
```

**Step 2**
When creating the filters and contracts to enable the EPGs participating in the L3Out to communicate, configure the QoS priority. The contract in this example includes the QoS priority, level1, for traffic ingressing on the L3Out. Alternatively, it could define a target DSCP value. QoS policies are supported on either the contract or the subject.

The filter also has the matchDscp="EF" criteria, so that traffic with this specific TAG received by the L3out processes through the queue specified in the contract subject.

**Note**
VRF enforcement should be ingress, for QOS or custom QOS on L3out interface, VRF enforcement need be egress, only when the QOS classification is going to be done in the contract for traffic between EPG and L3out or L3out to L3out.
Configuring QoS Contract for L3Out Using NX-OS Style CLI

This section describes how to configure QoS for L3Outs using Contracts.

**Note**

Starting with Release 4.0(1), we recommend using custom QoS policies for L3Out QoS as described in Configuring QoS Directly on L3Out Using CLI, on page 63 instead.

**Procedure**

**Step 1**

Configure the VRF for egress mode and enable policy enforcement to support QoS priority enforcement on the L3Out.

**Example:**

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# contract enforce egress
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)#
```

**Step 2**

Configure QoS.

When creating filters (access-list), include the `match dscp` command with target DSCP level.

When configuring contracts, include the QoS class for traffic ingressing on the L3Out. Alternatively, you can define a target DSCP value. QoS policies are supported on either the contract or the subject.

VRF enforcement must be ingress, for QoS or custom QoS on L3out interface, VRF enforcement need be egress, only when the QOS classification is going to be done in the contract for traffic between EPG and L3out or L3out to L3out.

**Note**

If QoS classification is set in the contract and VRF enforcement is egress, then contract QoS classification would override the L3Out interface QoS or Custom QoS classification.

**Example:**

```xml
<vzFilter name="http-filter">
  <vzEntry name="http-e" etherT="ip" prot="tcp" matchDscp="EF"/>
</vzFilter>
<vzBrCP name="httpCtrct" prio="level1" scope="context">
  <vzSubj name="subj1">
    <vzRsSubjFiltAtt tnVzFilterName="http-filter"/>
  </vzSubj>
</vzBrCP>
```
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# match dscp EF
apic1(config-tenant-acl)# exit
apic1(config-tenant-contr)# contract httpCtrct
apic1(config-tenant-contr)# scope vrf
apic1(config-tenant-contr)# qos-class level1
apic1(config-tenant-contr)# subject http-subject
apic1(config-tenant-contr-subj)# access-group http-filter both
apic1(config-tenant-contr-subj)# exit
apic1(config-tenant-contr)# exit
apic1(config)#

Configuring QoS Contracts for L3Outs Using Cisco APIC GUI

This section describes how to configure QoS for L3Outs using Contracts.

---

**Note**

Starting with Release 4.0(1), we recommend using custom QoS policies for L3Out QoS as described in Configuring QoS Directly on L3Out Using GUI, on page 62 instead.

---

**Procedure**

**Step 1**

Configure the VRF instance for the tenant consuming the L3Out to support QoS to be enforced on the border leaf switch that is used by the L3Out.

a) From the main menu bar, choose Tenants > <tenant-name>.
b) In the Navigation pane, expand Networking, right-click VRFS, and choose Create VRF.
c) Enter the name of the VRF.
d) In the Policy Control Enforcement Preference field, choose Enforced.

VRF enforcement should be ingress, for QoS or custom QoS on L3out interface, VRF enforcement need be egress, only when the QOS classification is going to be done in the contract for traffic between EPG and L3out or L3out to L3out.

e) In the Policy Control Enforcement Direction choose Egress

Not required, please see the above comment.

f) Complete the VRF configuration according to the requirements for the L3Out.

**Step 2**

When configuring filters for contracts to enable communication between the EPGs consuming the L3Out, include a QoS class or target DSCP to enforce the QoS priority in traffic ingressing through the L3Out.

a) On the Navigation pane, under the tenant that will consume the L3Out, expand Contracts, right-click Filters and choose Create Filter.
b) In the Name field, enter a filter name.
c) In the Entries field, click + to add a filter entry.
d) Add the Entry details, click **Update** and **Submit**.

e) Expand the previously created filter and click on a filter entry.

e) Set the **Match DSCP** field to the desired DSCP level for the entry, for example, **EF**.

**Step 3**  
Add a contract.

a) Under **Contracts**, right-click **Standard** and choose **Create Contract**.

b) Enter the name of the contract.

c) In the **QoS Class** field, choose the QoS priority for the traffic governed by this contract. Alternatively, you can choose a **Target DSCP** value.

   If QOS classification is set in the contract and VRF enforcement is egress, then contract QOS classification would override the L3out interface QOS or Custom QOS classification. So either we need to configure this one or the new one.

d) Click the **+** icon on **Subjects** to add a subject to the contract.

e) Enter a name for the subject.

f) In the QoS Priority field, choose the desired priority level. You cannot choose **Unspecified**.

g) Under **Filter Chain**, click the **+** icon on **Filters** and choose the filter you previously created, from the drop down list.

h) Click **Update**.

i) On the **Create Contract Subject** dialog box, click **OK**.
Routing Protocol Support

This chapter contains the following sections:

- About Routing Protocol Support, on page 69
- BGP External Routed Networks with BFD Support, on page 69
- OSPF External Routed Networks, on page 102
- EIGRP External Routed Networks, on page 108

About Routing Protocol Support

Routing within the Cisco ACI fabric is implemented using BGP (with BFD support) and the OSPF or EIGRP routing protocols.

IP source routing is not supported in the ACI fabric.

BGP External Routed Networks with BFD Support

The following sections provide more information on BGP external routed networks with BFD support.

Guidelines for Configuring a BGP Layer 3 Outside Network Connection

When configuring a BGP external routed network, follow these guidelines:

- Whenever a router ID is created on a leaf switch, it creates an internal loopback address. When setting up a BGP connection on a leaf switch, your router ID cannot be the same as the interface IP address as it is not supported on the ACI leaf switch. The router ID must be a different address in a different subnet. On the external Layer 3 device, the router ID can be the loopback address or an interface address. Ensure that the route to leaf router ID is present in the routing table of the Layer3 device either through static route or OSPF configuration. Also, when setting up the BGP neighbor on a Layer 3 device, the peer IP address that is used must be the router ID of the leaf switch.

- While configuring two external Layer 3 networks with BGP on the same node, loopback addresses must be explicitly defined. Failing to follow this guideline can prevent BGP from being established.

- By definition, the router ID is a loopback interface. To change the router ID and assign a different address for loopback, you must create a loopback interface policy. (The loopback policy can be configured as one for each address family, IPv4 and IPv6.) If you do not wish to create a loopback policy, then you
can enable a router ID loopback which is enabled by default. If the router ID loopback is disabled, no loopback is created for the specific Layer 3 outside on which it is deployed.

- This configuration task is applicable for iBGP and eBGP. If the BGP configuration is on a loopback address then it can be an iBGP session or a multi-hop eBGP session. If the peer IP address is for a physical interface where the BGP peer is defined, then the physical interface is used.

- The user must configure an IPv6 address to enable peering over loopback using IPv6.

- The autonomous system feature can only be used for eBGP peers. It enables a router to appear to be a member of a second autonomous system (AS), in addition to its real AS. Local AS allows two ISPs to merge without modifying peering arrangements. Routers in the merged ISP become members of the new autonomous system but continue to use their old AS numbers for their customers.

- Starting with release 1.2(1x), tenant networking protocol policies for BGP l3extOut connections can be configured with a maximum prefix limit that enables monitoring and restricting the number of route prefixes received from a peer. Once the max prefix limit is exceeded, a log entry can be recorded, further prefixes can be rejected, the connection can be restarted if the count drops below the threshold in a fixed interval, or the connection is shut down. Only one option can be used at a time. The default setting is a limit of 20,000 prefixes, after which new prefixes are rejected. When the reject option is deployed, BGP accepts one more prefix beyond the configured limit and the APIC raises a fault.

---

**Note**

Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link. On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface. For the appropriate MTU values for each platform, see the relevant configuration guides.

We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as: `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1`.

---

### BGP Connection Types and Loopback Guidelines

The ACI supports the following BGP connection types and summarizes the loopback guidelines for them:

<table>
<thead>
<tr>
<th>BGP Connection Type</th>
<th>Loopback required</th>
<th>Loopback same as Router ID</th>
<th>Static/OSPF route required</th>
</tr>
</thead>
<tbody>
<tr>
<td>iBGP direct</td>
<td>No</td>
<td>Not applicable</td>
<td>No</td>
</tr>
</tbody>
</table>

---

Routing Protocol Support

---

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
### Configuring BGP External Routed Networks

Use the procedures in the following sections to configure BGP external routed networks.

#### Configuring BGP External Routed Network Using the GUI

**Before you begin**

The tenant, VRF, and bridge domain where you configure the BGP external routed network is already created.

**Procedure**

<table>
<thead>
<tr>
<th>BGP Connection Type</th>
<th>Loopback required</th>
<th>Loopback same as Router ID</th>
<th>Static/OSPF route required</th>
</tr>
</thead>
<tbody>
<tr>
<td>iBGP loopback peering</td>
<td>Yes, a separate loopback per L3Out</td>
<td>No, if multiple Layer 3 out are on the same node</td>
<td>Yes</td>
</tr>
<tr>
<td>eBGP direct</td>
<td>No</td>
<td>Not applicable</td>
<td>No</td>
</tr>
<tr>
<td>eBGP loopback peering (multi-hop)</td>
<td>Yes, a separate loopback per L3Out</td>
<td>No, if multiple Layer 3 out are on the same node</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- **Step 1** In the Navigation pane, expand **Tenant_name > Networking > External Routed Networks**.
- **Step 2** Right-click, and click **Create Routed Outside**.
- **Step 3** In the **Create Routed Outside** dialog box, perform the following actions:
  a) In the **Name** field, enter a name for the external routed network policy.
  b) Click the **BGP** checkbox.
     - **Note**: BGP peer reachability must be available in one of two ways. You must either configure static routes or enable OSPF.
  c) (Optional) In the **Route Control Enforcement** field, check the **Import** check box.
     - **Note**: Check this check box if you wish to enforce import control with BGP.
  d) From the **VRF** field drop-down list, choose the desired VRF.
  e) Expand the **Route Control for Dampening** field, and choose the desired address family type and route dampening policy. Click **Update**.
     - In this step, the policy can be created either with step 4 or there is also an option to create route profile in the drop-down list where the policy name is selected.
  f) Expand **Nodes and Interfaces Protocol Policies**.
  g) In the **Create Node Profile** dialog box, enter a name for the node profile.
  h) Expand **Nodes**.
  i) From the **Select Node** dialog box, from the **Node ID** field drop-down list, choose a node.
j) In the **Router ID** field, enter the router ID.

k) Expand **Loopback Address**, and in the **IP** field, enter the IP address. Click **Update**.

   **Note** Enter an IPv6 address. If you did not add the router ID in the earlier step, you can add an IPv4 address in the **IP** field.

l) Click **OK**.

**Step 4**

In the Navigation pane, expand **Tenant_name > Networking > Route Profiles**. Right-click **Route Profiles**, and click **Create Route Profile**. In the **Create Route Profile** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the route control VRF.

b) Expand the **Create Route Control Context** dialog box.

c) In the **Name** field, enter a name for the route control VRF.

d) From the **Set Attribute** drop-down list, choose **Create Action Rule Profile**.

   When creating an action rule, set the route dampening attributes as desired.

**Step 5**

In the **Create Interface Profiles** dialog box, perform the following actions:

a) In the **Name** field, enter an interface profile name.

b) In the **Interfaces** area, choose the desired interface tab, and then expand the interface.

**Step 6**

In the **Select Routed Interface** dialog box, perform the following actions:

a) From the **Path** field drop-down list, choose the node and the interface.

b) In the **IP Address** field, enter the IP address.

   **Note** Depending upon your requirements, you can add an IPv6 address or an IPv4 address.

c) (Optional) If you entered an IPv6 address in the earlier step, in the **Link-local Address** field, enter an IPv6 address.

d) Expand **BGP Peer Connectivity Profile** field.

**Step 7**

In the **Create Peer Connectivity Profile** dialog box, perform the following actions:

a) In the **Peer Address** field, the dynamic neighbor feature is available. If desired by the user, any peer within a specified subnet can communicate or exchange routes with BGP.

   Enter an IPv4 of an IPv6 address to correspond with IPv4 or IPv6 addresses entered in the earlier in the steps.

b) In the **BGP Controls** field, check the desired controls.

c) In the **Autonomous System Number** field, choose the desired value.

d) (Optional) In the **Weight for routes from this neighbor** field, choose the desired value.

e) (Optional) In the **Private AS Control** field, check the check box for **Remove AS**.

f) (Optional) In the **Local Autonomous System Number Config** field, choose the desired value.

   Optionally required for the local autonomous system feature for eBGP peers.

g) (Optional) In the **Local Autonomous System Number** field, choose the desired value.

   Optionally required for the local autonomous system feature for eBGP peers.

   **Note** The value in this field must not be the same as the value in the **Autonomous System Number** field.

h) Click **OK**.
Step 8
Perform the following actions:

a) In the **Select Routed Interface** dialog box, click **OK**.

b) In the **Create Interface Profile** dialog box, click **OK**.

c) In the **Create Node Profile** dialog box, click **OK**.
The **External EPG Networks** area is displayed.

d) In **Create Routed Outside** dialog box, choose the node profile you created earlier, and click **Next**.

Step 9
Expand **External EPG Networks**, and in the **Create External Network** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the external network.

b) Expand **Subnet**.

c) In the **Create Subnet** dialog box, in the **IP address** field, enter the subnet addresses as required.

**Note**
Enter an IPv4 or IPv6 address depending upon what you have entered in earlier steps.

When creating the external subnet, you must configure either both the BGP loopbacks in the prefix EPG or neither of them. If you configure only one BGP loopback, then BGP neighborship is not established.

d) In the **Scope** field, check the check boxes for **Export Route Control Subnet**, **Import Route Control Subnet**, and **Security Import Subnet**. Click **OK**.

**Note**
Check the **Import Route Control Subnet** check box if you wish to enforce import control with BGP.

Step 10
In the **Create External Network** dialog box, click **OK**.

Step 11
In the **Create Routed Outside** dialog box, click **Finish**.
The eBGP is configured for external connectivity.

---

**Configuring BGP External Routed Network Using the NX-OS Style CLI**

**Procedure**

The following shows how to configure the BGP external routed network using the NX-OS CLI:

**Example:**

```bash
apic1(config-leaf)# template route-profile damp_rp tenant t1
This template will be available on all leaves where tenant t1 has a VRF deployment
apic1(config-leaf-template-route-profile)# set dampening 15 750 2000 60
apic1(config-leaf-template-route-profile)# exit
apic1(config-leaf)#
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant t1 vrf ctx3
apic1(config-leaf-bgp-vrf)# neighbor 32.0.1.0/24 l3out l3out-bgp
apic1(config-leaf-bgp-vrf-neighbor)# update-source ethernet 1/16.401
apic1(config-leaf-bgp-vrf-neighbor)# address-family ipv4 unicast
apic1(config-leaf-bgp-vrf-neighbor-af)# weight 400
apic1(config-leaf-bgp-vrf-neighbor-af)# exit
apic1(config-leaf-bgp-vrf-neighbor)# remote-as 65001
apic1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive
```
api1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive-all
api1(config-leaf-bgp-vrf-neighbor)# private-as-control remove-exclusive-all-replace-as
api1(config-leaf-bgp-vrf-neighbor)# exit
api1(config-leaf-bgp-vrf)# address-family ipv4 unicast
api1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
This template will be inherited on all leaves where VRF ctx3 has been deployed
api1(config-leaf-bgp-vrf-af)# exit
api1(config-leaf-bgp-vrf)# address-family ipv6 unicast
api1(config-leaf-bgp-vrf-af)# inherit bgp dampening damp_rp
This template will be inherited on all leaves where VRF ctx3 has been deployed
api1(config-leaf-bgp-vrf-af)# exit

Configuring BGP External Routed Network Using the REST API

Before you begin
The tenant where you configure the BGP external routed network is already created.
The following shows how to configure the BGP external routed network using the REST API:
For Example:

Procedure

Example:

```xml
<l3extOut descr="" dn="uni/tn-t1/out-l3out-bgp" enforceRtctrl="export" name="l3out-bgp"
    ownerKey="" ownerTag="" targetDscp="unspecified">
    <l3extRsEctx tnFvCtxName="ctx3"/>
    <l3extRsNodeP configIssues="" descr="" name="l3extLNodeP_1" ownerKey="" ownerTag=""
tag="yellow-green" targetDscp="unspecified">
        <l3extLNodeP configIssues="" descr="" name="l3extLNodeP_2" ownerKey="" ownerTag="" tag="yellow-green">
            <l3extRsNodeL3OutAtt rtrId="1.1.1.1" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
            <l3extLIfP descr="" name="l3extLIfP_2" ownerKey="" ownerTag="" tag="yellow-green">
                <l3extRsNdIfPol tnNdIfPolName="/"/>
                <l3extRsIngressQosDppPol tnQosDppPolName="/"/>
                <l3extRsEgressQosDppPol tnQosDppPolName="/"/>
                <l3extRsPathL3OutAtt addr="3001::31:0:1:2/120" descr="/" encap="vlan-3001" encapScope="local"
                    ifInstT="sub-interface" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit"
                    tDn="topology/pod-1/path-paths-101/pathep-[eth1/8]" targetDscp="unspecified">
                    <bgpPeerP addr="3001::31:0:1:0/120" allowedSelfAsCnt="3" ctrl="send-com,send-ext-com"
descr="/" name="/" peerCtrl="/" privateASctrl="/remove-all,remove-exclusive,replace-as"
ttl="1" weight="1000">
                        <bgpRsPeerPfxPol name="/"/>
                        <bgpAsP addr="3001" descr="/"/>
                    </bgpPeerP>
                </l3extRsPathL3OutAtt>
            </l3extLIfP>
        </l3extRsNodeL3OutAtt>
    </l3extRsNodeP>
</l3extOut>
```

Routing Protocol Support

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Configuring BGP Max Path

The following feature enables you to add the maximum number of paths to the route table to enable equal cost, multipath load balancing.

Configuring BGP Max Path Using the GUI

Before you begin

The appropriate tenant and the BGP external routed network are created and available.

Procedure

**Step 1** Log in to the APIC GUI, and on the menu bar, click Tenants > <Your_Tenant> > Networking > Protocol Policies > BGP > BGP Address Family Context and right click Create BGP Address Family Context Policy.

**Step 2** In the Create BGP Address Family Context Policy dialog box, perform the following tasks:

a) In the Name field, enter a name for the policy.
b) Click the eBGP Distance field confirm the value for your implementation.
c) Click the iBGP Distance field confirm the value for your implementation.
d) Click the Local Distance field confirm the value for your implementation.
e) Click the eBGP Max ECMP field confirm the value for your implementation.
f) Click the iBGP Max ECMP field confirm the value for your implementation.
g) Click Submit after you have updated your entries.

Step 3  Click Tenants > <Your_Tenant> > Networking > VRFs > <your_VRF>
Step 4  Review the configuration details of the subject VRF.
Step 5  Access the BGP Context Per Address Family field and select IPv6 in the Address Family drop-down list.
Step 6  Access the BGP Address Family Context you created in the BGP Address Family Context drop-down list and associate it with the subject VRF.
Step 7  Click Submit.

Configuring BGP Max Path Using the NX-OS Style CLI

Before you begin:
The appropriate tenant and the BGP external routed network are created and available.

The two properties which enable you to configure more paths are maxEcmp and maxEcmpIbgp in the bgpCtxAfPol object. After you configure these two properties, they are propagated to the rest of your implementation.

Use the following commands when logged in to BGP:

maximum-paths [ibgp]
no maximum-paths [ibgp]

Example:

apic1(config)# leaf 101
apic1(config-leaf)# template bgp address-family newAf tenant t1
This template will be available on all nodes where tenant t1 has a VRF deployment
apic1(config-bgp-af)# maximum-paths ?
<1-16> Maximum number of equal-cost paths for load sharing. The default is 16.
ibgp Configure multipath for IBGP paths
apic1(config-bgp-af)# maximum-paths 10
apic1(config-bgp-af)# maximum-paths ibgp 8
apic1(config-bgp-af)# end
apic1#
no maximum-paths [ibgp]

Configuring BGP Max Path Using the REST API

This following example provides information on how to configure the BGP Max Path feature using the REST API:

```xml
<fvTenant descr="" dn="uni/tn-t1" name="t1">
  <fvCtx name="v1">
    <fvRsCtxToBgpCtxAfPol af="ipv4-ucast" tnBgpCtxAfPolName="bgpCtxPol1"/>
  </fvCtx>
  <bgpCtxAfPol name="bgpCtxPol1" maxEcmp="8" maxEcmpIbgp="4"/>
</fvTenant>
```
Configuring AS Path Prepend

Use the procedures in the following sections to configure AS Path Prepend.

Configuring AS Path Prepend

A BGP peer can influence the best-path selection by a remote peer by increasing the length of the AS-Path attribute. AS-Path Prepend provides a mechanism that can be used to increase the length of the AS-Path attribute by prepending a specified number of AS numbers to it.

AS-Path prepending can only be applied in the outbound direction using route-maps. AS Path prepending does not work in iBGP sessions.

The AS Path Prepend feature enables modification as follows:

<table>
<thead>
<tr>
<th>Prepend</th>
<th>Appends the specified AS number to the AS path of the route matched by the route map.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td>• You can configure more than one AS number. • 4 byte AS numbers are supported. • You can prepend a total 32 AS numbers. You must specify the order in which the AS Number is inserted into the AS Path attribute.</td>
</tr>
</tbody>
</table>

| Prepend-last-as | Prepends the last AS numbers to the AS path with a range between 1 and 10. |

The following table describes the selection criteria for implementation of AS Path Prepend:

<table>
<thead>
<tr>
<th>Prepend</th>
<th>1</th>
<th>Prepends the specified AS number.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepend-last-as</td>
<td>2</td>
<td>Prepends the last AS numbers to the AS path.</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>Prepend(1)</td>
<td>Prepends the specified AS number.</td>
</tr>
</tbody>
</table>

Configuring AS Path Prepend Using the GUI

Before you begin

A configured tenant.

Procedure

Step 1 Log in to the APIC GUI, and on the menu bar, click Tenants > tenant_name > Networking > External Routed Networks > Set Rules for Route Maps and right click Create Set Rules For A Route Map.

Step 2 In the Create Set Rules For A Route Map dialog box, perform the following tasks:

a) In the Name field, enter a name.

b) Click the Set AS Path icon to open the Create Set AS Path dialog box.

Step 3 Select the criterion Prepend AS to prepend AS numbers.

Step 4 Enter the AS number and its order and then click Update. Repeat if multiple AS numbers must be prepended.
Step 5  Select the criterion **Prepend Last-AS** to prepend the last AS number a specified number of times.
Step 6  Enter **Count** (1-10).
Step 7  On the **Create Set Rules For A Route Map** display, confirm the listed criteria for the set rule based on AS Path and click **Finish**.
Step 8  On the APIC GUI menu bar, click **Tenants > tenant_name > Networking > External Routed Networks > Set Rules for Route Maps** and right click your profile.
Step 9  Confirm the **Set AS Path** values the bottom of the screen.

---

**Configuring AS Path Prepend Using the NX-OS Style CLI**

This section provides information on how to configure the AS Path Prepend feature using the NX-OS style command line interface (CLI).

**Before you begin**

A configured tenant.

**Procedure**

To modify the autonomous system path (AS Path) for Border Gateway Protocol (BGP) routes, you can use the **set as-path** command. The set as-path command takes the form of

```
set as-path {'prepend as-num [ ,... as-num ] | prepend-last-as num}
```

**Example:**

```
apic1(config-leaf-vrf-template-route-profile)# set as-path {'prepend as-num [ ,... as-num ] | prepend-last-as num}
apic1(config)# leaf 103
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# template route-profile rp1
apic1(config-leaf-vrf-template-route-profile)# set as-path ?
prepend Prepend to the AS-Path
prepend-last-as Prepend last AS to the as-path
apic1(config-leaf-vrf-template-route-profile)# set as-path prepend 100, 101, 102, 103
apic1(config-leaf-vrf-template-route-profile)# set as-path prepend-last-as 8
apic1(config-leaf-vrf-template-route-profile)# exit
apic1(config-leaf-vrf)# exit
```

**What to do next**

To disable AS Path prepend, use the no form of the shown command:

```
apic1(config-leaf-vrf-template-route-profile)# [no] set as-path { prepend as-num [ ,... as-num ] | prepend-last-as num}
apic1(config-leaf-vrf-template-route-profile)# exit
```

---

Routing Protocol Support

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Configuring AS Path Prepend Using the REST API

This following example provides information on how to configure the AS Path Prepend feature using the REST API:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<fvTenant name="coke">
  <rtctrlAttrP name="attrp1">
    <rtctrlSetASPath criteria="prepend">
      <rtctrlSetASPathASN asn="100" order="1"/>
      <rtctrlSetASPathASN asn="200" order="10"/>
      <rtctrlSetASPathASN asn="300" order="5"/>
    </rtctrlSetASPath/>
    <rtctrlSetASPath criteria="prepend-last-as" lastnum="9"/>
  </rtctrlAttrP>

  <l3extOut name="out1">
    <rtctrlProfile name="rp1">
      <rtctrlCtxP name="ctxp1" order="1">
        <rtctrlScope>
          <rtctrlRsScopeToAttrP tnRtctrlAttrPName="attrp1"/>
        </rtctrlScope>
      </rtctrlCtxP>
    </rtctrlProfile>
  </l3extOut>
</fvTenant>
```

BGP External Routed Networks with AS Override

Use the procedures in the following sections to configure BGP external routed networks with AS override.

About BGP Autonomous System Override

Loop prevention in BGP is done by verifying the Autonomous System number in the Autonomous System Path. If the receiving router sees its own Autonomous System number in the Autonomous System path of the received BGP packet, the packet is dropped. The receiving router assumes that the packet originated from its own Autonomous System and has reached the same place from where it originated initially. This setting is the default to prevent route loops from occurring.

The default setting to prevent route loops from occurring could create an issue if you use the same Autonomous System number along various sites and disallow user sites with identical Autonomous System numbers to link by another Autonomous System number. In such a scenario, routing updates from one site is dropped when the other site receives them.

To prevent such a situation from occurring, beginning with the Cisco APIC Release 3.1(2m), you can now enable the BGP Autonomous System override feature to override the default setting. You must also enable the Disable Peer AS Check at the same time.

The Autonomous System override function replaces the Autonomous System number from the originating router with the Autonomous System number of the sending BGP router in the AS Path of the outbound routes. This feature can be enabled per feature per address family (IPv4 or IPv6).

The Autonomous System Override feature is supported with GOLF Layer 3 configurations and Non-GOLF Layer 3 configurations.
Router 1 and Router 2 are the two customers with multiple sites (Site-A and Site-B). Customer Router 1 operates under AS 100 and customer Router 2 operates under AS 200.

The above diagram illustrates the Autonomous System (AS) override process as follows:

1. Router 1-Site-A advertises route 10.3.3.3 with AS100.
2. Router PE-1 propagates this as an internal route to PE2 as AS100.
3. Router PE-2 prepends 10.3.3.3 with AS121 (replaces 100 in the AS path with 121), and propagates the prefix.
4. Router 2-Site-B accepts the 10.3.3.3 update.

Configuring BGP External Routed Network with Autonomous System Override Enabled Using the GUI

Before you begin

- The Tenant, VRF, Bridge Domain are created.
- The External Routed Network that is in a non-GOLF setting, a logical node profile, and the BGP peer connectivity profile are created.

Procedure

**Step 1**
On the menu bar, choose Tenants > Tenant_name > Networking > External Routed Network > Non-GOLF Layer 3 Out_name > Logical Node Profiles.

**Step 2**
In the Navigation pane, choose the appropriate BGP Peer Connectivity Profile.

**Step 3**
In the Work pane, under Properties for the BGP Peer Connectivity Profile, in the BGP Controls field, perform the following actions: to...
a) Check the check box for the AS override field to enable the Autonomous System override function.
b) Check the check box for the Disable Peer AS Check field.

Note You must check the check boxes for AS override and Disable Peer AS Check for the AS override feature to take effect.

c) Choose additional fields as required.

Step 4 Click Submit.

---

Configuring BGP External Routed Network with Autonomous System Override Enabled Using the REST API

Procedure

Configure the BGP External Routed Network with Autonomous override enabled.

Note The line of code that is in bold displays the BGP AS override portion of the configuration. This feature was introduced in the Cisco APIC Release 3.1(2m).

Example:

```xml
<fvTenant name="coke">
  <fvCtx name="coke" status=""/>
  <bgpRtTargetP af="ipv4-ucast">
    <bgpRtTarget type="import" rt="route-target:as4-nn2:1234:1300"/>
    <bgpRtTarget type="export" rt="route-target:as4-nn2:1234:1300"/>
  </bgpRtTargetP>
  <bgpRtTargetP af="ipv6-ucast">
    <bgpRtTarget type="import" rt="route-target:as4-nn2:1234:1300"/>
    <bgpRtTarget type="export" rt="route-target:as4-nn2:1234:1300"/>
  </bgpRtTargetP>
</fvCtx>

<fvBD name="cokeBD">
  <!-- Association from Bridge Doamin to Private Network -->
  <fvRsCtx tnFvCtxName="coke"/>
  <fvRsBDToOut tnL3extOutName="routAccounting"/>
  <!-- Subnet behind the bridge domain-->
  <fvSubnet ip="20.1.1.1/16" scope="public"/>
  <fvSubnet ip="2000:1::1/64" scope="public"/>
</fvBD>

<fvBD name="cokeBD2">
  <!-- Association from Bridge Doamin to Private Network -->
  <fvRsCtx tnFvCtxName="coke"/>
  <fvRsBDToOut tnL3extOutName="routAccounting"/>
  <!-- Subnet behind the bridge domain-->
  <fvSubnet ip="30.1.1.1/16" scope="public"/>
</fvBD>

<vzBrCP name="webCtrct" scope="global">
  <vzSubj name="http">
    <vzRsSubjFiltAtt tnVzFilterName="default"/>
  </vzSubj>
</vzBrCP>
```
Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)

Routing Protocol Support

Configuring BGP External Routed Network with Autonomous System Override Enabled Using the REST API

```xml
<!/vzBrCP>

<!-- GOLF L3Out -->
<13extOut name="routeAccounting">
    <13extConsLbl name="golf_transit" owner="infra" status=""/>
    <bgpExtP/>
    <13extInstP name="accountingInst">
        <!--
        <l3extSubnet ip="192.2.2.0/24" scope="import-security,import-rtctrl" />
        <l3extSubnet ip="192.3.2.0/24" scope="export-rtctrl"/>
        <l3extSubnet ip="192.5.2.0/24" scope="export-rtctrl"/>
        <l3extSubnet ip="64:ff9b::c007:200/120" scope="export-rtctrl" />
        -->
        <l3extSubnet ip="0.0.0.0/0"
            scope="export-rtctrl,import-security"
            aggregate="export-rtctrl"
        />
    </13extInstP>
    <fvsRsProv tnVzBrCPName="webCtrct"/>
    <13extRsEctx tnFvCtxName="coke"/>
</13extOut>

<fvAp name="cokeAp">
    <fvAEPg name="cokeEPg"/>
    <fvRsBd tnFvBDName="cokeBD"/>
    <fvRsPathAtt tDn="topology/pod-1/paths-103/pathep-[eth1/20]" encap="vlan-100" instrImedcy="immediate" mode="regular"/>
    <fvRsCons tnVzBrCPName="webCtrct"/>
</fvAEPg>

<!-- Non GOLF L3Out-->
<13extOut name="NonGolfOut">
    <bgpExtP/>
    <13extLNodeP name="bLeaf">
        <!--
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="20.1.13.1"/>
        -->
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="20.1.13.1"/>
        <l3extLoopBackIfP addr="1.1.1.1"/>
    <ipRouteP ip="2.2.2.2/32" />
    <ipNextHopP nhAddr="20.1.12.3"/>
</13extRsLNodeP>

</13extRsLNodeP>
</13extOut>
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Configuring Per VRF Per Node BGP Timer Values

Use the procedures in the following sections to configure per VRF per node BGP timer values.

Per VRF Per Node BGP Timer Values

Prior to the introduction of this feature, for a given VRF, all nodes used the same BGP timer values.

With the introduction of the per VRF per node BGP timer values feature, BGP timers can be defined and associated on a per VRF per node basis. A node can have multiple VRFs, each corresponding to a `fvCtx`. A node configuration (`l3extLNodeP`) can now contain configuration for BGP Protocol Profile (`bgpProtP`) which in turn refers to the desired BGP Context Policy (`bgpCtxPol`). This makes it possible to have a different node within the same VRF contain different BGP timer values.

For each VRF, a node has a `bgpDom` concrete MO. Its name (primary key) is the VRF, `<fvTenant>:<fvCtx>`. It contains the BGP timer values as attributes (for example, `holdIntvl`, `kaIntvl`, `maxAsLimit`).

All the steps necessary to create a valid Layer 3 Out configuration are required to successfully apply a per VRF per node BGP timer. For example, MOs such as the following are required: `fvTenant`, `fvCtx`, `l3extOut`, `l3extInstP`, `LNodeP`, `bgpRR`.

On a node, the BGP timer policy is chosen based on the following algorithm:

- If `bgpProtP` is specified, then use `bgpCtxPol` referred to under `bgpProtP`.
- Else, if specified, use `bgpCtxPol` referred to under corresponding `fvCtx`.
- Else, if specified, use the default policy under the tenant, for example, `uni/tn-<tenant>/bgpCtxP-default`.
• Else, use the default policy under tenant common, for example, uni/tn-common/bgpCtxP-default. This one is pre-programmed.

Configuring a Per VRF Per Node BGP Timer Using the Advanced GUI

When a BGP timer is configured on a specific node, then the BGP timer policy on the node is used and the BGP policy timer associated with the VRF is ignored.

Before you begin

A tenant and a VRF are already configured.

Procedure

Step 1

Step 2
In the create BGP Timers Policy dialog box, perform the following actions:
  a) In the Name field, enter the BGP Timers policy name.
  b) In the available fields, choose the appropriate values as desired. Click Submit.

A BGP timer policy is created.

Step 3
Navigate to the External Routed Network, and create a Layer 3 Out with BGP enabled by performing the following actions:
  a) Right-click Create Routed Outside.
  b) In the Create Routed Outside dialog box, specify the name of the Layer 3 Out.
  c) Check the check box to enable BGP.
  d) Expand Nodes and Interfaces Protocol Policies.

Step 4
To create a new node, in the Create Node Profile dialog box, perform the following actions:
  a) In the Name field, enter a name for the node profile.
  b) In the BGP Timers field, from the drop-down list, choose the BGP timer policy that you want to associate with this specific node. Click Finish.

A specific BGP timer policy is now applied to the node.

Note
To associate an existing node profile with a BGP timer policy, right-click the node profile, and associate the timer policy.

If a timer policy is not chosen specifically in the BGP Timers field for the node, then the BGP timer policy that is associated with the VRF under which the node profile resides automatically gets applied to this node.

Step 5
To verify the configuration, in the Navigation pane, perform the following steps:
  a) Expand Layer 3 Out > External Routed Network_name > Logical Node Profiles > Logical Node Profiles_name > BGP Protocol Profile.
  b) In the Work pane, the BGP protocol profile that is associated with the node profile is displayed.
Configuring a Per VRF Per Node BGP Timer Using the REST API

The following example shows how to configure Per VRF Per node BGP timer in a node. Configure `bgpProtP` under `l3extLNodeP` configuration. Under `bgpProtP`, configure a relation (`bgpRsBgpNodeCtxPol`) to the desired BGP Context Policy (`bgpCtxPol`).

**Procedure**

Configure a node specific BGP timer policy on `node1`, and configure `node2` with a BGP timer policy that is not node specific.

**Example:**

```
POST https://apic-ip-address/mo.xml
```

```xml
<fvTenant name="tn1">
    <bgpCtxPol name="pol1" staleIntvl="25"/>
    <bgpCtxPol name="pol2" staleIntvl="35"/>
    <fvCtx name="ctx1">
        <fvRsBgpCtxPol tnBgpCtxPolName="pol1"/>
    </fvCtx>
    <l3extout name="out1">
        <l3extRsEctx toFvCtxName="ctx1"/>
    </l3extout>
    <l3extLNodeP name="node1">
        <bgpProtP name="protp1">
            <bgpRsBgpNodeCtxPol tnBgpCtxPolName="pol2"/>
        </bgpProtP>
    </l3extLNodeP>
    <l3extLNodeP name="node2"/>
</fvTenant>
```

In this example, `node1` gets BGP timer values from policy `pol2`, and `node2` gets BGP timer values from `pol1`. The timer values are applied to the `bgpDom` corresponding to VRF `tn1:ctx1`. This is based upon the BGP timer policy that is chosen following the algorithm described in the *Per VRF Per Node BGP Timer Values* section.

Deleting a Per VRF Per Node BGP Timer Using the REST API

The following example shows how to delete an existing Per VRF Per node BGP timer in a node.

**Procedure**

Delete the node specific BGP timer policy on `node1`.

**Example:**

```
POST https://apic-ip-address/mo.xml
```

```xml
<fvTenant name="tn1">
    <bgpCtxPol name="pol1" staleIntvl="25"/>
    <bgpCtxPol name="pol2" staleIntvl="35"/>
    <fvCtx name="ctx1">
        <fvRsBgpCtxPol tnBgpCtxPolName="pol1"/>
    </fvCtx>
    <l3extout name="out1"/>
</fvTenant>
```
The code phrase `<bgpProtP name="protp1" status="deleted" />` in the example above, deletes the BGP timer policy. After the deletion, node1 defaults to the BGP timer policy for the VRF with which node1 is associated, which is pol1 in the above example.

### Configuring a Per VRF Per Node BGP Timer Policy Using the NX-OS Style CLI

#### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Configure BGP ASN and the route reflector before creating a timer policy.</td>
</tr>
<tr>
<td>Example:</td>
<td>api1(config)# bgp-fabric&lt;br&gt;api1(config)# route-reflector spine 102&lt;br&gt;api1(config-bgp-fabric)# as 42&lt;br&gt;api1(config-bgp-fabric)# exit&lt;br&gt;api1(config)# exit&lt;br&gt;api1#</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Create a timer policy.</td>
</tr>
<tr>
<td>Example:</td>
<td>api1(config)# leaf 101&lt;br&gt;api1(config-leaf)# template bgp timers pol1 tenant tn1&lt;br&gt;This template will be available on all nodes where tenant tn1 has a VRF deployment&lt;br&gt;api1(config-bgp-timers)# timers bgp 120 240&lt;br&gt;api1(config-bgp-timers)# graceful-restart stalepath-time 500&lt;br&gt;api1(config-bgp-timers)# maxas-limit 300&lt;br&gt;api1(config-bgp-timers)# exit&lt;br&gt;api1(config-leaf)# exit&lt;br&gt;api1(config)# exit&lt;br&gt;api1#</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Display the configured BGP policy.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
## Command or Action

```
apic1# show run leaf 101 template bgp timers pol7
# Command: show running-config leaf 101 template bgp timers pol7
leaf 101
template bgp timers pol7 tenant tn1
timers bgp 120 240
graceful-restart stalepath-time 500
maxas-limit 300
exit
exit
```

## Purpose

Refer to a specific policy at a node.

**Example:**
```
apic1# config
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 42
apic1(config-leaf-bgp)# vrf member tenant
  tn1 vrf ctx1
apic1(config-leaf-bgp-vrf)# inherit node-only bgp timer pol7
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit
apic1(config)# exit
apic1#
```

## Step 4

Display the node specific BGP timer policy.

**Example:**
```
apic1# show run leaf 101 router bgp 42 vrf member tenant
  tn1 vrf ctx1
# Command: show running-config leaf 101 router bgp 42 vrf member tenant
  tn1 vrf ctx1
leaf 101
router bgp 42
  vrf member tenant
  tn1 vrf ctx1
  inherit node-only bgp timer pol7
  exit
  exit
  exit
apic1#
```

---

### Troubleshooting Inconsistency and Faults

The following inconsistencies or faults could occur under certain conditions:

If different Layer 3 Outs (l3Out) are associated with the same VRF (fvCtx), and on the same node, the bgpProtP is associated with different policies (bgpCtxPol), a fault will be raised. In the case of the example below, both Layer 3 Outs (out1 and out2) are associated with the same VRF (ctx1). Under out1, node1 is
associated with the BGP timer protocol pol1 and under out2, node1 is associated with a different BGP timer protocol pol2. This will raise a fault.

tn1
cxt1
out1
cxt1
node1
 protp pol1

out2
cxt1
node1
 protp pol2

If such a fault is raised, change the configuration to remove the conflict between the BGP timer policies.

**Configuring BFD Support**

Use the procedures in the following sections to configure BFD support.

**Bidirectional Forwarding Detection**

Use Bidirectional Forwarding Detection (BFD) to provide sub-second failure detection times in the forwarding path between ACI fabric border leaf switches configured to support peering router connections.

BFD is particularly useful in the following scenarios:

- When the peering routers are connected through a Layer 2 device or a Layer 2 cloud where the routers are not directly connected to each other. Failures in the forwarding path may not be visible to the peer routers. The only mechanism available to control protocols is the hello timeout, which can take tens of seconds or even minutes to time out. BFD provides sub-second failure detection times.

- When the peering routers are connected through a physical media that does not support reliable failure detection, such as shared Ethernet. In this case too, routing protocols have only their large hello timers to fall back on.

- When many protocols are running between a pair of routers, each protocol has its own hello mechanism for detecting link failures, with its own timeouts. BFD provides a uniform timeout for all the protocols, which makes convergence time consistent and predictable.

Observe the following BFD guidelines and limitations:

- Starting from APIC Release 3.1(1), BFD between leaf and spine switches is supported on fabric-interfaces for IS-IS. In addition, BFD feature on spine switch is supported for OSPF and static routes.

- BFD is supported on modular spine switches that have -EX and -FX line cards (or newer versions), and BFD is also supported on the Nexus 9364C non-modular spine switch (or newer versions).

- BFD between VPC peers is not supported.

- Multithop BFD is not supported.

- BFD over iBGP is not supported for loopback address peers.

- BFD sub interface optimization can be enabled in an interface policy. One sub-interface having this flag will enable optimization for all the sub-interfaces on that physical interface.
• BFD for BGP prefix peer not supported.

Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link. On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface.

For the appropriate MTU values for each platform, see the relevant configuration guides.

We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1`.

---

### Configuring BFD Globally on Leaf Switch Using the GUI

**Procedure**

**Step 1** On the menu bar, choose **Fabric > Access Policies**.

**Step 2** In the **Navigation** pane, expand the **Switch Policies > Policies > BFD**.

There are two types of bidirectional forwarding detection (BFD) configurations available:

• BFD IPV4

• BFD IPV6

For each of these BFD configurations, you can choose to use the default policy or create a new one for a specific switch (or set of switches).

**Note** By default, the APIC controller creates default policies when the system comes up. These default policies are global, bi-directional forwarding detection (BFD) configuration polices. You can set attributes within that default global policy in the **Work** pane, or you can modify these default policy values. However, once you modify a default global policy, note that your changes affect the entire system (all switches). If you want to use a specific configuration for a particular switch (or set of switches) that is not the default, create a switch profile as described in the next step.

**Step 3** To create a switch profile for a specific global BFD policy (which is not the default), in the **Navigation** pane, expand the **Switch Policies > Profiles > Leaf Profiles**.

The **Profiles - Leaf Profiles** screen appears in the **Work** pane.

**Step 4** On the right side of the **Work** pane, under **ACTIONS**, select **Create Leaf Profile**.

The **Create Leaf Profile** dialog box appears.

**Step 5** In the **Create Leaf Profile** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the leaf switch profile.

b) In the **Description** field, enter a description of the profile. (This step is optional.)

c) In the **Switch Selectors** field, enter the appropriate values for **Name** (name the switch), **Blocks** (select the switch), and **Policy Group** (select **Create Access Switch Policy Group**).
The **Create Access Switch Policy Group** dialog box appears where you can specify the Policy Group identity properties.

**Step 6**
In the **Create Access Switch Policy Group** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the policy group.

b) In the **Description** field, enter a description of the policy group. (This step is optional.)

c) Choose a BFD policy type (**BFD IPV4 Policy** or **BFD IPV6 Policy**), then select a value (**default** or **Create BFD Global Ipv4 Policy** for a specific switch or set of switches).

**Step 7**
Click **SUBMIT**.

Another way to create a BFD global policy is to right-click on either **BFD IPV4** or **BFD IPV6** in the **Navigation** pane.

**Step 8**
To view the BFD global configuration you created, in the **Navigation** pane, expand the **Switch Policies > Policies > BFD**.

---

**Configuring BFD Globally on Spine Switch Using the GUI**

**Procedure**

**Step 1**
On the menu bar, choose **Fabric > Access Policies**.

**Step 2**
In the **Navigation** pane, expand the **Switch Policies > Policies > BFD**.

There are two types of bidirectional forwarding detection (BFD) configurations available:

- **BFD IPV4**
- **BFD IPV6**

For each of these BFD configurations, you can choose to use the default policy or create a new one for a specific switch (or set of switches).

**Note**
By default, the APIC controller creates default policies when the system comes up. These default policies are global, bi-directional forwarding detection (BFD) configuration polices. You can set attributes within that default global policy in the **Work** pane, or you can modify these default policy values. However, once you modify a default global policy, note that your changes affect the entire system (all switches). If you want to use a specific configuration for a particular switch (or set of switches) that is not the default, create a switch profile as described in the next step.

**Step 3**
To create a spine switch profile for a specific global BFD policy (which is not the default), in the **Navigation** pane, expand the **Switch Policies > Profiles > Spine Profiles**.

The **Profiles- Spine Profiles** screen appears in the **Work** pane.

**Step 4**
On the right side of the **Work** pane, under **ACTIONS**, select **Create Spine Profile**.

The **Create Spine Profile** dialog box appears.

**Step 5**
In the **Create Spine Profile** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the switch profile.

b) In the **Description** field, enter a description of the profile. (This step is optional.)

c) In the **Spine Selectors** field, enter the appropriate values for **Name** (name the switch), **Blocks** (select the switch), and **Policy Group** (select **Create Spine Switch Policy Group**).

The **Create Spine Switch Policy Group** dialog box appears where you can specify the Policy Group identity properties.
Step 6  In the **Create Spine Switch Policy Group** dialog box, perform the following actions:

a)  In the **Name** field, enter a name for the policy group.

b)  In the **Description** field, enter a description of the policy group. (This step is optional.)

c)  Choose a BFD policy type (**BFD IPV4 Policy** or **BFD IPV6 Policy**), then select a value (**default** or **Create BFD Global Ipv4 Policy** for a specific switch or set of switches).

Step 7  Click **SUBMIT**.

Another way to create a BFD global policy is to right-click on either **BFD IPV4** or **BFD IPV6** in the Navigation pane.

Step 8  To view the BFD global configuration you created, in the Navigation pane, expand the **Switch Policies > Policies > BFD**.

---

**Configuring BFD Globally on Leaf Switch Using the NX-OS Style CLI**

**Procedure**

Step 1  To configure the BFD IPV4 global configuration (bfdIpv4InstPol) using the NX-OS CLI:

**Example:**

```
apici# configure
apici(config)# template bfd ip bfd_ipv4_global_policy
apici(config-bfd)# [no] echo-address 1.2.3.4
apici(config-bfd)# [no] slow-timer 2500
apici(config-bfd)# [no] min-tx 100
apici(config-bfd)# [no] min-rx 70
apici(config-bfd)# [no] multiplier 3
apici(config-bfd)# [no] echo-rx-interval 500
apici(config-bfd)# exit
```

Step 2  To configure the BFD IPV6 global configuration (bfdIpv6InstPol) using the NX-OS CLI:

**Example:**

```
apici# configure
apici(config)# template bfd ipv6 bfd_ipv6_global_policy
apici(config-bfd)# [no] echo-address 34::1/64
apici(config-bfd)# [no] slow-timer 2500
apici(config-bfd)# [no] min-tx 100
apici(config-bfd)# [no] min-rx 70
apici(config-bfd)# [no] multiplier 3
apici(config-bfd)# [no] echo-rx-interval 500
apici(config-bfd)# exit
```

Step 3  To configure access leaf policy group (infraAccNodePGrp) and inherit the previously created BFD global policies using the NX-OS CLI:

**Example:**

```
apici# configure
apici(config)# template leaf-policy-group test_leaf_policy_group
apici(config-leaf-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apici(config-leaf-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apici(config-leaf-policy-group)# exit
```
### Configuring BFD Globally on Spine Switch Using the NX-OS Style CLI

Use this procedure to configure BFD globally on spine switch using the NX-OS style CLI.

#### Procedure

**Step 1** To configure the BFD IPv4 global configuration (bfdIpv4InstPol) using the NX-OS CLI:

**Example:**

```
apic1# configure
apic1(config)# template bfd ip bfd_ipv4_global_policy
apic1(config-bfd)# [no] echo-address 1.2.3.4
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

**Step 2** To configure the BFD IPv6 global configuration (bfdIpv6InstPol) using the NX-OS CLI:

**Example:**

```
apic1# configure
apic1(config)# template bfd ipv6 bfd_ipv6_global_policy
apic1(config-bfd)# [no] echo-address 34::1/64
apic1(config-bfd)# [no] slow-timer 2500
apic1(config-bfd)# [no] min-tx 100
apic1(config-bfd)# [no] min-rx 70
apic1(config-bfd)# [no] multiplier 3
apic1(config-bfd)# [no] echo-rx-interval 500
apic1(config-bfd)# exit
```

**Step 3** To configure spine policy group and inherit the previously created BFD global policies using the NX-OS CLI:

**Example:**

```
apic1# configure
apic1(config)# template spine-policy-group test_spine_policy_group
apic1(config-spine-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apic1(config-spine-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apic1(config-spine-policy-group)# exit
```

**Step 4** To associate the previously created spine policy group onto a spine switch using the NX-OS CLI:

**Example:**

```
apic1(config)# template spine-policy-group test_spine_policy_group
apic1(config-spine-policy-group)# [no] inherit bfd ip bfd_ipv4_global_policy
apic1(config-spine-policy-group)# [no] inherit bfd ipv6 bfd_ipv6_global_policy
apic1(config-spine-policy-group)# exit
```
Configuring BFD Globally Using the REST API

Procedure

The following REST API shows the global configuration for bidirectional forwarding detection (BFD):

Example:

```xml
<polUni>
  <infraInfra>
    <bfdIpv4InstPol name="default" echoSrcAddr="1.2.3.4" slowIntvl="1000" minTxIntvl="150"
      minRxIntvl="250" detectMult="5" echoRxIntvl="200"/>
    <bfdIpv6InstPol name="default" echoSrcAddr="34::1/64" slowIntvl="1000" minTxIntvl="150"
      minRxIntvl="250" detectMult="5" echoRxIntvl="200"/>
  </infraInfra>
</polUni>
```

Configuring BFD Interface Override Using the GUI

There are three supported interfaces (routed L3 interfaces, the external SVI interface, and the routed sub-interfaces) on which you can configure an explicit bi-directional forwarding detection (BFD) configuration. If you don't want to use the global configuration, yet you want to have an explicit configuration on a given interface, you can create your own global configuration, which gets applied to all the interfaces on a specific switch or set of switches. This interface override configuration should be used if you want more granularity on a specific switch on a specific interface.

Before you begin

A tenant has already been created.

Procedure

Step 1  On the menu bar, choose Tenant.

Step 2  In the Navigation pane (under Quick Start), expand the Tenant you created Tenant_name > Networking > External Routed Networks.

Step 3  Right-click on External Routed Networks and select Create Routed Outside. The Create Routed Outside dialog box appears.

Step 4  In the Create Routed Outside dialog box, under Define the Routed Outside, there should be an existing configuration already set up. If not, enter the values to define the identity of the Routed Outside configuration.
Step 5 Under Nodes And Interfaces Protocol Profiles, at the bottom of the Create Routed Outside dialog box, click the "+" (expand) button. The Create Node Profile dialog box appears.

Step 6 Under Specify the Node Profile, enter the name of the node profile in the Name field.

Step 7 Click the "+" (expand) button located to the right of the Nodes field. The Select Node dialog box appears.

Step 8 Under Select node and Configure Static Routes, select a node in the Node ID field.

Step 9 Enter the router ID in the Router ID field.

Step 10 Click OK. The Create Node Profile dialog box appears.

Step 11 Click the "+" (expand) button located to the right of the Interface Profiles field. The Create Interface Profile dialog box appears.

Step 12 Enter the name of the interface profile in the Name field.

Step 13 Select the desired user interface for the node you previously created, by clicking one of the Interfaces tabs:

- Routed Interfaces
- SVI
- Routed Sub-Interfaces

Step 14 In the BFD Interface Profile field, enter BFD details. In the Authentication Type field, choose No authentication or Keyed SHA1. If you choose to authenticate (by selecting Keyed SHA1), enter the Authentication Key ID, enter the Authentication Key (password), then confirm the password by re-entering it next to Confirm Key.

Step 15 For the BFD Interface Policy field, select either the common/default configuration (the default BFD policy), or create your own BFD policy by selecting Create BFD Interface Policy. If you select Create BFD Interface Policy, the Create BFD Interface Policy dialog box appears where you can define the BFD interface policy values.

Step 16 Click SUBMIT.

Step 17 To see the configured interface level BFD policy, navigate to Networking > Protocol Polices > BFD.

---

**Configuring BFD Interface Override Using the NX-OS Style CLI**

**Procedure**

**Step 1** To configure BFD Interface Policy (bfdIfPol) using the NX-OS CLI:

**Example:**

```
apicl# configure
apicl(config)# tenant t0
apicl(config-tenant)# vrf context v0
apicl(config-tenant-vrf)# exit
apicl(config-tenant)# exit
apicl(config)# leaf 101
apicl(config-leaf)# vrf context tenant t0 vrf v0
apicl(config-leaf-vrf)# exit
```
Step 2 To inherit the previously created BFD interface policy onto a L3 interface with IPv4 address using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface Ethernet 1/15
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password
```

Step 3 To inherit the previously created BFD interface policy onto an L3 interface with IPv6 address using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface Ethernet 1/15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# bfd ipv6 tenant mode
apic1(config-leaf-if)# bfd ipv6 inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ipv6 authentication keyed-sha1 key 10 key password
```

Step 4 To configure BFD on a VLAN interface with IPv4 address using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# vrf member tenant t0 vrf v0
apic1(config-leaf-if)# bfd ip tenant mode
apic1(config-leaf-if)# bfd ip inherit interface-policy bfdPol1
apic1(config-leaf-if)# bfd ip authentication keyed-sha1 key 10 key password
```

Step 5 To configure BFD on a VLAN interface with IPv6 address using the NX-OS CLI:

Example:

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface vlan 15
apic1(config-leaf-if)# ipv6 address 2001::10:1/64 preferred
apic1(config-leaf-if)# vrf member tenant t0 vrf v0
apic1(config-leaf-if)# bfd ipv6 tenant mode
```
Configuring BFD Interface Override Using the REST API

Procedure

The following REST API shows the interface override configuration for bidirectional forwarding detection (BFD):

Example:

```xml
<fvTenant name="ExampleCorp">
  <bfdIfPol name="bfdIfPol" minTxIntvl="400" minRxIntvl="400" detectMult="5" echoRxIntvl="400" echoAdminSt="disabled"/>
  <l3extOut name="l3-out">
    <l3extLNodeP name="leaf1">
      <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2"/>
      <l3extLIfP name='portIpv4'>
        <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInst='13-port' addr="10.0.0.1/24" mtu="1500"/>
        <bfdIfP type="sha1" key="password">
          <bfdRsIfPol tnBfdIfPolName='bfdIfPol'/>
        </bfdIfP>
      </l3extLIfP>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>
```

Configuring BFD Consumer Protocols Using the GUI

This procedure provides the steps to enable bi-directional forwarding detection (BFD) in the consumer protocols (OSPF, BGP, EIGRP, Static Routes, and IS-IS), which are consumers of the BFD feature. To consume the BFD on these protocols, you must enable a flag in them.

Note

These four consumer protocols are located in the left navigation pane under Tenant > Networking > Protocol Policies.

Before you begin

A tenant has already been created.
Procedure

Step 1
On the menu bar, choose Tenant.

Step 2
To configure BFD in the BGP protocol, in the Navigation pane (under Quick Start), expand the Tenant you created Tenant_name > Networking > Protocol Policies > BGP > BGP Peer Prefix.

Step 3
On the right side of the Work pane, under ACTIONS, select Create BGP Peer Prefix Policy. The Create BGP Peer Prefix Policy dialog box appears.

Note
You can also right-click on BGP Peer Prefix from the left navigation pane to select Create BGP Peer Prefix to create the policy.

Step 4
Enter a name in the Name field and provide values in the remaining fields to define the BGP peer prefix policy.

Step 5
Click SUBMIT.
The BGP peer prefix policy you created now appears under BGP Peer Prefix in the left navigation pane.

Step 6
In the Navigation pane, go back to Networking > External Routed Networks.

Step 7
Right-click on External Routed Networks and select Create Routed Outside.
The Create Routed Outside dialog box appears.

Step 8
In the Create Routed Outside dialog box, enter the name in the Name field. Then, to the right side of the Name field, select the BGP protocol.

Step 9
In the Nodes and Interfaces Protocol Profiles section, click the "+" (expand) button.
The Create Node Profile dialog box appears.

Step 10
In the BGP Peer Connectivity section, click the "+" (expand) button.
The Create BGP Peer Connectivity Profile dialog box appears.

Step 11
In the Create BGP Peer Connectivity Profile dialog box, next to Peer Controls, select Bidirectional Forwarding Detection to enable BFD on the BGP consumer protocol, shown as follows (or uncheck the box to disable BFD).

Step 12
To configure BFD in the OSPF protocol, in the Navigation pane, go to Networking > Protocol Policies > OSPF > OSPF Interface.

Step 13
On the right side of the Work pane, under ACTIONS, select Create OSPF Interface Policy.
The Create OSPF Interface Policy dialog box appears.

Note
You can also right-click on OSPF Interface from the left navigation pane and select Create OSPF Interface Policy to create the policy.

Step 14
Enter a name in the Name field and provide values in the remaining fields to define the OSPF interface policy.

Step 15
In the Interface Controls section of this dialog box, you can enable or disable BFD. To enable it, check the box next to BFD, which adds a flag to the OSPF consumer protocol, shown as follows (or uncheck the box to disable BFD).

Step 16
Click SUBMIT.

Step 17
To configure BFD in the EIGRP protocol, in the Navigation pane, go back to Networking > Protocol Policies > EIGRP > EIGRP Interface.

Step 18
On the right side of the Work pane, under ACTIONS, select Create EIGRP Interface Policy.
The Create EIGRP Interface Policy dialog box appears.

Note
You can also right-click on EIRGP Interface from the left navigation pane and select Create EIGRP Interface Policy to create the policy.
Step 19 Enter a name in the **Name** field and provide values in the remaining fields to define the OSPF interface policy.

Step 20 In the **Control State** section of this dialog box, you can enable or disable BFD. To enable it, check the box next to **BFD**, which adds a flag to the EIGRP consumer protocol (or uncheck the box to disable BFD).

Step 21 Click **SUBMIT**.

Step 22 To configure BFD in the Static Routes protocol, in the **Navigation** pane, go back to **Networking > External Routed Networks**.

Step 23 Right-click on **External Routed Networks** and select **Create Routed Outside**. The **Create Routed Outside** dialog box appears.

Step 24 In the **Define the Routed Outside** section, enter values for all the required fields.

Step 25 In the **Nodes and Interfaces Protocol Profiles** section, click the "+" (expand) button. The **Create Node Profile** dialog box appears.

Step 26 In the section **Nodes**, click the "+" (expand) button. The **Select Node** dialog box appears.

Step 27 In the **Static Routes** section, click the "+" (expand) button. The **Create Static Route** dialog box appears. Enter values for the required fields in this section.

Step 28 Next to **Route Control**, check the box next to **BFD** to enable (or uncheck the box to disable) BFD on the specified Static Route.

Step 29 Click **OK**.

Step 30 To configure BFD in the IS-IS protocol, in the **Navigation** pane go to **Fabric > Fabric Policies > Interface Policies > Policies > L3 Interface**.

Step 31 On the right side of the **Work** pane, under **ACTIONS**, select **Create L3 Interface Policy**. The **Create L3 Interface Policy** dialog box appears.

**Note** You can also right-click on **L3 Interface** from the left navigation pane and select **Create L3 Interface Policy** to create the policy.

Step 32 Enter a name in the **Name** field and provide values in the remaining fields to define the L3 interface policy.

Step 33 To enable BFD ISIS Policy, click **Enable**.

Step 34 Click **SUBMIT**.

---

### Configuring BFD Consumer Protocols Using the NX-OS Style CLI

#### Procedure

**Step 1** To enable BFD on the BGP consumer protocol using the NX-OS CLI:

**Example:**

```bash
apic1# configure
apic1(config)# bgp-fabric
apic1(config-bgp-fabric)# asn 200
apic1(config-bgp-fabric)# exit
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 200
apic1(config-bgp)# vrf member tenant t0 vrf v0
apic1(config-leaf-bgp-vrf)# neighbor 1.2.3.4
apic1(config-leaf-bgp-vrf-neighbor)# [no] bfd enable
```
Step 2 To enable BFD on the EIGRP consumer protocol using the NX-OS CLI:

Example:

apic1(config-leaf-if)# [no] ip bfd eigrp enable

Step 3 To enable BFD on the OSPF consumer protocol using the NX-OS CLI:

Example:

apic1(config-leaf-if)# [no] ip ospf bfd enable

apic1(config)# configure
apic1(config)# spine 103
apic1(config-spine)# interface ethernet 5/3.4
apic1(config-spine-if)# [no] ip ospf bfd enable

Step 4 To enable BFD on the Static Route consumer protocol using the NX-OS CLI:

Example:

apic1(config-leaf-vrf)# [no] ip route 10.0.0.1/16 10.0.0.5 bfd

apic1(config)# spine 103
apic1(config-spine)# vrf context tenant infra vrf overlay-1
apic1(config-spine-vrf)# [no] ip route 21.1.1.1/32 32.1.1.1 bfd

Step 5 To enable BFD on IS-IS consumer protocol using the NX-OS CLI:

Example:

apic1(config)# leaf 101
apic1(config-spine)# interface ethernet 1/49
apic1(config-spine-if)# isis bfd enabled
apic1(config-spine-if)# exit
apic1(config-spine)# exit

apic1(config)# spine 103
apic1(config-spine)# interface ethernet 5/2
apic1(config-spine-if)# isis bfd enabled
apic1(config-spine-if)# exit
apic1(config-spine)# exit

### Configuring BFD Consumer Protocols Using the REST API

#### Procedure

Step 1 The following example shows the interface configuration for bidirectional forwarding detection (BFD):

Example:

```
<fvTenant name="ExampleCorp">
  <bfdIfPol name="bfdIfPol" minTxIntvl="400" minRxIntvl="400" detectMult="5" echoRxIntvl="400" echoAdminSt="disabled"/>
```
Step 2

The following example shows the interface configuration for enabling BFD on OSPF and EIGRP:

**Example:**

**BFD on leaf switch**

```xml
<fvTenant name="ExampleCorp">
  <ospfIfPol name="ospf_intf_pol" cost="10" ctrl="bfd"/>
  <eigrpIfPol ctrl="nh-self,split-horizon,bfd" dn="uni/tn-Coke/eigrpIfPol-eigrp_if_default"/>
</fvTenant>
```

**Example:**

**BFD on spine switch**

```xml
<l3extLNodeP name="bSpine">
  <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-103" rtrId="192.3.1.8" />
  <l3extLoopBackIfP addr="10.10.3.1" />  
  <l3extInfraNodeP fabricExtCtrlPeering="false" />
  <l3extRsNodeL3OutAtt tDn="topology/pod-1/paths-103/pathep-[eth5/10]" encap='vlan-4' ifInstT='sub-interface' addr="20.3.10.1/24"/>
  <ospfIfP>
    <ospfRsIfPol tnOspfIfPolName='ospf_intf_pol'/>
  </ospfIfP>
  <bfdIfP name="test" type="sha1" key="hello" status="created,modified">
    <bfdRsIfPol tnBfdIfPolName='default' status="created,modified"/>
  </bfdIfP>
</l3extLNodeP>
```

Step 3

The following example shows the interface configuration for enabling BFD on BGP:

**Example:**

```xml
<fvTenant name="ExampleCorp">
  <l3extOut name="l3-out">
    <l3extLNodeP name="leaf1">
      <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2"/>
      <l3extLIfP name='portIpv4'>
        <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInstT='13-port' addr="10.0.0.1/24" mtu="1500"/>
        <bfdIfP type="sha1" key="password">
          <bfdRsIfPol tnBfdIfPolName='bfdIfPol'/>
        </bfdIfP>
      </l3extLIfP>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>
```
Step 4  The following example shows the interface configuration for enabling BFD on Static Routes:

Example:
BFD on leaf switch

```
<fvTenant name="ExampleCorp">
  <l3extOut name="l3-out">
    <l3extLNodeP name="leaf1">
      <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="2.2.2.2">
        <ipRouteP ip="192.168.3.4" rtCtrl="bfd">
          <ipNexthopP nhAddr="192.168.62.2"/>
        </ipRouteP>
        <l3extLIfP name='portIpv4'>
          <l3extRsPathL3OutAtt ifInstT='l3-port' addr="10.10.10.2/24" mtu="1500" status="created,modified"/>
        </l3extLIfP>
      </l3extRsNodeL3OutAtt>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>
```

Example:
BFD on spine switch

```
<l3extLNodeP name="bSpine">
  <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-103" rtrId="192.3.1.8">
    <ipRouteP ip="0.0.0.0" rtCtrl="bfd">
      <ipNexthopP nhAddr="192.168.62.2"/>
    </ipRouteP>
    <l3extLIfP name='portIf'>
      <l3extRsPathL3OutAtt ifInstT='l3-port' addr="10.10.10.2/24" mtu="1500" status="created,modified"/>
      <l3extLIfP name='portIpv4'>
        <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-103/pathep-[eth1/3]" ifInstT='13-port' addr="10.10.10.2/24" mtu="1500" status="created,modified"/>
      </l3extLIfP>
    </l3extLIfP>
  </l3extRsNodeL3OutAtt>
  <l3extLIfP name='portIf'>
    <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-103/pathep-[eth5/10]" encap='vlan-4' ifInstT='sub-interface' addr="20.3.10.1/24"/>
    <bfdIfP name="test" type="sha1" key="hello" status="created,modified">
      <bfdRsIfPol tnBfdIfPolName='default' status="created,modified"/>
    </bfdIfP>
  </l3extLIfP>
</l3extLNodeP>
```

Step 5  The following example shows the interface configuration for enabling BFD on IS-IS:

```
<13extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/11]" ifInstT='13-port' addr="10.0.0.1/24" mtu="1500">
  <bgpPeerP addr="4.4.4.4/24" allowedSelfAsCnt="3" ctrl="bfd" descr="">
    <bgpRsPeerPfxPol tnBgpPeerPfxPolName=""/>
    <bgpAsP asn="3" descr="" name=""/>
  </bgpPeerP>
</13extRsPathL3OutAtt>
```
OSPF External Routed Networks

Use the procedures in the following sections to configure OSPF external routed networks.

OSPF Layer 3 Outside Connections

OSPF Layer 3 Outside connections can be normal or NSSA areas. The backbone (area 0) area is also supported as an OSPF Layer 3 Outside connection area. ACI supports both OSPFv2 for IPv4 and OSPFv3 for IPv6. When creating an OSPF Layer 3 Outside, it is not necessary to configure the OSPF version. The correct OSPF process is created automatically based on the interface profile configuration (IPv4 or IPv6 addressing). Both
IPv4 and IPv6 protocols are supported on the same interface (dual stack) but it is necessary to create two separate interface profiles.

Layer 3 Outside connections are supported for the routed interfaces, routed sub-interfaces, and SVIs. The SVIs are used when there is a need to share the physical connect for both L2 and L3 traffic. The SVIs are supported on ports, port-channels, and VPC port-channels.

*Figure 15: OSPF Layer3 Out Connections*

When an SVI is used for an Layer 3 Outside connection, an external bridge domain is created on the border leaf switches. The external bridge domain allows connectivity between the two VPC switches across the ACI fabric. This allows both the VPC switches to establish the OSPF adjacencies with each other and the external OSPF device.

When running OSPF over a broadcast network, the time to detect a failed neighbor is the dead time interval (default 40 seconds). Reestablishing the neighbor adjacencies after a failure may also take longer due to designated router (DR) election.

---

**Note**

A link or port-channel failure to one VPC Node does not cause an OSPF adjacency to go down. The OSPF adjacency can stay up via the external BD accessible through the other VPC node.
Creating an OSPF External Routed Network for Management Tenant Using the GUI

- You must verify that the router ID and the logical interface profile IP address are different and do not overlap.
- The following steps are for creating an OSPF external routed network for a management tenant. To create an OSPF external routed network for a tenant, you must choose a tenant and create a VRF for the tenant.
- For more details, see Cisco APIC and Transit Routing.

Procedure

Step 1
On the menubar, choose TENANTS > mgmt.

Step 2
In the Navigation pane, expand Networking > External Routed Networks.

Step 3
Right-click External Routed Networks, and click Create Routed Outside.

Step 4
In the Create Routed Outside dialog box, perform the following actions:
- In the Name field, enter a name (RtdOut).
- Check the OSPF check box.
- In the OSPF Area ID field, enter an area ID.
- In the OSPF Area Control field, check the appropriate check box.
- In the OSPF Area Type field, choose the appropriate area type.
- In the OSPF Area Cost field, choose the appropriate value.
- In the VRF field, from the drop-down list, choose the VRF (inb).
  Note: This step associates the routed outside with the in-band VRF.
- h) From the External Routed Domain drop-down list, choose the appropriate domain.
- i) Click the + icon for Nodes and Interfaces Protocol Profiles area.

Step 5
In the Create Node Profile dialog box, perform the following actions:
- In the Name field, enter a name for the node profile. (borderLeaf).
- In the Nodes field, click the + icon to display the Select Node dialog box.
- In the Node ID field, from the drop-down list, choose the first node. (leaf1).
- In the Router ID field, enter a unique router ID.
- Uncheck the Use Router ID as Loopback Address field.
  Note: By default, the router ID is used as a loopback address. If you want them to be different, uncheck the Use Router ID as Loopback Address check box.
- f) Expand Loopback Addresses, and enter the IP address in the IP field. Click Update, and click OK.
  Enter the desired IPv4 or IPv6 IP address.
- g) In the Nodes field, expand the + icon to display the Select Node dialog box.
  Note: You are adding a second node ID.
- h) In the Node ID field, from the drop-down list, choose the next node. (leaf2).
In the **Router ID** field, enter a unique router ID.

j) Uncheck the **Use Router ID as Loopback Address** field.

---

**Note** By default, the router ID is used as a loopback address. If you want them to be different, uncheck the **Use Router ID as Loopback Address** check box.

k) Expand **Loopback Addresses**, and enter the IP address in the IP field. Click **Update**, and click **OK**. Click **OK**.

Enter the desired IPv4 or IPv6 IP address.

---

**Step 6** In the **Create Node Profile** dialog box, in the **OSPF Interface Profiles** area, click the + icon.

**Step 7** In the **Create Interface Profile** dialog box, perform the following tasks:

a) In the **Name** field, enter the name of the profile (portProf).

b) In the **Interfaces** area, click the **Routed Interfaces** tab, and click the + icon.

c) In the **Select Routed Interfaces** dialog box, in the **Path** field, from the drop-down list, choose the first port (leaf1, port 1/40).

d) In the **IP Address** field, enter an IP address and mask. Click **OK**.

e) In the **Interfaces** area, click the **Routed Interfaces** tab, and click the + icon.

f) In the **Select Routed Interfaces** dialog box, in the **Path** field, from the drop-down list, choose the second port (leaf2, port 1/40).

g) In the **IP Address** field, enter an IP address and mask. Click **OK**.

---

**Note** This IP address should be different from the IP address you entered for leaf1 earlier.

---

h) In the **Create Interface Profile** dialog box, click **OK**.

The interfaces are configured along with the OSPF interface.

---

**Step 8** In the **Create Node Profile** dialog box, click **OK**.

**Step 9** In the **Create Routed Outside** dialog box, click **Next**.

The **Step 2 External EPG Networks** area is displayed.

**Step 10** In the **External EPG Networks** area, click the + icon.

**Step 11** In the **Create External Network** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the external network (extMgmt).

b) Expand **Subnet** and in the **Create Subnet** dialog box, in the **IP address** field, enter an IP address and mask for the subnet.

c) In the **Scope** field, check the desired check boxes. Click **OK**.

d) In the **Create External Network** dialog box, click **OK**.

e) In the **Create Routed Outside** dialog box, click **Finish**.

---

**Note** In the **Work** pane, in the **External Routed Networks** area, the external routed network icon (RtdOut) is now displayed.

---

**Creating an OSPF External Routed Network for a Tenant Using the NX-OS CLI**

Configuring external routed network connectivity involves the following steps:

1. Create a VRF under Tenant.
2. Configure L3 networking configuration for the VRF on the border leaf switches, which are connected to the external routed network. This configuration includes interfaces, routing protocols (BGP, OSPF, EIGRP), protocol parameters, route-maps.

3. Configure policies by creating external-L3 EPGs under tenant and deploy these EPGs on the border leaf switches. External routed subnets on a VRF which share the same policy within the ACI fabric form one "External L3 EPG" or one "prefix EPG".

Configuration is realized in two modes:

- Tenant mode: VRF creation and external-L3 EPG configuration
- Leaf mode: L3 networking configuration and external-L3 EPG deployment

The following steps are for creating an OSPF external routed network for a tenant. To create an OSPF external routed network for a tenant, you must choose a tenant and then create a VRF for the tenant.

---

**Note**

The examples in this section show how to provide external routed connectivity to the "web" epg in the "OnlineStore" application for tenant "exampleCorp".

---

**Procedure**

**Step 1** Configure the VLAN domain.

*Example:*

```plaintext
api1(config)# vlan-domain dom_exampleCorp
api1(config-vlan)# vlan 5-1000
api1(config-vlan)# exit
```

**Step 2** Configure the tenant VRF and enable policy enforcement on the VRF.

*Example:*

```plaintext
api1(config)# tenant exampleCorp
api1(config-tenant)# vrf context exampleCorp_v1
api1(config-tenant-vrf)# contract enforce
api1(config-tenant-vrf)# exit
```

**Step 3** Configure the tenant BD and mark the gateway IP as “public”. The entry "scope public" makes this gateway address available for advertisement through the routing protocol for external-L3 network.

*Example:*

```plaintext
api1(config-tenant)# bridge-domain exampleCorp_b1
api1(config-tenant-bd)# vrf member exampleCorp_v1
api1(config-tenant-bd)# exit
api1(config-tenant)# interface bridge-domain exampleCorp_b1
api1(config-tenant-interface)# ip address 172.1.1.1/24 scope public
api1(config-tenant-interface)# exit
```

**Step 4** Configure the VRF on a leaf.

*Example:*

```plaintext
```
Step 5  Configure the OSPF area and add the route map.

Example:

```
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant exampleCorp vrf exampleCorp_v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.1 route-map map100 out
apic1(config-leaf-ospf-vrf)# exit
```

Step 6  Assign the VRF to the interface (sub-interface in this example) and enable the OSPF area.

Example:

```
Note  For the sub-interface configuration, the main interface (ethernet 1/11 in this example) must be converted to an L3 port through “no switchport” and assigned a vlan-domain (dom_exampleCorp in this example) that contains the encapsulation VLAN used by the sub-interface. In the sub-interface ethernet1/11.500, 500 is the encapsulation VLAN.
```

```
apic1(config-leaf)# interface ethernet 1/11
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member dom_exampleCorp
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/11.500
apic1(config-leaf-if)# vrf member tenant exampleCorp vrf exampleCorp_v1
apic1(config-leaf-if)# ip address 157.10.1.1/24
apic1(config-leaf-if)# ip router ospf default area 0.0.0.1
```

Step 7  Configure the external-L3 EPG policy. This includes the subnet to match for identifying the external subnet and consuming the contract to connect with the epg “web”.

Example:

```
apic1(config)# tenant t100
apic1(config-tenant)# external-l3 epg l3epg100
apic1(config-tenant-l3ext-epg)# vrf member v100
apic1(config-tenant-l3ext-epg)# match ip 145.10.1.0/24
apic1(config-tenant-l3ext-epg)# contract consumer web
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)#exit
```

Step 8  Deploy the external-L3 EPG on the leaf switch.

Example:

```
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t100 vrf v100
apic1(config-leaf-vrf)# external-l3 epg l3epg100
```
Creating OSPF External Routed Network for Management Tenant Using REST API

- You must verify that the router ID and the logical interface profile IP address are different and do not overlap.
- The following steps are for creating an OSPF external routed network for a management tenant. To create an OSPF external routed network for a tenant, you must choose a tenant and create a VRF for the tenant.
- For more details, see *Cisco APIC and Transit Routing*.

**Procedure**

Create an OSPF external routed network for management tenant.

**Example:**

POST: https://apic-ip-address/api/mo/uni/tn-mgmt.xml

```
<fvTenant name="mgmt">
  <fvBD name="bd1">
    <fvRsBDToOut tnL3extOutName="RtdOut" />
    <fvSubnet ip="1.1.1.1/16" />
    <fvSubnet ip="1.2.1.1/16" />
    <fvSubnet ip="40.1.1.1/24" scope="public" />
    <fvRsCtx tnFvCtxName="inb" />
  </fvBD>
  <fvCtx name="inb" />
  <l3extOut name="RtdOut">
    <l3extInstP name="extMgmt">
      <l3extInstP>
        <l3extNodeP name="borderLeaf">
          <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="10.10.10.10"/>
          <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-102" rtrId="10.10.10.11"/>
          <l3extLIfP name="portProfile">
            <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/40]" ifInstT='13-port' addr="192.168.62.1/24"/>
            <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-102/pathep-[eth1/40]" ifInstT='13-port' addr="192.168.62.5/24"/>
          </l3extLIfP>
        </l3extNodeP>
        <ospfIfP>
          <l3extLIfP>
            <l3extRsEctx tnFvCtxName="inb"/>
            <ospfExtCtx areaId="57"/>
          </l3extLIfP>
        </ospfIfP>
      </l3extInstP>
    </l3extInstP>
    <l3extLNodeP>
      <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/40]" ifInstT='13-port' addr="192.168.62.1/24"/>
      <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-102/pathep-[eth1/40]" ifInstT='13-port' addr="192.168.62.5/24"/>
    </l3extLNodeP>
  </l3extOut>
</fvTenant>
```

**EIGRP External Routed Networks**

Use the procedures in the following sections to configure EIGRP external routed networks.
About EIGRP Layer 3 Outside Connections

This example shows how to configure Enhanced Interior Gateway Routing Protocol (EIGRP) when using the Cisco APIC. The following information applies when configuring EIGRP:

- The tenant, VRF, and bridge domain must already be created.
- The Layer 3 outside tenant network must already be configured.
- The route control profile under routed outside must already be configured.
- The EIGRP VRF policy is the same as the EIGRP family context policy.
- EIGRP supports only export route control profile. The configuration related to route controls is common across all the protocols.

You can configure EIGRP to perform automatic summarization of subnet routes (route summarization) into network-level routes. For example, you can configure subnet 131.108.1.0 to be advertised as 131.108.0.0 over interfaces that have subnets of 192.31.7.0 configured. Automatic summarization is performed when there are two or more network router configuration commands configured for the EIGRP process. By default, this feature is enabled. For more information, see Route Summarization.

EIGRP Protocol Support

EIGRP protocol is modeled similar to other routing protocols in the Cisco Application Centric Infrastructure (ACI) fabric.

**Supported Features**

The following features are supported:

- IPv4 and IPv6 routing
- Virtual routing and forwarding (VRF) and interface controls for each address family
- Redistribution with OSPF across nodes
- Default route leak policy per VRF
- Passive interface and split horizon support
- Route map control for setting tag for exported routes
- Bandwidth and delay configuration options in an EIGRP interface policy
- Authentication support

**Unsupported Features**

The following features are not supported:

- Stub routing
- EIGRP used for BGP connectivity
- Multiple EIGRP L3extOut$ on the same node
- Summary prefix
• Per interface distribute lists for import and export

Categories of EIGRP Functions

EIGRP functions can be broadly categorized as follows:

• Protocol policies
• L3extOut configurations
• Interface configurations
• Route map support
• Default route support
• Transit support

Primary Managed Objects That Support EIGRP

The following primary managed objects provide EIGRP support:

• **EIGRP Address Family Context Policy** eigrpCtxAfPol: Address Family Context policy configured under fvTenant (Tenant/Protocols).
  • fvRsCtxToEigrpCtxAfPol: Relation from a VRF to a eigrpCtxAfPol for a given address family (IPv4 or IPv6). There can be only one relation for each address family.
  • eigrpIfPol: EIGRP Interface policy configured in fvTenant.
  • eigrpExtP: Enable flag for EIGRP in an L3extOut.
  • eigrpIfP: EIGRP interface profile attached to an l3extL1P.
  • eigrpRsIfPol: Relation from EIGRP interface profile to an eigrpIfPol.
  • Defrtleak: Default route leak policy under an L3extOut.

EIGRP Protocol Policies Supported Under a Tenant

The following EIGRP protocol policies are supported under a tenant:

• **EIGRP Interface policy** (eigrpIfPol)—contains the configuration that is applied for a given address family on an interface. The following configurations are allowed in the interface policy:

  • **Hello interval** in seconds
  • **Hold interval** in seconds
  • One or more of the following interface control flags:
    • **split horizon**
    • **passive**
    • **next hop self**
• **EIGRP Address Family Context Policy** (*eigrpCtxAfPol*)—contains the configuration for a given address family in a given VRF. An *eigrpCtxAfPol* is configured under tenant protocol policies and can be applied to one or more VRFs under the tenant. An *eigrpCtxAfPol* can be enabled on a VRF through a relation in the VRF-per-address family. If there is no relation to a given address family, or the specified *eigrpCtxAfPol* in the relation does not exist, then the default VRF policy created under the common tenant is used for that address family.

The following configurations are allowed in the *eigrpCtxAfPol*:

- Administrative distance for internal route
- Administrative distance for external route
- Maximum ECMP paths allowed
- Active timer interval
- Metric version (32-bit / 64-bit metrics)

**Guidelines and Limitations When Configuring EIGRP**

When configuring EIGRP, follow these guidelines:

- Configuring EIGRP and BGP for the same Layer 3 outside is not supported.
- Configuring EIGRP and OSPF for the same Layer 3 outside is not supported.
- There can be one EIGRP Layer 3 Out per node per VRF. If multiple VRFs are deployed on a node, each VRF can have its own Layer 3 Out.
- Multiple EIGRP peers from a single Layer 3 Out is supported. This enables you to connect to multiple EIGRP devices from the same node with a single Layer 3 Out.

**Configuring EIGRP Using the GUI**

**Procedure**

**Step 1** On the menu bar, choose **Tenants > All Tenants**.

**Step 2** In the **Work** pane, double click a tenant.

**Step 3** In the **Navigation** pane, expand the **Tenant_name > Networking > Protocol Policies > EIGRP**.

**Step 4** Right-click **EIGRP Address Family Context** and choose **Create EIGRP Address Family Context Policy**.

**Step 5** In the **Create EIGRP Address Family Context Policy** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the context policy.

b) In the **Active Interval (min)** field, choose an interval timer.

c) In the **External Distance** and the **Internal Distance** fields, choose the appropriate values.

d) In the **Maximum Path Limit** field, choose the appropriate load balancing value between interfaces (per node/per leaf switch).

e) In the **Metric Style** field, choose the appropriate metric style. Click **Submit**.

In the **Work** pane, the context policy details are displayed.
Step 6 To apply the context policy on a VRF, in the Navigation pane, expand Networking > VRFs.

Step 7 Choose the appropriate VRF, and in the Work pane under Properties, expand EIGRP Context Per Address Family.

Step 8 In the EIGRP Address Family Type drop-down list, choose an IP version.

Step 9 In the EIGRP Address Family Context drop-down list, choose the context policy. Click Update, and Click Submit.

Step 10 To enable EIGRP within the Layer 3 Out, in the Navigation pane, click Networking > External Routed Networks, and click the desired Layer 3 outside network.

Step 11 In the Work pane under Properties, check the checkbox for EIGRP, and enter the EIGRP Autonomous System number. Click Submit.

Step 12 To create an EIGRP interface policy, in the Navigation pane, click Networking > Protocol Policies > EIGRP Interface and perform the following actions:
   a) Right-click EIGRP Interface, and click Create EIGRP Interface Policy.
   b) In the Create EIGRP Interface Policy dialog box, in the Name field, enter a name for the policy.
   c) In the Control State field, check the desired checkboxes to enable one or multiple controls.
   d) In the Hello Interval (sec) field, choose the desired interval.
   e) In the Hold Interval (sec) field, choose the desired interval. Click Submit.
   f) In the Bandwidth field, choose the desired bandwidth.
   g) In the Delay field, choose the desired delay in tens of microseconds or pico seconds.

In the Work pane, the details for the EIGRP interface policy are displayed.

Step 13 In the Navigation pane, click the appropriate external routed network where EIGRP was enabled, expand Logical Node Profiles and perform the following actions:
   a) Expand an appropriate node and an interface under that node.
   b) Right-click the interface and click Create EIGRP Interface Profile.
   c) In the Create EIGRP Interface Profile dialog box, in the EIGRP Policy field, choose the desired EIGRP interface policy. Click Submit.

Note The EIGRP VRF policy and EIGRP interface policies define the properties used when EIGRP is enabled. EIGRP VRF policy and EIGRP interface policies are also available as default policies if the user does not want to create new policies. So, if a user does not explicitly choose either one of the policies, the default policy is automatically utilized when EIGRP is enabled.

This completes the EIGRP configuration.

---

**Configuring EIGRP Using the NX-OS-Style CLI**

**Procedure**

**Step 1** SSH to an Application Policy Infrastructure Controller (APIC) in the fabric:

Example:

```
# ssh admin@node_name
```

**Step 2** Enter the configure mode:
Example:

apic1# configure

Step 3
Enter the configure mode for a tenant:

Example:

apic1(config)# tenant tenant1

Step 4
Configure the Layer 3 Outside on the tenant:

Example:

apic1(config-tenant)# show run
# Command: show running-config tenant tenant1
# Time: Tue Feb 16 09:44:09 2016
tenant tenant1
  vrf context l3out
    exit
l3out l3out-L1
  vrf member l3out
    exit
l3out l3out-L3
  vrf member l3out
    exit
external-l3 epg tenant1 l3out l3out-L3
  vrf member l3out
  match ip 0.0.0.0/0
  match ip 3.100.0.0/16
  match ipv6 43:101::/48
  contract consumer default
    exit
external-l3 epg tenant1 l3out l3out-L1
  vrf member l3out
  match ipv6 23:101::/48
  match ipv6 13:101::/48
  contract provider default
    exit
exit

Step 5
Configure a VRF for EIGRP on a leaf:

Example:

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant tenant1 vrf l3out l3out l3out-L1
apic1(config-leaf-vrf)# show run
# Command: show running-config leaf 101 vrf context tenant tenant1 vrf l3out l3out l3out-L1
# Time: Tue Feb 16 09:44:45 2016
leaf 101
  vrf context tenant tenant1 vrf l3out l3out l3out-L1
    router-id 3.1.1.1
    route-map l3out-L1_in
      scope global
      ip prefix-list tenant1 permit 1:102::/48
      match prefix-list tenant1
        exit
    exit
route-map l3out-L1_out
  scope global
  ip prefix-list tenant1 permit 3.102.10.0/23
  ip prefix-list tenant1 permit 3.102.100.0/31
  ip prefix-list tenant1 permit 3.102.20.0/24
  ip prefix-list tenant1 permit 3.102.30.0/25
  ip prefix-list tenant1 permit 3.102.40.0/26

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
ip prefix-list tenant1 permit 3.102.50.0/27
ip prefix-list tenant1 permit 3.102.60.0/28
ip prefix-list tenant1 permit 3.102.70.0/29
ip prefix-list tenant1 permit 3.102.80.0/30
ip prefix-list tenant1 permit 3.102.90.0/32
<OUTPUT TRUNCATED>
ip prefix-list tenant1 permit ::/0
match prefix-list tenant1
exit
exit
route-map l3out-L1_shared
scope global
exit
exit

Step 6  Configure the EIGRP interface policy:

Example:
apic1(config-leaf)# template eigrp interface-policy tenant1 tenant tenant1
This template will be available on all leaves where tenant tenant1 has a VRF deployment

apic1(config-template-eigrp-if-pol)# show run
# Command: show running-config leaf 101 template eigrp interface-policy tenant1 tenant

# Time: Tue Feb 16 09:45:50 2016
leaf 101
template eigrp interface-policy tenant1 tenant tenant1
  ip hello-interval eigrp default 10
  ip hold-interval eigrp default 30
  ip throughput-delay eigrp default 20 tens-of-micro
  ip bandwidth eigrp default 20
exit
exit

Step 7  Configure the EIGRP VRF policy:

Example:
apic1(config-leaf)# template eigrp vrf-policy tenant1 tenant tenant1
This template will be available on all leaves where tenant tenant1 has a VRF deployment

apic1(config-template-eigrp-vrf-pol)# show run
# Command: show running-config leaf 101 template eigrp vrf-policy tenant1 tenant

# Time: Tue Feb 16 09:46:31 2016
leaf 101
template eigrp vrf-policy tenant1 tenant tenant1
  metric version 64bit
exit
exit

Step 8  Configure the EIGRP VLAN interface and enable EIGRP in the interface:

Example:
apic1(config-leaf)# interface vlan 1013
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface vlan 1013

# Time: Tue Feb 16 09:46:59 2016
leaf 101
  interface vlan 1013
    vrf member tenant tenant1 vrf l3out
    ip address 101.13.1.2/24
    ip router eigrp default
    ipv6 address 101:13::1:2/112 preferred
    ipv6 router eigrp default
    ipv6 link-local fe80::101:13:1:2
Routing Protocol Support

Configuring EIGRP Using the NX-OS-Style CLI

inherit eigrp ip interface-policy tenant1
inherit eigrp ipv6 interface-policy tenant1
exit
exit
apic1(config-leaf-if)# ip summary-address ?
eigrp Configure route summarization for EIGRP
apic1(config-leaf-if)# ip summary-address eigrp default 11.11.0.0/16 ?
<CR>
apic1(config-leaf-if)# ip summary-address eigrp default 11.11.0.0/16
apic1(config-leaf-if)# ip summary-address eigrp default 11:11:1::/48
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface vlan 1013
# Time: Tue Feb 16 09:47:34 2016
leaf 101
  interface vlan 1013
    vrf member tenant tenant1 vrf l3out
    ip address 101.13.1.2/24
    ip router eigrp default
    ip summary-address eigrp default 11.11.0.0/16
    ip summary-address eigrp default 11:11:1::/48
    ipv6 address 101:13::1:2/112 preferred
    ipv6 router eigrp default
    ipv6 link-local fe80::101:13:1:2
    inherit eigrp ip interface-policy tenant1
    inherit eigrp ipv6 interface-policy tenant1
  exit
exit

Step 9

Apply the VLAN on the physical interface:

Example:

apic1(config-leaf)# interface ethernet 1/5
apic1(config-leaf-if)# show run
# Command: show running-config leaf 101 interface ethernet 1 / 5
# Time: Tue Feb 16 09:48:05 2016
leaf 101
  interface ethernet 1/5
    vlan-domain member cli
    switchport trunk allowed vlan 1213 tenant tenant13 external-svi l3out l3out-L1
    switchport trunk allowed vlan 1613 tenant tenant17 external-svi l3out l3out-L1
    switchport trunk allowed vlan 1013 tenant tenant1 external-svi l3out-L1
    switchport trunk allowed vlan 666 tenant ten_v6_c1i external-svi l3out_c1i_L1
    switchport trunk allowed vlan 1513 tenant tenant16 external-svi l3out-L1
    switchport trunk allowed vlan 1313 tenant tenant14 external-svi l3out-L1
    switchport trunk allowed vlan 1413 tenant tenant15 external-svi l3out-L1
    switchport trunk allowed vlan 1113 tenant tenant12 external-svi l3out-L1
    switchport trunk allowed vlan 712 tenant mgmt external-svi l3out inband_L1
    switchport trunk allowed vlan 1913 tenant tenant10 external-svi l3out l3out-L1
    switchport trunk allowed vlan 300 tenant tenant1 external-svi l3out-L1
  exit

Step 10

Enable router EIGRP:

Example:

apic1(config-eigrp-vrf)# show run
# Command: show running-config leaf 101 router eigrp default vrf member tenant tenant1 vrf l3out
# Time: Tue Feb 16 09:49:05 2016
leaf 101
  router eigrp default
  exit
  router eigrp default

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Configuring EIGRP Using the REST API

Procedure

Step 1 Configure an EIGRP context policy.

Example:

```xml
<polUni>
  <fvTenant name="cisco_6">
    <eigrpCtxAfPol actIntvl="3" descr="" dn="uni/tn-cisco_6/eigrpCtxAfP-eigrp_default_pol"
      extDist="170"
      intDist="90" maxPaths="8" metricStyle="narrow" name="eigrp_default_pol" ownerKey=""
      ownerTag="/">
      <eigrpCtxAfPol actIntvl="3" descr="" dn="uni/tn-cisco_6/eigrpCtxAfP-eigrp_default_pol"
        extDist="170"
        intDist="90" maxPaths="8" metricStyle="narrow" name="eigrp_default_pol" ownerKey=""
        ownerTag="/">
    </fvTenant>
</polUni>
```

Step 2 Configure an EIGRP interface policy.

Example:

```xml
<polUni>
  <fvTenant name="cisco_6">
    <eigrpIfPol bw="10" ctrl="nh-self,split-horizon" delay="10" delayUnit="tens-of-micro"
      descr="" dn="uni/tn-cisco_6/eigrpIfPol-eigrp_if_default"
      helloIntvl="5" holdIntvl="15" name="eigrp_if_default" ownerKey="" ownerTag="/">
    </fvTenant>
</polUni>
```

Step 3 Configure an EIGRP VRF.

Example:

IPv4:

```xml
<polUni>
  <fvTenant name="cisco_6">
    <fvCtx name="dev">
      <fvRsCtxToEigrpCtxAfPol tnEigrpCtxAfPolName="eigrp_ctx_pol_v4" af="1" />
    </fvCtx>
  </fvTenant>
</polUni>
```

IPv6:
Step 4  Configure an EIGRP Layer3 Outside.

Example:

IPv4

```xml
<polUni>
  <fvTenant name="cisco_6">
    <fvCtxt name="dev">
      <fvRsCtxToEigrpCtxAfPol tnEigrpCtxAfPolName="eigrp_ctx_pol_v6" af="ipv6-ucast"/>
    </fvCtxt>
  </fvTenant>
</polUni>
```

IPv6

```xml
<polUni>
  <fvTenant name="cisco_6">
    <fvCtxt name="dev">
      <fvRsCtxToEigrpCtxAfPol tnEigrpCtxAfPolName="eigrp_ctx_pol_v6" af="ipv6-ucast"/>
    </fvCtxt>
  </fvTenant>
</polUni>
```

IPv4 and IPv6

```xml
<polUni>
  <fvTenant name="cisco_6">
    <fvCtxt name="dev">
      <fvRsCtxToEigrpCtxAfPol tnEigrpCtxAfPolName="eigrp_ctx_pol_v6" af="ipv6-ucast"/>
    </fvCtxt>
  </fvTenant>
</polUni>
```
Step 5  (Optional) Configure the interface policy knobs.

Example:

```xml
<polUni>
  <fvTenant name="cisco_6">
    <eigrpIfPol bw="1000000" ctrl="nh-self,split-horizon" delay="10"
               delayUnit="tens-of-micro" helloIntvl="5" holdIntvl="15" name="default"/>
  </fvTenant>
</polUni>
```

The bandwidth (bw) attribute is defined in Kbps. The delayUnit attribute can be "tens of micro" or "pico".
Route Summarization

This chapter contains the following sections:

- Route Summarization, on page 119
- Configuring Route Summarization for BGP, OSPF, and EIGRP Using the REST API, on page 119
- Configuring Route Summarization for BGP, OSPF, and EIGRP Using the NX-OS Style CLI, on page 121
- Configuring Route Summarization for BGP, OSPF, and EIGRP Using the GUI, on page 122

Route Summarization

Route summarization simplifies route tables by replacing many specific addresses with a single address. For example, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 can be replaced with 10.1.0.0/16. Route summarization policies enable routes to be shared efficiently among border leaf switches and their neighboring leaf switches. BGP, OSPF, or EIGRP route summarization policies are applied to a bridge domain or transit subnet. For OSPF, inter-area and external route summarization are supported. Summary routes are exported; they are not advertised within the fabric.

Configuring Route Summarization for BGP, OSPF, and EIGRP Using the REST API

Procedure

Step 1  Configure BGP route summarization using the REST API as follows:

Example:

```
<vTenant name="common">
  <vCtx name="vrf1">
    <bgpRtSummPol name="bgp_rt_summ" cntrl='as-set'/>
    <l3extOut name="l3_ext_pol" >>
      <l3extLNodeP name="bLeaf">
        <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="20.10.1.1"/>
        <l3extLIfP name='portIf'>
          <l3extRsPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/31]"
```
Step 2  Configure OSPF inter-area and external summarization using the following REST API:

Example:

```xml
<?xml version="1.0" encoding="utf-8"?>
<fvTenant name="t20">
  <!-- Ospf Inter External route summarization Policy-->
  <ospfRtSummPol cost="unspecified" interAreaEnabled="no" name="ospfext"/>
  <!-- Ospf Inter Area route summarization Policy-->
  <ospfRtSummPol cost="16777215" interAreaEnabled="yes" name="interArea"/>
  <fvCtx name="ctx0" pcEnfDir="ingress" pcEnfPref="enforced"/>
  <!-- L3OUT backbone Area-->
  <l3extOut enforceRtctrl="export" name="l3_1" ownerKey="" ownerTag="" targetDscp="unspecified">
    <l3extRsEctx tnFvCtxName="ctx0"/>
    <l3extLNodeP name="node-101">
      <l3extRsNodeL3OutAttr trId="20.1.3.2" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
      <l3extLIfP name="intf-1">
        <l3extRsPathL3OutAtt addr="20.1.5.2/24" encap="vlan-1001" ifInstT="sub-interface" tDn="topology/pod-1/paths-101/pathep-[eth1/33]"/>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP name="l3InstP1">
      <fvRsProv tnVzBrCPName="default"/>
      <!-- Ospf External Area route summarization-->
      <l3extSubnet aggregate="" ip="193.0.0.0/8" name="" scope="export-rtctrl">
        <l3extRsSubnetToRtSumm tDn="uni/tn-t20/ospfrtsumm-ospfext"/>
      </l3extSubnet>
    </l3extInstP>
    <l3extInstP>
      <ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="backbone" areaType="regular"/>
    </l3extInstP>
  </l3extOut>
  <!-- L3OUT Regular Area-->
  <l3extOut enforceRtctrl="export" name="l3_2">
    <l3extRsEctx tnFvCtxName="ctx0"/>
    <l3extLNodeP name="node-101">
      <l3extRsNodeL3OutAttr trId="20.1.3.2" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
      <l3extLIfP name="intf-2">
        <l3extRsPathL3OutAtt addr="20.1.2.2/24" encap="vlan-1014" ifInstT="sub-interface" tDn="topology/pod-1/paths-101/pathep-[eth1/11]"/>
      </l3extLIfP>
    </l3extLNodeP>
    <l3extInstP matchT="AtleastOne" name="l3InstP2">
      <fvRsCons tnVzBrCPName="default"/>
      <!-- Ospf Inter Area route summarization-->
      <l3extSubnet aggregate="" ip="197.0.0.0/8" name="" scope="export-rtctrl">
        <l3extRsSubnetToRtSumm tDn="uni/tn-t20/ospfrtsumm-interArea"/>
      </l3extSubnet>
    </l3extInstP>
  </l3extOut>
</fvTenant>
```
Step 3 Configure EIGRP summarization using the following REST API:

Example:

```xml
<fvTenant name="exampleCorp">
  <l3extOut name="out1">
    <l3extInstP name="eigrpSummInstp">
      <l3extSubnet aggregate="" descr="" ip="197.0.0.0/8" name="" scope="export-rtctrl">
        <l3extRsSubnetToRtSumm/>
      </l3extSubnet>
    </l3extInstP>
  </l3extOut>
</fvTenant>
```

Note: There is no route summarization policy to be configured for EIGRP. The only configuration needed for enabling EIGRP summarization is the summary subnet under the InstP.

---

### Configuring Route Summarization for BGP, OSPF, and EIGRP Using the NX-OS Style CLI

**Procedure**

**Step 1** Configure BGP route summarization using the NX-OS CLI as follows:

a) Enable BGP as follows:

Example:

```
apic1(config)# pod 1
apic1(config-pod)# bgp fabric
apic1(config-pod-bgp)# asn 10
apic1(config-pod)# exit
apic1(config)# leaf 101
apic1(config-leaf)# router bgp 10
```

b) Configure the summary route as follows:

Example:

```
apic1(config-bgp)# vrf member tenant common vrf vrf1
apic1(config-leaf-bgp-vrf)# aggregate-address 10.0.0.0/8
```

**Step 2** Configure OSPF external summarization using the NX-OS CLI as follows:

Example:
Step 3 Configure OSPF inter-area summarization using the NX-OS CLI as follows:

apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant common vrf vrf1
apic1(config-leaf-ospf-vrf)# summary-address 10.0.0.0/8

Step 4 Configure EIGRP summarization using the NX-OS CLI as follows:

Example:

apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/31 (Or interface vlan <vlan-id>)
apic1(config-leaf-if)# ip summary-address eigrp default 10.0.0.0/8

Note There is no route summarization policy to be configured for EIGRP. The only configuration needed for enabling EIGRP summarization is the summary subnet under the InstP.

Configuring Route Summarization for BGP, OSPF, and EIGRP Using the GUI

Before you begin

For each of the following configurations, you have already created an L3 Out. For the L3 Out, you can create external routed networks, subnets, and route summarization polices.

Procedure

Step 1 Configure BGP route summarization using the GUI as follows:

a) On the menu bar, choose Tenants > common
b) In the Navigation pane, expand Networking > External Routed Networks.
c) Right-click on External Routed Networks, then select Create Routed Outside.
   The Create Routed Outside dialog box appears.
d) In the work pane, check the check box next to BGP.
e) Enter a name in the Name field, then click NEXT.
   The External EPG Networks dialog box appears.
f) In the work pane, click the + sign.
   The Define an External Network dialog box appears.
g) Enter a name in the Name field, then click the + sign above Route Summarization Policy.
   The Create Subnet dialog box appears.
h) In the Specify the Subnet dialog box, you can associate a route summarization policy to the subnet as follows:
Example:

- Enter an IP address in the **IP Address** field.
- Check the check box next to **Export Route Control Subnet**.
- Check the check box next to **External Subnets for the External EPG**.
- From the **BGP Route Summarization Policy** drop-down menu, select either **default** for an existing (default) policy or **Create BGP route summarization policy** to create a new policy.
- If you selected **Create BGP route summarization policy**, the **Create BGP Route Summarization Policy** dialog box appears. Enter a name for it in the **Name** field, check the **Control State** check box for **Generate AS-SET information**, click **SUBMIT**, click **OK**, click **OK**, click **FINISH**.

**Step 2** Configure OSPF inter-area and external summarization using GUI as follows:

a) On the menu bar, choose **Tenants > common**
b) In the Navigation pane, expand **Networking > External Routed Networks > Networks**
c) In the work pane, click the + sign above **Route Summarization Policy**.
The **Create Subnet** dialog box appears.
d) In the **Specify the Subnet** dialog box, you can associate a route summarization policy to the subnet as follows:

Example:

- Enter an IP address in the **IP Address** field.
- Check the check box next to **Export Route Control Subnet**.
- Check the check box next to **External Subnets for the External EPG**.
- From the **OSPF Route Summarization Policy** drop-down menu, choose either **default** for an existing (default) policy or **Create OSPF route summarization policy** to create a new policy.
- If you chose **Create OSPF route summarization policy**, the **Create OSPF Route Summarization Policy** dialog box appears. Enter a name for it in the **Name** field, check the check box next to **Inter-Area Enabled**, enter a value next to **Cost**, click **SUBMIT**.

**Step 3** Configure EIGRP summarization using the GUI as follows:

a) On the menu bar, choose **Tenants > common**
b) In the Navigation pane, expand **Networking > External Routed Networks**.
c) Right-click on **External Routed Networks**, choose **Create Routed Outside**.
The **Create Routed Outside** dialog box appears.
d) In the work pane, check the check box next to **EIGRP**.
e) Enter a name in the **Name** field, click **NEXT**.
The **External EPG Networks** dialog box appears.
f) In the work pane, click the + sign.
The **Define an External Network** dialog box appears.
g) Enter a name in the **Name** field, then click the + sign above **Route Summarization Policy**.
The **Create Subnet** dialog box appears.
h) In the **Specify the Subnet** dialog box, you can associate a route summarization policy to the subnet as follows:

Example:
• Enter an IP address in the **IP Address** field.
• Check the check box next to **Export Route Control Subnet**.
• Check the check box next to **External Subnets for the External EPG**.
• Check the check box next to **EIGRP Route Summarization**, click **OK**, click **OK**, click **FINISH**.
Route Control

This chapter contains the following sections:

- Route Maps/Profiles with Explicit Prefix Lists, on page 125
- Routing Control Protocols, on page 138

Route Maps/Profiles with Explicit Prefix Lists

About Route Map/Profile

The route profile is a logical policy that defines an ordered set (rtctrlCtxP) of logical match action rules with associated set action rules. The route profile is the logical abstract of a route map. Multiple route profiles can be merged into a single route map. A route profile can be one of the following types:

- Match Prefix and Routing Policy: Pervasive subnets (fvSubnet) and external subnets (l3extSubnet) are combined with a route profile and merged into a single route map (or route map entry). Match Prefix and Routing Policy is the default value.

- Match Routing Policy Only: The route profile is the only source of information to generate a route map, and it will overwrite other policy attributes.

Note

When explicit prefix list is used, the type of the route profile should be set to "match routing policy only".

After the match and set profiles are defined, the route map must be created in the Layer 3 Out. Route maps can be created using one of the following methods:

- Create a "default-export" route map for export route control, and a "default-import" route map for import route control.

- Create other route maps (not named default-export or default-import) and setup the relation from one or more l3extInstPs or subnets under the l3extInstP.

- In either case, match the route map on explicit prefix list by pointing to the rtctrlSubjP within the route map.
In the export and import route map, the set and match rules are grouped together along with the relative sequence across the groups (rtctrlCtxP). Additionally, under each group of match and set statements (rtctrlCtxP) the relation to one or more match profiles are available (rtctrlSubjP).

Any protocol enabled on Layer 3 Out (for example BGP protocol), will use the export and import route map for route filtering.

About Explicit Prefix List Support for Route Maps/Profile

In Cisco APIC, for public bridge domain (BD) subnets and external transit networks, inbound and outbound route controls are provided through an explicit prefix list. Inbound and outbound route control for Layer 3 Out is managed by the route map/profile (rtctrlProfile). The route map/profile policy supports a fully controllable prefix list for Layer 3 Out in the Cisco ACI fabric.

The subnets in the prefix list can represent the bridge domain public subnets or external networks. Explicit prefix list presents an alternate method and can be used instead of the following:

• Advertising BD subnets through BD to Layer 3 Out relation.

Note: The subnet in the BD must be marked public for the subnet to be advertised out.

• Specifying a subnet in the l3extInstP with export/import route control for advertising transit and external networks.

Explicit prefix list is defined through a new match type that is called match route destination (rtctrlMatchRtDest). An example usage is provided in the API example that follows.
Additional information about match rules, set rules when using explicit prefix list are as follows:

**Match Rules**

- Under the tenant (fvTenant), you can create match profiles (rtctrlSubjP) for route map filtering. Each match profile can contain one or more match rules. Match rule supports multiple match types. Prior to Cisco APIC release 2.1(x), match types supported were explicit prefix list and community list.

Starting with Cisco APIC release 2.1(x), explicit prefix match or match route destination (rtctrlMatchRtDest) is supported.

Match prefix list (rtctrlMatchRtDest) supports one or more subnets with an optional aggregate flag. Aggregate flags are used for allowing prefix matches with multiple masks starting with the mask mentioned in the configuration till the maximum mask allowed for the address family of the prefix. This is the equivalent of the "le" option in the prefix-list in NX-OS software (example, 10.0.0.0/8 le 32).

The prefix list can be used for covering the following cases:

- Allow all (0.0.0.0/0 with aggregate flag, equivalent of 0.0.0.0/0 le 32)
- One or more of specific prefixes (example: 10.1.1.0/24)
- One or more of prefixes with aggregate flag (example, equivalent of 10.1.1.0/24 le 32).
When a route map with a match prefix “0.0.0.0/0 with aggregate flag” is used under an L3Out EPG in the export direction, the rule is applied only for redistribution from dynamic routing protocols. Therefore, the rule is not applied to the following (in routing protocol such as OSPF or EIGRP):

- Bridge domain (BD) subnets
- Directly connected subnets on the border leaf switch
- Static routes defined on the L3Out

The explicit prefix match rules can contain one or more subnets, and these subnets can be bridge domain public subnets or external networks. Subnets can also be aggregated up to the maximum subnet mask (/32 for IPv4 and /128 for IPv6).

When multiple match rules of different types are present (such as match community and explicit prefix match), the match rule is allowed only when the match statements of all individual match types match. This is the equivalent of the AND filter. The explicit prefix match is contained by the subject profile (rtctrlSubjP) and will form a logical AND if other match rules are present under the subject profile.

Within a given match type (such as match prefix list), at least one of the match rules statement must match. Multiple explicit prefix match (rtctrlMatchRtDest) can be defined under the same subject profile (rtctrlSubjP) which will form a logical OR.

**Set Rules**

- Set policies must be created to define set rules that are carried with the explicit prefixes such as set community, set tag.

**Aggregation Support for Explicit Prefix List**

Each prefix (rtctrlMatchRtDest) in the match prefixes list can be aggregated to support multiple subnets matching with one prefix list entry.

**Aggregated Prefixes and BD Private Subnets**

Although subnets in the explicit prefix list match may match the BD private subnets using aggregated or exact match, private subnets will not be advertised through the routing protocol using the explicit prefix list. The scope of the BD subnet must be set to "public" for the explicit prefix list feature to advertise the BD subnets.

**Differences in Behavior for 0.0.0.0/0 with Aggregation**

The 0.0.0.0/0 with Aggregate configuration creates an IP prefix-list equivalent to “0.0.0.0/0 le 32”. The 0.0.0.0/0 with Aggregate configuration can be used mainly in two situations:

- “Export Route Control Subnet” with “Aggregate Export” scope in L3Out subnet under the L3Out network (L3Out EPG)
- An explicit prefix-list (Match Prefix rule) assigned to a route map with the name “default-export”
When used with the “Export Route Control Subnet” scope under the L3Out subnet, the route map will only match routes learned from dynamic routing protocols. It will not match BD subnets or directly-connected networks.

When used with the explicit route map configuration, the route map will match all routes, including BD subnets and directly-connected networks.

Consider the following examples to get a better understanding of the expected and unexpected (inconsistent) behavior in the two situations described above.

**Scenario 1**

For the first scenario, we configure a route map (with a name of `rpm_with_catch_all`) using a configuration post similar to the following:

```xml
<rteCtrlProfile name="rpm_with_catch_all">
  <rteCtrlCtxP action="permit">
    <rteCtrlScope/>
  </rteCtrlCtxP>
</rteCtrlProfile>
```

Consider the following example to get a better understanding of the expected and unexpected (inconsistent) behavior in the two situations described above.

**Scenario 1**

For the first scenario, we configure a route map (with a name of `rpm_with_catch_all`) using a configuration post similar to the following:

```xml
<ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="backbone" areaType="regular">
  <ospfIfP authKeyId="1" authType="none">
    <ospfRsIfPol/>
  </ospfIfP>
</ospfExtP>
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
With this route map, what we would expect with 0.0.0.0/0 is that all the routes would go with the property metricType="ospf-type1", but only for the OSPF route.

In addition, we also have a subnet configured under a bridge domain (for example, 209.165.201.0/27), with a bridge domain to L3Out relation, using a route map with a pervasive subnet (fvSubnet) for a static route. However, even though the route map shown above is combinable, we do not want it applied for the subnet configured under the bridge domain, because we want 0.0.0.0/0 in the route map above to apply only for the transit route, not on the static route.

Following is the output for the show route-map and show ip prefix-list commands, where exp-ctx-st-2555939 is the name of the outbound route map for the subnet configured under the bridge domain, and the name of the prefix list is provided within the output from the show route-map command:

```
leaf4# show route-map exp-ctx-st-2555939
route-map exp-ctx-st-2555939, deny, sequence 1
  Match clauses:
    tag: 4294967295
  Set clauses:
route-map exp-ctx-st-2555939, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-st16391-2555939-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:

leaf4# show ip prefix-list IPv4-st16391-2555939-exc-int-inferred-export-dst
ip prefix-list IPv4-st16391-2555939-exc-int-inferred-export-dst: 1 entries
  seq 1 permit 209.165.201.0/27

leaf4#
```

In this situation, everything behaves as expected, because when the bridge domain subnet goes out, it is not applying the rpm_with_catch_all route map policies.

**Scenario 2**

For the second scenario, we configure a "default-export" route map for export route control, where an explicit prefix-list (Match Prefix rule) is assigned to the "default-export" route map, using a configuration post similar to the following:

```
leaf4# show route-map exp-ctx-st-2555939
route-map exp-ctx-st-2555939, deny, sequence 1
  Match clauses:
    tag: 4294967295
  Set clauses:
route-map exp-ctx-st-2555939, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-st16391-2555939-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:

leaf4# show ip prefix-list IPv4-st16391-2555939-exc-int-inferred-export-dst
ip prefix-list IPv4-st16391-2555939-exc-int-inferred-export-dst: 1 entries
  seq 1 permit 209.165.201.0/27
```

With this route map, what we would expect with 0.0.0.0/0 is that all the routes would go with the property metricType="ospf-type1", but only for the OSPF route.

In addition, we also have a subnet configured under a bridge domain (for example, 209.165.201.0/27), with a bridge domain to L3Out relation, using a route map with a pervasive subnet (fvSubnet) for a static route. However, even though the route map shown above is combinable, we do not want it applied for the subnet configured under the bridge domain, because we want 0.0.0.0/0 in the route map above to apply only for the transit route, not on the static route.

Following is the output for the show route-map and show ip prefix-list commands, where exp-ctx-st-2555939 is the name of the outbound route map for the subnet configured under the bridge domain, and the name of the prefix list is provided within the output from the show route-map command:

```
leaf4# show route-map exp-ctx-st-2555939
route-map exp-ctx-st-2555939, deny, sequence 1
  Match clauses:
    tag: 4294967295
  Set clauses:
route-map exp-ctx-st-2555939, permit, sequence 15801
  Match clauses:
    ip address prefix-lists: IPv4-st16391-2555939-exc-int-inferred-export-dst
    ipv6 address prefix-lists: IPv6-deny-all
  Set clauses:

leaf4# show ip prefix-list IPv4-st16391-2555939-exc-int-inferred-export-dst
ip prefix-list IPv4-st16391-2555939-exc-int-inferred-export-dst: 1 entries
  seq 1 permit 209.165.201.0/27
```

In this situation, everything behaves as expected, because when the bridge domain subnet goes out, it is not applying the rpm_with_catch_all route map policies.

**Scenario 2**

For the second scenario, we configure a "default-export" route map for export route control, where an explicit prefix-list (Match Prefix rule) is assigned to the "default-export" route map, using a configuration post similar to the following:
Notice that this default-export route map has similar information as the rpm_with_catch_all route map, where the IP is set to 0.0.0.0/0 (ip=0.0.0.0/0), and the set rule in the default-export route map is configured only with the Set Metric Type (tnRtctrlAttrPName=set_metric_type).

Similar to the situation in the previous example, we also have the same subnet configured under the bridge domain, with a bridge domain to L3Out relation, as we did in the previous example.

However, following is the output in this scenario for the show route-map and show ip prefix-list commands:

```
leaf4# show route-map exp-ctx-st-2555939
route-map exp-ctx-st-2555939, deny, sequence 1
  Match clauses:
    tag: 4294967295
  Set clauses:
    route-map exp-ctx-st-2555939, permit, sequence 8201
      Match clauses:
        ip address prefix-lists:
          IPV4+st16391-2555939-exc-int-out-default-export2set-rule0pfx-only-dst
            ipv6 address prefix-lists: IPv6-deny-all
          Set clauses:
            metric-type type-1
```
Notice that in this situation, when the bridge domain subnet goes out, it is applying the default-export route map policies. In this situation, that route map matches all routes, including BD subnets and directly-connected networks. This is inconsistent behavior.

**Guidelines and Limitations**

- You must choose one of the following two methods to configure your route maps. If you use both methods, it will result in double entries and undefined route maps.
  - Add routes under the bridge domain (BD) and configure a BD to Layer 3 Outside relation
  - Configure the match prefix under rtctrlSubjP match profiles.
- Starting 2.3(x), **deny-static** implicit entry has been removed from Export Route Map. The user needs to configure explicitly the permit and deny entries required to control the export of static routes.
- Route-map per peer in an L3Out is not supported. Route-map can only be applied on L3Out as a whole.

  Following are possible workarounds to this issue:
  - Block the prefix from being advertised from the other side of the neighbor.
  - Block the prefix on the route-map on the existing L3Out where you don't want to learn the prefix, and move the neighbor to another L3Out where you want to learn the prefix and create a separate route-map.
  - Creating route-maps using a mixture of GUI and API commands is not supported. As a possible workaround, you can create a route-map different from the default route-map using the GUI, but the route-map created through the GUI on an L3Out cannot be applied to per-peer.

**Configuring a Route Map/Profile with Explicit Prefix List Using the GUI**

**Before you begin**

- Tenant and VRF must be configured.
- The VRF must be enabled on the leaf switch.

**Procedure**

**Step 1**  
On the menu bar, click **Tenant**, and in the **Navigation** pane, expand **Tenant_name > Networking > External Routed Networks > Match Rules for Route Maps**.
Step 2 Right click **Match Rules for Route Maps**, and click **Create Match Rule for a Route Map**.

Step 3 In the **Create Match Rule for a Route Map** dialog box, enter a name for the rule and choose the desired community terms.

Step 4 In the **Create Match Rule** dialog box, expand **Match Prefix** and perform the following actions:

a) In the **IP** field, enter the explicit prefix list.

The explicit prefix can denote a BD subnet or an external network.

b) Check the **Aggregate** check box only if you desire an aggregate prefix. Click **Update**, and click **Submit**.

The match rule can have one or more of the match destination rules and one or more match community terms. Across the match types, the AND filter is supported, so all conditions in the match rule must match for the route match rule to be accepted. When there are multiple match prefixes in **Match Destination Rules**, the OR filter is supported. Any one match prefix is accepted as a route type if it matches.

Step 5 Under **External Routed Networks**, click and choose the available default layer 3 out.

If you desire another layer 3 out, you can choose that instead.

Step 6 Right-click **Route Maps/Profiles**, and click **Create Route Map/Profile**.

Step 7 In the **Create Route Map** dialog box, use a default route map, or enter a name for the desired route map.

For the purpose of this example, we use **default_export** route map.

Step 8 In the **Type** field, choose **Match Routing Policy Only**.

The Match Routing policy is the global RPC match destination route. The other option in this field is Match Prefix and Routing Policy which is the combinable RPC match destination route.

Step 9 Expand the + icon to display the **Create Route Control Context** dialog box.

Step 10 Enter a name for route control context, and choose the desired options for each field. To deny routes that match criteria defined in match rule (**Step 11**), select the action **deny**. The default action is **permit**.

Step 11 In the **Match Rule** field, choose the rule that was created earlier.

Step 12 In the **Set Rule** field, choose **Create Set Rules for a Route Map**.

Typically in the route map/profile you have a match and so the prefix list is allowed in and out, but in addition some attributes are being set for these routes, so that the routes with the attributes can be matched further.

Step 13 In the **Create Set Rules for a Route Map** dialog box, enter a name for the action rule and check the desired check boxes. Click **Submit**.

Step 14 In the **Create Route Control Context** dialog box, click **OK**. And in the **Create Route Map/Profile** dialog box, click **Submit**.

This completes the creation of the route map/profile. The route map is a combination of match action rules and set action rules. The route map is associated with export profile or import profile or redistribute profile as desired by the user. You can enable a protocol with the route map.
### Configuring Route Map/Profile with Explicit Prefix List Using NX-OS Style CLI

**Before you begin**
- Tenant and VRF must be configured through the NX-OS CLI.
- The VRF must be enabled on the leaf switch through the NX-OS CLI.

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>apic1# configure</code></td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><code>leaf node-id</code></td>
<td>Specifies the leaf to be configured.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>apic1(config)# leaf 101</code></td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>template route group group-name tenant tenant-name</code></td>
<td>Creates a route group template.</td>
</tr>
<tr>
<td>Example:</td>
<td><code>apic1(config-leaf)# template route group gl1 tenant exampleCorp</code></td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>The route group (match rule) can have one or more of the IP prefixes and one or more match community terms. Across the match types, the AND filter is supported, so all conditions in the route group must match for the route match rule to be accepted. When there are multiple IP prefixes in route group, the OR filter is supported. Any one match prefix is accepted as a route type if it matches.</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td>`ip prefix permit prefix/masklen [le{32</td>
<td>128 }]`</td>
</tr>
<tr>
<td>Example:</td>
<td><code>apic1(config-route-group)# ip prefix permit 15.15.15.0/24</code></td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>The IP prefix can denote a BD subnet or an external network. Use optional argument le 32 for IPv4 and le 128 for IPv6 if you desire an aggregate prefix.</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td>`community-list [ standard</td>
<td>expanded] community-list-name expression`</td>
</tr>
<tr>
<td>Example:</td>
<td><code>apic1(config-route-group)# community-list standard com1 65535:20</code></td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Command or Action</td>
<td>Purpose</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>exit</td>
<td>Exit template mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>apic1(config-route-group)# exit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>apic1(config-leaf)#</td>
</tr>
<tr>
<td>7</td>
<td>vrf context tenant tenant-name vrf vrf-name [l3out {BGP</td>
<td>EIGRP</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>apic1(config-leaf)# vrf context tenant exampleCorp vrf v1</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>If you enter the optional l3out string, the L3Out must be an L3Out that you configured through the NX-OS CLI.</td>
</tr>
<tr>
<td>8</td>
<td>template route-profile profile-name [route-control-context-name order-value]</td>
<td>Creates a template containing set actions that should be applied to the matched routes.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>apic1(config-leaf-vrf)# template route-profile rp1 ctxl 1</td>
</tr>
<tr>
<td>9</td>
<td>set attribute value</td>
<td>Add desired attributes (set actions) to the template.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>apic1(config-leaf-vrf-template-route-profile)# set metric 128</td>
</tr>
<tr>
<td>10</td>
<td>exit</td>
<td>Exit template mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>apic1(config-leaf-vrf-template-route-profile)# exit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>apic1(config-leaf-vrf)#</td>
</tr>
<tr>
<td>11</td>
<td>route-map map-name</td>
<td>Create a route-map and enter the route-map configuration mode.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>apic1(config-leaf-vrf)# route-map bgpMap</td>
</tr>
<tr>
<td>12</td>
<td>match route group group-name [order number] [deny]</td>
<td>Match a route group that has already been created, and enter the match mode to configure the route-profile. Additionally choose the keyword Deny if routes matching the match criteria defined in route group needs to be denied. The default is Permit.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>apic1(config-leaf-vrf-route-map)# match route group g1 order 1</td>
</tr>
<tr>
<td>13</td>
<td>inherit route-profile profile-name</td>
<td>Inherit a route-profile (set actions).</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>apic1(config-leaf-vrf-route-map-match)# inherit route-profile rp1</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong></td>
<td>These actions will be applied to the matched routes. Alternatively, the set actions can be configured inline instead of inheriting a route-profile.</td>
</tr>
</tbody>
</table>
### Configuring Route Map/Profile with Explicit Prefix List Using REST API

**Before you begin**
- Tenant and VRF must be configured.

**Procedure**

Configure the route map/profile using explicit prefix list.
Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<fvTenant name="PM" status="">
  <rtctrlAttrP name="set_dest">
    <rtctrlSetComm community="regular:as2-NN2:5:24" />
  </rtctrlAttrP>

  <rtctrlAttrP name="allow_dest">
    <rtctrlMatchRtDest ip="192.169.0.0/24" />
    <rtctrlMatchCommTerm name="term1">
      <rtctrlMatchCommFactor community="regular:as2-NN2:5:24" status="" />
      <rtctrlMatchCommFactor community="regular:as2-NN2:5:25" status="" />
    </rtctrlMatchCommTerm>
    <rtctrlMatchCommRegexTerm commType="regular" regex="200:*" status="" />
  </rtctrlAttrP>

  <rtctrlAttrP name="deny_dest">
    <rtctrlMatchRtDest ip="192.168.0.0/24" />
  </rtctrlAttrP>

  <fvCtx name="ctx" />

  <l3extOut name="L3Out_1" enforceRtctrl="import,export" status="">
    <l3extRsEctx tnFvCtxName="ctx" />
    <l3extLNodeP name="bLeaf">
      <l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="1.2.3.4" />
      <l3extLIfP name="portIF">
        <l3extRaPathL3OutAtt tDn="topology/pod-1/paths-101/pathep-[eth1/25]"
          ifInstT="sub-interface" encap="vlan-1503" addr="10.11.12.11/24" />
        <ospfIfP />
      </l3extLIfP>
      <bgpPeerP addr="5.16.57.18/32" ctrl="send-com" />
      <bgpPeerP addr="6.16.57.18/32" ctrl="send-com" />
    </l3extLNodeP>
    <bgpExtP />
    <ospfExtP areaId="0.0.0.59" areaType="nssa" status="" />
    <l3extInstP name="l3extInstP_1" status="">
      <l3extSubnet ip="17.11.1.11/24" scope="import-security" />
    </l3extInstP>
  </l3extOut>

  <rtctrlProfile name="default-export" type="global" status="">
    <rtctrlCtxP name="ctx_deny" action="deny" order="1">
      <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="deny_dest" status="" />
    </rtctrlCtxP>
    <rtctrlCtxP name="ctx_allow" order="2">
      <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="allow_dest" status="" />
    </rtctrlCtxP>
    <rtctrlScope name="scope" status="">
      <rtctrlRsScopeToAttrP tnRtctrlAttrPName="set_dest" status="" />
    </rtctrlScope>
  </rtctrlProfile>

  <fvBD>  
    <fvRsBDToOut tnL3extOutName="L3Out_1" />
    <fvRsCtx tnFvCtxName="ctx" />
    <fvSubnet ip="40.1.1.12/24" scope="public" />
    <fvSubnet ip="40.1.1.2/24" scope="private" />
    <fvSubnet ip="2003::4/64" scope="public" />
  </fvBD>
</fvTenant>
```
Routing Control Protocols

About Configuring a Routing Control Protocol Using Import and Export Controls

This topic provides a typical example that shows how to configure a routing control protocol using import and export controls. It assumes that you have configured Layer 3 outside network connections with BGP. You can also perform these tasks for a Layer 3 outside network configured with OSPF.

Note

Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link. On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface.

For the appropriate MTU values for each platform, see the relevant configuration guides.

We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1`.

Configuring a Route Control Protocol to Use Import and Export Controls, With the GUI

This example assumes that you have configured the Layer 3 outside network connections using BGP. It is also possible to perform these tasks for a network configured using OSPF.

This task lists steps to create import and export policies. By default, import controls are not enforced, so the import control must be manually assigned.

Before you begin

- The tenant, private network, and bridge domain are created.
- The Layer 3 outside for tenant networks is created.

Procedure

Step 1
On the menu bar, click TENANTS > Tenant_name > Networking > External Routed Networks > Layer3_Outside_name .

Step 2
Right click Layer3_Outside_name and click Create Route Map.

Step 3
In the Create Route Map dialog box, perform the following actions:

a) From the Name field drop-down list, choose the appropriate route profile.
Depending on your selection, whatever is advertised on the specific outside is automatically used.

b) In the **Type** field, choose **Match Prefix AND Routing Policy**.
c) Expand **Order**.

**Step 4**

In the **Create Route Control Context** dialog box, perform the following actions:

a) In the **Order** field, choose the desired order number.
b) In the **Name** field, enter a name for the route control private network.
c) From the **Match Rule** field drop-down list, click **Create Match Rule For a Route Map**.
d) In the **Create Match Rule** dialog box, in the **Name** field, enter a route match rule name. Click **Submit**.

Specify the match community regular expression term and match community terms as desired. Match community factors will require you to specify the name, community and scope.

e) From the **Set Attribute** drop-down list, choose **Create Set Rules For a Route Map**.
f) In the **Create Set Rules For a Route Map** dialog box, in the **Name** field, enter a name for the rule.
g) Check the check boxes for the desired rules you want to set, and choose the appropriate values that are displayed for the choices. Click **Submit**.

The policy is created and associated with the action rule.
h) Click **OK**.
i) In the **Create Route Map** dialog box, click **Submit**.

**Step 5**

In the **Navigation** pane, choose **Route Profile > route_profile_name > route_control_private_network_name**. In the **Work** pane, under **Properties** the route profile policy and the associated action rule name are displayed.

**Step 6**

In the **Navigation** pane, click the **Layer3_Outside_name**.
In the **Work** pane, the **Properties** are displayed.

**Step 7**

(Optional) Click the **Route Control Enforcement** field and check the **Import** check box to enable the import policy.

The import control policy is not enabled by default but can be enabled by the user. The import control policy is supported for BGP and OSPF, but not for EIGRP. If the user enables the import control policy for an unsupported protocol, it will be automatically ignored. The export control policy is supported for BGP, EIGRP, and OSPF.

**Note** If BGP is established over OSPF, then the import control policy is applied only for BGP and ignored for OSPF.

**Step 8**

To create a customized export policy, right-click **Route Map/Profiles**, click **Create Route Map**, and perform the following actions:

a) In the **Create Route Map** dialog box, from the drop-down list in the **Name** field, choose or enter a name for the export policy.
b) Expand the + sign in the dialog box.
c) In the **Create Route Control Context** dialog box, in the **Order** field, choose a value.
d) In the **Name** field, enter a name for the route control private network.
e) (Optional) From the **Match Rule** field drop-down list, choose **Create Match Rule For a Route Map**, and create and attach a match rule policy if desired.
f) From the **Set Attribute** field drop-down list, choose **Create Set Rules For a Route Map** and click **OK**. Alternatively, if desired, you can choose an existing set action, and click **OK**.
g) In the **Create Set Rules For A Route Map** dialog box, in the **Name** field, enter a name.
h) Check the check boxes for the desired rules you want to set, and choose the appropriate values that are displayed for the choices. Click **Submit**.
In the Create Route Control Context dialog box, the policy is created and associated with the action rule.

i) Click OK.

j) In the Create Route Map dialog box, click Submit.

In the Work pane, the export policy is displayed.

Note: To enable the export policy, it must first be applied. For the purpose of this example, it is applied to all the subnets under the network.

**Step 9**

In the Navigation pane, expand External Routed Networks > `External_Routed_Network_name` > Networks > `Network_name`, and perform the following actions:

a) Expand Route Control Profile.

b) In the Name field drop-down list, choose the policy created earlier.

c) In the Direction field drop-down list, choose Route Control Profile. Click Update.

---

### Configuring a Route Control Protocol to Use Import and Export Controls, With the NX-OS Style CLI

**Before you begin**

- The tenant, private network, and bridge domain are created.
- The Layer 3 outside tenant network is configured.

**Procedure**

**Step 1**

Import Route control using match community, match prefix-list

**Example:**

```bash
apic1# configure
apic1(config)# leaf 101

# Create community-list
apic1(config-leaf)# template community-list standard CL_1 65536:20 tenant exampleCorp
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1

# Create Route-map and use it for BGP import control.
apic1(config-leaf-vrf)# route-map bgpMap

# Match prefix-list and set route-profile actions for the match.
apic1(config-leaf-vrf-route-map)# ip prefix-list list1 permit 13.13.13.0/24
apic1(config-leaf-vrf-route-map)# ip prefix-list list1 permit 14.14.14.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list list1
apic1(config-leaf-vrf-route-map-matched)# set tag 200
apic1(config-leaf-vrf-route-map-matched)# set local-preference 64
apic1(config-leaf)# router bgp 100
apic1(config-bgp)# vrf member tenant exampleCorp vrf v1
```
Step 2 Export Route Control using match BD, default-export route-profile

Example:

# Create custom and "default-export" route-profiles
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant exampleCorp vrf v1
apic1(config-leaf-vrf)# template route-profile default-export
apic1(config-leaf-vrf)# template route-profile bd-rtctrl
apic1(config-leaf-vrf)# template route-profile default-export

# Create a Route-map and match on BD, prefix-list
apic1(config-leaf-vrf)# route-map bgpMap
apic1(config-leaf-vrf-route-map)# match bridge-domain bd1
apic1(config-leaf-vrf-route-map)# match prefix-list p1
apic1(config-leaf-vrf-route-map)# inherit route-profile bd-rtctrl

Note In this case, public-subnets from bd1 and prefixes matching prefix-list p1 are exported out using route-profile "default-export", while public-subnets from bd2 are exported out using route-profile "bd-rtctrl".

Configuring a Route Control Protocol to Use Import and Export Controls, With the REST API

This example assumes that you have configured the Layer 3 outside network connections using BGP. It is also possible to perform these tasks for a network using OSPF.

Before you begin

• The tenant, private network, and bridge domain are created.
• The Layer 3 outside tenant network is configured.

Procedure

Configure the route control protocol using import and export controls.

Example:

<l3extOut descr="" dn="uni/tn-Ten_ND/out-L3Out1" enforceRtctrl="export" name="L3Out1" ownerKey="" ownerTag="" targetDscp="unspecified">
    <l3extLNodeP descr="" name="LNodeP1" ownerKey="" ownerTag="" targetDscp="unspecified">
        <l3extRsNodeL3OutAtt rtrId="1.2.3.4" rtrIdLoopBack="yes" targetDscp="unspecified">
Route Control

Configuring a Route Control Protocol to Use Import and Export Controls, With the REST API

```xml
<ldn="topology/pod-1/node-101">
  <l3extLoopBackIfP addr="2000::3" descr="" name=""/>
  </l3extRsNodeL3OutAtt>
  <l3extLIfP descr="" name="IFP1" ownerKey="" ownerTag="" tag="yellow-green">
    <ospfIfP authKeyId="1" authType="none" descr="" name="">
      <ospfRsIfPol tnOspfIfPolName=""/>
    </ospfIfP>
    <l3extRsNdIfPol tnNdIfPolName=""/>
    <l3extRsPathL3OutAtt addr="10.11.12.10/24" descr="" encap="unknown" ifInstT="l3-port"
      llAddr="::" mac="00:22:BD:F8:19:FF" mtu="1500" tDn="topology/pod-1/paths-101/pathep-[eth1/17]"
      targetDscp="unspecified"/>
  </l3extLIfP>
  </l3extLNodeP>
  <l3extRsEctx tnFvCtxName="PVN1"/>
  <l3extInstP descr="" matchT="AtleastOne" name="InstP1" prio="unspecified"
    targetDscp="unspecified">
    <fvRsCustQosPol tnQosCustomPolName=""/>
    <l3extSubnet aggregate="" descr="" ip="192.168.1.0/24" name="" scope=""/>
  </l3extInstP>
  <ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="0.0.0.1"
    areaType="nssa" descr=""/>
  <rtctrlProfile descr="" name="default-export" ownerKey="" ownerTag="">
    <rtctrlCtxP descr="" name="routecontrolpvtnw" order="3">
      <rtctrlScope descr="" name="">
        <rtctrlRsScopeToAttrP tnRtctrlAttrPName="actionruleprofile2"/>
      </rtctrlScope>
    </rtctrlCtxP>
    </rtctrlProfile>
  </l3extOut>
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
CHAPTER 9

Common Pervasive Gateway

This chapter contains the following sections:

- Overview, on page 143
- Configuring Common Pervasive Gateway Using the GUI, on page 144
- Configuring Common Pervasive Gateway Using the NX-OS Style CLI, on page 146
- Configuring Common Pervasive Gateway Using the REST API, on page 146

Overview

The Common Pervasive Gateway feature is being deprecated and is not actively maintained anymore.

When operating more than one Cisco ACI fabric, we highly recommend that you deploy Multi-Site instead of interconnecting multiple individual ACI fabrics to each other through leaf switches using the Common Pervasive Gateway feature. The Common Pervasive Gateway feature is currently not supported because no validations and quality assurance tests are performed in this topology for many other new features, such as L3 multicast. Hence, although Cisco ACI had the Common Pervasive Gateway feature for interconnecting ACI fabrics prior to Multi-Site, we highly recommend that you design a new ACI fabric with Multi-Site instead when there is a requirement to interconnect separate APIC domains.

This example shows how to configure Common Pervasive Gateway for IPv4 when using the Cisco APIC.

Two ACI fabrics can be configured with an IPv4 common gateway on a per bridge domain basis. Doing so enables moving one or more virtual machine (VM) or conventional hosts across the fabrics while the host retains its IP address. VM host moves across fabrics can be done automatically by the VM hypervisor. The ACI fabrics can be co-located, or provisioned across multiple sites. The Layer 2 connection between the ACI fabrics can be a local link, or can be across a bridged network. The following figure illustrates the basic common pervasive gateway topology.
Depending upon the topology used to interconnect two Cisco ACI fabrics, it is required that the interconnecting devices filter out the traffic source with the Virtual MAC address of the gateway switch virtual interface (SVI).

**Configuring Common Pervasive Gateway Using the GUI**

**Before you begin**

- The tenant and VRF are created.
- The bridge domain virtual MAC address and the subnet virtual IP address must be the same across all ACI fabrics for that bridge domain. Multiple bridge domains can be configured to communicate across connected ACI fabrics. The virtual MAC address and the virtual IP address can be shared across bridge domains.
- The Bridge domain that is configured to communicate across ACI fabrics must be in flood mode
- Only one EPG from a bridge domain (If the BD has multiple EPGs) should be configured on a border Leaf on the port which is connected to the second Fabric.
- Do not connect hosts directly to an inter-connected Layer 2 network that enables a pervasive common gateway among the two ACI fabrics.

**Procedure**

1. On the menu bar, click **Tenants**.
2. In the **Navigation** pane, expand the **Tenant_name > Networking > Bridge Domains**.
3. Right-click **Bridge Domains**, and click **Create Bridge Domain**.
4. In the **Create Bridge Domain** dialog box, perform the required actions to choose the appropriate attributes:
a) In the **Main** tab, in the **Name** field, enter a name for the bridge domain, and choose the desired values for the remaining fields.

b) In the **L3 configurations** tab, expand Subnets, and in the **Create Subnets** dialog box, in the **Gateway IP** field, enter the IP address.

   For example, 192.0.2.1/24.

c) In the **Treat as virtual IP address** field, check the check box.

d) In the **Make this IP address primary** field, check the check box to specify this IP address for DHCP relay.

   Checking this check box affects DHCP relay only.

e) Click **Ok**, then click **Next** to advance to the **Advanced/Troubleshooting** tab, then click **Finish**.

### Step 5

Double click the **Bridge Domain** that you just created in the **Work** pane, and perform the following action:

a) Click the **Policy** tab, then click the **L3 Configurations** subtab.

b) Expand **Subnets** again, and in the **Create Subnets** dialog box, to create the physical IP address in the **Gateway IP** field, use the same subnet which is configured as the virtual IP address.

   For example, if you used 192.0.2.1/24 for the virtual IP address, you might use 192.0.2.2/24 here for the physical IP address.

   **Note**  
   The Physical IP address must be unique across the ACI fabric.

c) Click **Submit** to complete the configuration in the **Create Subnet** window.

### Step 6

In the **L3 Configurations** tab for the same bridge domain that you just created, click the **Virtual MAC Address** field, and change **Not Configured** to the appropriate value, then click **Submit**.

**Note**  
The default BD MAC address values are the same for all ACI fabrics; this configuration requires the bridge domain MAC values to be unique for each ACI fabric.

   Confirm that the bridge domain MAC (pmac) values for each fabric are unique.

**Note**  
This step essentially ties the virtual MAC address that you enter in this field with the virtual IP address that you entered in the previous step. If you were to delete the virtual MAC address at some point in the future, you should also remove the check from the **Treat as virtual IP address** field for the IP address that you entered in the previous step.

### Step 7

To create an L2Out EPG to extend the BD to another fabric, in the Navigation pane, right-click **External Bridged Networks** and open the **Create Bridged Outside** dialog box, and perform the following actions:

a) In the **Name** field, enter a name for the bridged outside.

b) In the **Bridge Domain** field, select the bridge domain already previously created.

c) In the **Encap** field, enter the VLAN encapsulation to match the other fabric l2out encapsulation.

d) In the **Path Type** field, select **Port**, **PC**, or **VPC** to deploy the EPG and click **Next**.

e) To create an External EPG network click in the **Name** field, enter a name for the network and you can specify the QoS class and click **Finish** to complete Common Pervasive configuration.
Configuring Common Pervasive Gateway Using the NX-OS Style CLI

Before you begin

• The tenant, VRF, and bridge domain are created.

Procedure

Configure Common Pervasive Gateway.

Example:

apic1#configure
apic1(config)#tenant demo
apic1(config-tenant)#bridge-domain test
apic1(config-tenant-bd)#l2-unknown-unicast flood
apic1(config-tenant-bd)#arp flooding
apic1(config-tenant-bd)#exit

apic1(config-tenant)#interface bridge-domain test
apic1(config-tenant-interface)#multi-site-mac-address 12:34:56:78:9a:bc
apic1(config-tenant-interface)#mac-address 00:CC:CC:CC:C1:01 (Should be unique for each ACI fabric)
apic1(config-tenant-interface)#ip address 192.168.10.1/24 multi-site
apic1(config-tenant-interface)#ip address 192.168.10.254/24 (Should be unique for each ACI fabric)

Configuring Common Pervasive Gateway Using the REST API

Before you begin

• The tenant, VRF, and bridge domain are created.

Procedure

Configure Common Pervasive Gateway.

Example:

<!-Things that are bolded only matters--><?xml version="1.0" encoding="UTF-8"?>
<!-- api/policymgr/mo/.xml -->
<polUni>
    <fvTenant name="test">
        <fvCtx name="test"/>

        <fvBD name="test" vmac="12:34:56:78:9a:bc">
            ...
        </fvBD>
    </fvTenant>
</polUni>
<fvRsCtx tnFvCtxName="test"/>
<!-- Primary address -->
<fvSubnet ip="192.168.15.254/24" preferred="yes"/>
<!-- Virtual address -->
<fvSubnet ip="192.168.15.1/24" virtual="yes"/>
</fvBD>

<fvAp name="test">
  <fvAEPg name="web">
    <fvRsBd tnFvBDName="test">
      <fvRsPathAtt tdn="topology/pod-1/paths-101/pathep-[eth1/3]" encap="vlan-1002"/>
    </fvRsPathAtt>
  </fvAEPg>
</fvAp>
</fvTenant>
</polUni>
Static Route on a Bridge Domain

This chapter contains the following sections:

- About Static Routes in Bridge Domains, on page 149
- Configuring a Static Route on a Bridge Domain Using the GUI, on page 149
- Configuring a Static Route on a Bridge Domain Using the NX-OS Style CLI, on page 150
- Configuring a Static Route on a Bridge Domain Using the REST API, on page 151

About Static Routes in Bridge Domains

With Cisco APIC Release 3.0(2), support is added to configure a static route in a pervasive bridge domain (BD) to enable routes to virtual services behind firewalls.

This feature enables endpoint (EP) reachability to IP addresses that are not directly connected to the pervasive bridge domain, using regular EPGs.

When a static route is configured, the APIC deploys it to all the leaf switches that use the bridge domain and all the leaf switches that have contracts associated to the bridge domain.

You can configure endpoint reachability using the APIC GUI, the NX-OS Style CLI, and the REST API.

Guidelines and Limitations

- The subnet mask must be /32 (/128 for IPv6) pointing to one IP address out of the fabric. Do not add routes within bridge domain subnets that are already defined.
- The next hop must be inside the same bridge domain that this EPG is associated with.
- The feature is supported on Cisco Nexus 9000 series switches with names that end in EX, and later (for example, N9K-C93180LC-EX).

Configuring a Static Route on a Bridge Domain Using the GUI

- When creating the subnet for the static route, it is configured under the EPG (fvSubnet object under fvAEPg), associated with the pervasive BD (fBD), not the BD itself.
- The subnet mask must be /32 (/128 for IPv6) pointing to one IP address or one endpoint. It is contained in the EPG associated with the pervasive BD.
Before you begin

The tenant, VRF, BD, and EPG are created.

Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>On the menu bar, click Tenants &gt; tenant-name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>In the Navigation pane, expand Application Profiles and click the application profile name.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Click Application EPGs and expand the EPG for the static route.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Expand Subnets, right-click the subnet for the static route, and choose Create Endpoints Behind EPG Subnet.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Enter the NextHop IP Address for the endpoint and click Update.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Click Submit.</td>
</tr>
</tbody>
</table>

Configuring a Static Route on a Bridge Domain Using the NX-OS Style CLI

To configure a static route in a pervasive bridge domain (BD), use the following NX-OS style CLI commands:

Before you begin

The tenant, VRF, BD and EPG are configured.

- When creating the subnet for the static route, it is configured under the EPG (fvSubnet object under fvAEPg), associated with the pervasive BD (fvBD), not the BD itself.
- The subnet mask must be /32 (/128 for IPv6) pointing to one IP address or one endpoint. It is contained in the EPG associated with the pervasive BD.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> apic1# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> tenant tenant-name</td>
<td>Creates a tenant or enters tenant configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> apic1(config)# tenant t1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> application ap-name</td>
<td>Creates an application profile or enters application profile mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> apic1(config-tenant)# application ap1</td>
<td></td>
</tr>
</tbody>
</table>
**Command or Action** | **Purpose**
---|---
**Step 4**  
epg `epg-name`  
*Example:*  
apic1(config-tenant-app)# epg ep1  
<> <A.B.C.D> [scope <scope>]
| Creates an EPG or enters EPG configuration mode.

**Step 5**  
endpoint ip `A.B.C.D/LEN` next-hop `A.B.C.D` [scope `scope`]
*Example:*  
apic1(config-tenant-app-epg)# endpoint ip 125.12.1.1/32 next-hop 26.0.14.101
| Creates an endpoint behind the EPG. The subnet mask must be /32 (/128 for IPv6) pointing to one IP address or one endpoint.

### Example

The following example shows the commands to configure an endpoint behind an EPG.

```
apic1# config
  apic1(config)# tenant t1
  apic1(config-tenant)# application api
  apic1(config-tenant-app)# epg ep1
  apic1(config-tenant-app-epg)# endpoint ip 125.12.1.1/32 next-hop 26.0.14.101
```

---

### Configuring a Static Route on a Bridge Domain Using the REST API

- When creating the subnet for the static route, it is configured under the EPG (fvSubnet object under fvAEPg), associated with the pervasive BD (fvBD), not the BD itself.
- The subnet mask must be /32 (/128 for IPv6) pointing to one IP address or one endpoint. It is contained in the EPG associated with the pervasive BD.

### Before you begin

The tenant, VRF, BD, and EPG have been created.

### Procedure

To configure a static route for the BD used in a pervasive gateway, enter a post such as the following example:

**Example:**

```
<fvAEPg name="ep1">
  <fvRsBd tnFvBDName="bd1"/>
  <fvSubnet ip="2002:0db8:85a2:0000:0000:8a2e:0370:7344/128" ctrl="no-default-gateway" >
  </fvEpReachability>
```
Configuring a Static Route on a Bridge Domain Using the REST API

```xml
<ipNextHopEpP nhAddr="2001:0db8:85a3:0000:0000:8a2e:0370:7343/128" />
</fvEpReachability>
</fvSubnet>
</fvAEPg>
```
MP-BGP Route Reflectors

This chapter contains the following sections:

• BGP Protocol Peering to External BGP Speakers, on page 153
• Configuring an MP-BGP Route Reflector Using the GUI, on page 155
• Configuring an MP-BGP Route Reflector for the ACI Fabric, on page 155
• Configuring an MP-BGP Route Reflector Using the REST API, on page 156
• Verifying the MP-BGP Route Reflector Configuration, on page 156

BGP Protocol Peering to External BGP Speakers

ACI supports peering between the border leaves and the external BGP speakers using iBGP and eBGP. ACI supports the following connections for BGP peering:

• iBGP peering over OSPF
• eBGP peering over OSPF
• iBGP peering over direct connection
• eBGP peering over direct connection
• iBGP peering over static route

Note

When OSPF is used with BGP peering, OSPF is only used to learn and advertise the routes to the BGP peering addresses. All route control applied to the Layer 3 Outside Network (EPG) are applied at the BGP protocol level.

ACI supports a number of features for iBGP and eBGP connectivity to external peers. The BGP features are configured on the BGP Peer Connectivity Profile.

The BGP peer connectivity profile features are described in the following table:
<table>
<thead>
<tr>
<th>BGP Features</th>
<th>Feature Description</th>
<th>NX-OS Equivalent Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow Self-AS</td>
<td>Works with Allowed AS Number Count setting.</td>
<td>allowas-in</td>
</tr>
<tr>
<td>Disable peer AS check</td>
<td>Disable checking of the peer AS number when advertising.</td>
<td>disable-peer-as-check</td>
</tr>
<tr>
<td>Next-hop self</td>
<td>Always set the next hop attribute to the local peering address.</td>
<td>next-hop-self</td>
</tr>
<tr>
<td>Send community</td>
<td>Send the community attribute to the neighbor.</td>
<td>send-community</td>
</tr>
<tr>
<td>Send community extended</td>
<td>Send the extended community attribute to the neighbor.</td>
<td>send-community extended</td>
</tr>
<tr>
<td>Password</td>
<td>The BGP MD5 authentication.</td>
<td>password</td>
</tr>
<tr>
<td>Allowed AS Number Count</td>
<td>Works with Allow Self-AS feature.</td>
<td>allowas-in</td>
</tr>
<tr>
<td>Disable connected check</td>
<td>Disable connected check for the directly connected EBGP neighbors (allowing EBGP neighbor peering from the loopbacks).</td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>Set the TTL value for EBGP multihop connections. It is only valid for EBGP.</td>
<td>ebgp-multihop &lt;TTL&gt;</td>
</tr>
<tr>
<td>Autonomous System Number</td>
<td>Remote Autonomous System number of the peer.</td>
<td>neighbor &lt;x.x.x.x&gt; remote-as</td>
</tr>
<tr>
<td>Local Autonomous System Number Configuration</td>
<td>Options when using the Local AS feature. (No Prepend+replace-AS+dual-AS etc).</td>
<td></td>
</tr>
<tr>
<td>Local Autonomous System Number</td>
<td>The local AS feature used to advertise a different AS number than the AS assigned to the fabric MP-BGP Route Reflector Profile. It is only supported for the EBGP neighbors and the local AS number must be different than the route reflector policy AS.</td>
<td>local-as xxx &lt;no-prepend&gt; &lt;replace-as&gt; &lt;dual-as&gt;</td>
</tr>
</tbody>
</table>
Configuring an MP-BGP Route Reflector Using the GUI

Procedure

Step 1
On the menu bar, choose System > System Settings.

Step 2
In the Navigation pane, right-click BGP Route Reflector, and click Create Route Reflector Node Policy EP.

Step 3
In the Create Route Reflector Node Policy EP dialog box, from the Spine Node drop-down list, choose the appropriate spine node. Click Submit.

Note: Repeat the above steps to add additional spine nodes as required.

The spine switch is marked as the route reflector node.

Step 4
In the BGP Route Reflector properties area, in the Autonomous System Number field, choose the appropriate number. Click Submit.

Note: The autonomous system number must match the leaf connected router configuration if Border Gateway Protocol (BGP) is configured on the router. If you are using routes learned using static or Open Shortest Path First (OSPF), the autonomous system number value can be any valid value.

Step 5
On the menu bar, choose Fabric > Fabric Policies > POD Policies.

Step 6
In the Navigation pane, expand and right-click Policy Groups, and click Create POD Policy Group.

Step 7
In the Create POD Policy Group dialog box, in the Name field, enter the name of a pod policy group.

Step 8
In the BGP Route Reflector Policy drop-down list, choose the appropriate policy (default). Click Submit.

The BGP route reflector policy is associated with the route reflector pod policy group, and the BGP process is enabled on the leaf switches.

Step 9
In the Navigation pane, choose Pod Policies > Profiles > default. In the Work pane, from the Fabric Policy Group drop-down list, choose the pod policy that was created earlier. Click Submit.

The pod policy group is now applied to the fabric policy group.

Configuring an MP-BGP Route Reflector for the ACI Fabric

To distribute routes within the ACI fabric, an MP-BGP process must first be operating, and the spine switches must be configured as BGP route reflectors.

The following is an example of an MP-BGP route reflector configuration:

```
Note
In this example, the BGP fabric ASN is 100. Spine switches 104 and 105 are chosen as MP-BGP route reflectors.
```

```
apic1(config)# bgp-fabric
```
Configuring an MP-BGP Route Reflector Using the REST API

Procedure

Step 1  Mark the spine switches as route reflectors.

**Example:**

POST https://apic-ip-address/api/policymgr/mo/uni/fabric.xml

```
<bgpInstPol name="default">
  <bgpAsP asn="1" />
  <bgpRRP>
    <bgpRRNodePEp id="<spine_id1>"/>
    <bgpRRNodePEp id="<spine_id2>"/>
  </bgpRRP>
</bgpInstPol>
```

Step 2  Set up the pod selector using the following post.

**Example:**

For the FuncP setup—

POST https://apic-ip-address/api/policymgr/mo/uni.xml

```
<fabricFuncP>
  <fabricPodPGrp name="bgpRRPodGrp">
    <fabricRsPodPGrpBGPRRP tnBgpInstPolName="default"/>
  </fabricPodPGrp>
</fabricFuncP>
```

**Example:**

For the PodP setup—

POST https://apic-ip-address/api/policymgr/mo/uni.xml

```
<fabricPodP name="default">
  <fabricPodS name="default" type="ALL">
    <fabricRsPodPGrp tDn="uni/fabric/funcprof/podpgrp-bgpRRPodGrp"/>
  </fabricPodS>
</fabricPodP>
```

Verifying the MP-BGP Route Reflector Configuration

Procedure

Step 1  Verify the configuration by performing the following actions:
a) Use secure shell (SSH) to log in as an administrator to each leaf switch as required.
b) Enter the `show processes | grep bgp` command to verify the state is S.

If the state is NR (not running), the configuration was not successful.

**Step 2**

Verify that the autonomous system number is configured in the spine switches by performing the following actions:

a) Use the SSH to log in as an administrator to each spine switch as required.

b) Execute the following commands from the shell window

   **Example:**
   
   ```
   cd /mit/sys/bgp/inst
   ```

   **Example:**
   
   ```
   grep asn summary
   ```

The configured autonomous system number must be displayed. If the autonomous system number value displays as 0, the configuration was not successful.
Verifying the MP-BGP Route Reflector Configuration
Switch Virtual Interface

This chapter contains the following sections:

- SVI External Encapsulation Scope, on page 159
- SVI Auto State, on page 164

SVI External Encapsulation Scope

About SVI External Encapsulation Scope

In the context of a Layer 3 Out configuration, a switch virtual interfaces (SVI), is configured to provide connectivity between the ACI leaf switch and a router.

By default, when a single Layer 3 Out is configured with SVI interfaces, the VLAN encapsulation spans multiple nodes within the fabric. This happens because the ACI fabric configures the same bridge domain (VXLAN VNI) across all the nodes in the fabric where the Layer 3 Out SVI is deployed as long as all SVI interfaces use the same external encapsulation (SVI) as shown in the figure.

However, when different Layer 3 Outs are deployed, the ACI fabric uses different bridge domains even if they use the same external encapsulation (SVI) as shown in the figure:
Starting with Cisco APIC release 2.3, it is now possible to choose the behavior when deploying two (or more) Layer 3 Outs using the same external encapsulation (SVI).

The encapsulation scope can now be configured as Local or VRF:

- Local scope (default): The example behavior is displayed in the figure titled *Local Scope Encapsulation and Two Layer 3 Outs*.

- VRF scope: The ACI fabric configures the same bridge domain (VXLAN VNI) across all the nodes and Layer 3 Out where the same external encapsulation (SVI) is deployed. See the example in the figure titled *VRF Scope Encapsulation and Two Layer 3 Outs*.
Encapsulation Scope Syntax

The options for configuring the scope of the encapsulation used for the Layer 3 Out profile are as follows:

- **Ctx**—The same external SVI in all Layer 3 Outs in the same VRF for a given VLAN encapsulation. This is a global value.

- **Local**—A unique external SVI per Layer 3 Out. This is the default value.

The mapping among the CLI, API, and GUI syntax is as follows:

<table>
<thead>
<tr>
<th>CLI</th>
<th>API</th>
<th>GUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>L3out</td>
<td>local</td>
<td>Local</td>
</tr>
<tr>
<td>vrf</td>
<td>ctx</td>
<td>VRF</td>
</tr>
</tbody>
</table>

The CLI commands to configure encapsulation scope are only supported when the VRF is configured through a named Layer 3 Out configuration.

Guidelines for SVI External Encapsulation Scope

To use SVI external encapsulation scope, follow these guidelines:

- If deploying the Layer 3 Outs on the same node, the OSPF areas in both the Layer 3 Outs must be different.

- If deploying the Layer 3 Outs on the same node, the BGP peer configured on both the Layer 3 Outs must be different.
Configuring SVI External Encapsulation Scope Using the GUI

Before you begin
- The tenant and VRF configured.
- A Layer 3 Out is configured and a logical node profile under the Layer 3 Out is configured.

Procedure

Step 1
On the menu bar, click > Tenants > Tenant_name. In the Navigation pane, click Networking > External Routed Networks > External Routed Network_name > Logical Node Profiles > Logical Interface Profile.

Step 2
In the Navigation pane, right-click Logical Interface Profile, and click Create Interface Profile.

Step 3
In the Create Interface Profile dialog box, perform the following actions:

a) In the Step 1 Identity screen, in the Name field, enter a name for the interface profile.
b) In the remaining fields, choose the desired options, and click Next.
c) In the Step 2 Protocol Profiles screen, choose the desired protocol profile details, and click Next.
d) In the Step 3 Interfaces screen, click the SVI tab, and click the + icon to open the Select SVI dialog box.
e) In the Specify Interface area, choose the desired values for the various fields.
f) In the Encap Scope field, choose the desired encapsulation scope value. Click OK.

The default value is Local.

The SVI External encapsulation scope is configured in the specified interface.

Configuring SVI Interface Encapsulation Scope Using NX-OS Style CLI

The following example displaying steps for an SVI interface encapsulation scope setting is through a named Layer 3 Out configuration.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>Enter the configure mode.</td>
<td>Enters the configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>apic1# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>Enter the switch mode.</td>
<td>Enters the switch mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>apic1(config)# leaf 104</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td>Create the VLAN interface.</td>
<td>Creates the VLAN interface. The VLAN range</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>is 1-4094.</td>
</tr>
<tr>
<td>apic1(config-leaf)# interface vlan 2001</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring SVI Interface Encapsulation Scope Using the REST API

#### Before you begin

The interface selector is configured.

#### Procedure

Configure the SVI interface encapsulation scope.

**Example:**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- /api/node/mo/.xml -->
<polUni>
  <fVTenant name="coke">
    <l3extOut descr="" dn="uni/tn-coke/out-l3out1" enforceRtctrl="export" name="l3out1" nameAlias="" ownerKey="" ownerTag="" targetDscp="unspecified">
      <l3extRsL3DomAtt tDn="uni/l3dom-Dom1"/>
      <l3extRsEctx tnFvCtxName="vrf0"/>
      <l3extLNodeP configIssues="" descr="" name="__ui_node_101" nameAlias="" ownerKey="" ownerTag="" tag="yellow-green" targetDscp="unspecified">
        <l3extRsNodeL3OutAtt rtrId="1.1.1.1" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
        <l3extLIfP descr="" name="int1_11" nameAlias="" ownerKey="" ownerTag="" tag="yellow-green">
          <l3extRsPathL3OutAtt addr="0.0.0.0" llAddr="0.0.0.0" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" tDn="topology/pod-1/paths-101/pathep-[eth1/5]" targetDscp="unspecified"/>
          <l3extRsNdIfP olName=""/>
          <l3extRsIngressQosDppPol tnQosDppPolName=""/>
          <l3extRsEgressQosDppPol tnQosDppPolName=""/>
        </l3extLIfP>
      </l3extLNodeP>
    </l3extLNodeP>
    <l3extInstP descr="" matchT="AtleastOne" name="epgl" nameAlias="" prefGrMemb="exclude" prio="unspecified" targetDscp="unspecified">
      <l3extSubnet aggregate="" descr="" ip="101.10.10.1/24" name="" nameAlias="" scope="import-security"/>
      <fvRsQosPol polName=""/>
    </l3extInstP>
  </l3extLNodeP>
</l3extOut>
```

---

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 4** | Specify the encapsulation scope.  
**Example:**
apci1(config-leaf-if)# encap scope vrf context | Specifies the encapsulation scope. |
| **Step 5** | Exit the interface mode.  
**Example:**
apci1(config-leaf-if)# exit | Exits the interface mode. |
SVI Auto State

About SVI Auto State

Note
This feature is available in the APIC Release 2.2(3x) release and going forward with APIC Release 3.1(1). It is not supported in APIC Release 3.0(x).

The Switch Virtual Interface (SVI) represents a logical interface between the bridging function and the routing function of a VLAN in the device. SVI can have members that are physical ports, direct port channels, or virtual port channels. The SVI logical interface is associated with VLANs, and the VLANs have port membership.

The SVI state does not depend on the members. The default auto state behavior for SVI in Cisco APIC is that it remains in the up state when the auto state value is disabled. This means that the SVI remains active even if no interfaces are operational in the corresponding VLAN/s.

If the SVI auto state value is changed to enabled, then it depends on the port members in the associated VLANs. When a VLAN interface has multiple ports in the VLAN, the SVI goes to the down state when all the ports in the VLAN go down.

Table 5: SVI Auto State

<table>
<thead>
<tr>
<th>SVI Auto State</th>
<th>Description of SVI State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disabled</td>
<td>SVI remains in the up state even if no interfaces are operational in the corresponding VLAN/s. Disabled is the default SVI auto state value.</td>
</tr>
<tr>
<td>Enabled</td>
<td>SVI depends on the port members in the associated VLANs. When a VLAN interface contains multiple ports, the SVI goes into the down state when all the ports in the VLAN go down.</td>
</tr>
</tbody>
</table>

Guidelines and Limitations for SVI Auto State Behavior

Read the following guidelines:

- When you enable or disable the auto state behavior for SVI, you configure the auto state behavior per SVI. There is no global command.
Configuring SVI Auto State Using the GUI

Before you begin

- The tenant and VRF configured.
- A Layer 3 Out is configured and a logical node profile and a logical interface profile under the Layer 3 Out is configured.

Procedure

Step 1
On the menu bar, click > Tenants > Tenant_name. In the Navigation pane, click Networking > External Routed Networks > External Routed Network_name > Logical Node Profiles > Logical Interface Profile.

Step 2
In the Navigation pane, expand Logical Interface Profile, and click the appropriate logical interface profile.

Step 3
In the Work pane, click the + sign to display the SVI dialog box.

Step 4
To add an additional SVI, in the SVI dialog box, perform the following actions:

a) In the Path Type field, choose the appropriate path type.
b) In the Path field, from the drop-down list, choose the appropriate physical interface.
c) In the Encap field, choose the appropriate values.
d) In the Auto State field, choose the SVI in the Work pane, to view/change the Auto State value.

The default value is Disabled.

Note
To verify or change the Auto State value for an existing SVI, choose the appropriate SVI and verify or change the value.

Configuring SVI Auto State Using NX-OS Style CLI

Before you begin

- The tenant and VRF configured.
- A Layer 3 Out is configured and a logical node profile and a logical interface profile under the Layer 3 Out is configured.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>Enter the configure mode.</td>
<td>Enters the configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1# configure</td>
<td></td>
</tr>
</tbody>
</table>

<p>| <strong>Step 2</strong>              |                                              |
| Enter the switch mode.  | Enters the switch mode.                      |
| Example:                |                                              |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>apic1(config)# leaf 104</code></td>
<td></td>
</tr>
</tbody>
</table>

**Step 3**

Create the VLAN interface.

*Example:*

`apic1(config-leaf)# interface vlan 2001`

**Step 4**

Enable SVI auto state.

*Example:*

`apic1(config-leaf-if)# autostate`

**Step 5**

Exit the interface mode.

*Example:*

`apic1(config-leaf-if)# exit`

---

**Configuring SVI Auto State Using the REST API**

**Before you begin**

- The tenant and VRF configured.
- A Layer 3 Out is configured and a logical node profile and a logical interface profile under the Layer 3 Out is configured.

**Procedure**

Enable the SVI auto state value.

*Example:*

```xml
<fvTenant name="t1">
    <l3extOut name="out1">
        <l3extLNodeP name="__ui_node_101">
            <l3extLIfP descr="" name="__ui_eth1_10_vlan_99_af_ipv4">
                <l3extRsPathL3OutAtt addr="19.1.1.1/24" autostate="enabled" descr="">
                    <encap="vlan-100" encapScope="local" ifInstT="ext-svi" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" tDn="topology/pod-1/paths-101/pathep-[eth1/10]" targetDscp="unspecified" />
            </l3extLIfP>
        </l3extLNodeP>
    </l3extOut>
</fvTenant>
```

To disable the autostate, you must change the value to disabled in the above example. For example, `autostate="disabled"`. 
Chapter 13

Shared Services

This chapter contains the following sections:

- Shared Layer 3 Out, on page 167
- Layer 3 Out to Layer 3 Out Inter-VRF Leaking, on page 170

Shared Layer 3 Out

A shared Layer 3 outside network (l3extOut) configuration provides routed connectivity to external networks as a shared service. An l3extOut profile (l3extInstP) EPG provides routed connectivity to external networks. It can be provisioned as a shared service in any tenant (user, common, infra, or mgmt). Prior to release 1.2(1x), this configuration was only supported in the user and common tenants. An EPG in any tenant can use a shared services contract to connect with an l3extInstP EPG regardless of where in the fabric that l3extInstP EPG is provisioned. This simplifies the provisioning of routed connectivity to external networks; multiple tenants can share a single l3extInstP EPG for routed connectivity to external networks. Sharing an l3extInstP EPG is more efficient because it consumes only one session on the switch regardless of how many EPGs use the single shared l3extInstP EPG.

Note

All switches that will use l3extInstP EPG shared service contracts require the hardware and software support available starting with the APIC 1.2(1x) and switch 11.2(1x) releases. Refer to the Cisco APIC Management, Installation, Upgrade, and Downgrade Guide and Release Notes documentation for more details.

The figure below illustrates the major policy model objects that are configured for a shared l3extInstP EPG.
Take note of the following guidelines and limitations for shared Layer 3 outside network configurations:

- No tenant limitations: Tenants A and B can be any kind of tenant (user, common, infra, mgmt.). The shared l3extInstP EPG does not have to be in the common tenant.

- Flexible placement of EPGs: EPG A and EPG B in the illustration above are in different tenants. EPG A and EPG B could use the same bridge domain and VRF, but they are not required to do so. EPG A and EPG B are in different bridge domains and different VRFs but still share the same l3extInstP EPG.

- A subnet can be private, public, or shared. A subnet that is to be advertised into a consumer or provider EPG of an L3extOut must be set to shared. A subnet that is to be exported to an L3extOut must be set to public.

- The shared service contract is exported from the tenant that contains the l3extInstP EPG that provides shared Layer 3 outside network service. The shared service contract is imported into the tenants that contain the EPGs that consume the shared service.

- Do not use taboo contracts with a shared L3 out; this configuration is not supported.

- The l3extInstP as a shared service provider is supported, but only with non l3extInstP consumers (where the L3extOut EPG is the same as the l3extInstP).

- Traffic Disruption (Flap): When an l3instP EPG is configured with an external subnet of 0.0.0.0/0 with the scope property of the l3instP subnet set to shared route control (shared-rctrl), or shared security (shared-security), the VRF is redeployed with a global pcTag. This will disrupt all the external traffic in that VRF (because the VRF is redeployed with a global pcTag).

- Prefixes for a shared L3extOut must be unique. Multiple shared L3extOut configurations with the same prefix in the same VRF will not work. Be sure that the external subnets (external prefixes) that are advertised into a VRF are unique (the same external subnet cannot belong to multiple l3instPs). An L3extOut configuration (for example, named L3Out1) with prefix1 and a second Layer 3 outside
configuration (for example, named L3Out2) also with prefix1 belonging to the same VRF will not work (because only 1 pcTag is deployed). Different behaviors of L3extOut are possible when configured on
the same leaf switch under the same VRF. The two possible scenarios are as follows:

- Scenario 1 has an L3extOut with an SVI interface and two subnets (10.10.10.0/24 and 0.0.0.0/0) defined. If ingress traffic on the Layer 3 outside network has the matching prefix 10.10.10.0/24, then the ingress traffic uses the External EPG pcTag. If ingress traffic on the Layer 3 Outside network has the matching default prefix 0.0.0.0/0, then the ingress traffic uses the External Bridge pcTag.

- Scenario 2 has an L3extOut using a routed or routed-sub-interface with two subnets (10.10.10.0/24 and 0.0.0.0/0) defined. If ingress traffic on the Layer 3 outside network has the matching prefix 10.10.10.0/24, then the ingress traffic uses the External EPG pcTag. If ingress traffic on the Layer 3 outside network has the matching default prefix 0.0.0.0/0, then the ingress traffic uses the VRF pcTag.

- As a result of these described behaviors, the following use cases are possible if the same VRF and
same leaf switch are configured with L3extOut-A and L3extOut-B using an SVI interface:

  Case 1 is for L3extOut-A: This External Network EPG has two subnets defined: 10.10.10.0/24 &
  0.0.0.0/1. If ingress traffic on L3extOut-A has the matching prefix 10.10.10.0/24, it uses the external
  EPG pcTag & contract-A which is associated with L3extOut-A. When egress traffic on L3extOut-A
  has no specific match found, but there is a maximum prefix match with 0.0.0.0/1, it uses the External
  Bridge Domain (BD) pcTag & contract-A.

  Case 2 is for L3extOut-B: This External Network EPG has one subnet defined: 0.0.0.0/0. When
  ingress traffic on L3extOut-B has the matching prefix 10.10.10.0/24 (which is defined
  underL3extOut-A), it uses the External EPG pcTag of L3extOut-A and the contract-A which is tied
  with L3extOut-A. It does not use contract-B which is tied with L3extOut-B.

- Traffic not permitted: Traffic is not permitted when an invalid configuration sets the scope of the external
  subnet to shared route control (shared-rtctrl) as a subset of a subnet that is set to shared security
  (shared-security). For example, the following configuration is invalid:

  • shared rtctrl: 10.1.1.0/24, 10.1.2.0/24
  • shared security: 10.1.0.0/16

In this case, ingress traffic on a non-border leaf with a destination IP of 10.1.1.1 is dropped, since prefixes
10.1.1.0/24 and 10.1.2.0/24 are installed with a drop rule. Traffic is not permitted. Such traffic can be
enabled by revising the configuration to use the shared-rtctrl prefixes as shared-security prefixes as
well.

- Inadvertent traffic flow: Prevent inadvertent traffic flow by avoiding the following configuration scenarios:

  • **Case 1** configuration details:
    
    • A Layer 3 outside network configuration (for example, named L3extOut-1) with VRF1 is called
      provider1.
    
    • A second Layer 3 outside network configuration (for example, named L3extOut-2) with VRF2
      is called provider2.
    
    • L3extOut-1 VRF1 advertises a default route to the Internet, 0.0.0.0/0 which enables both
      shared-rtctrl and shared-security.
    
    • L3extOut-2 VRF2 advertises specific subnets to DNS and NTP, 192.0.0.0/8 which enables
      shared-rtctrl.
Layer 3 Out to Layer 3 Out Inter-VRF Leaking

Starting with Cisco APIC release 2.2(2e), when there are two Layer 3 Outs in two different VRFs, inter-VRF leaking is supported.

For this feature to work, the following conditions must be satisfied:

• A contract between the two Layer 3 Outs is required.

Layer 3 Out to Layer 3 Out Inter-VRF Leaking

Starting with Cisco APIC release 2.2(2e), when there are two Layer 3 Outs in two different VRFs, inter-VRF leaking is supported.

For this feature to work, the following conditions must be satisfied:

• A contract between the two Layer 3 Outs is required.
• Routes of connected and transit subnets for a Layer 3 Out are leaked by enforcing contracts (L3Out-L3Out as well as L3Out-EPG) and without leaking the dynamic or static routes between VRFs.

• Dynamic or static routes are leaked for a Layer 3 Out by enforcing contracts (L3Out-L3Out as well as L3Out-EPG) and without advertising directly connected or transit routes between VRFs.

• Shared Layer 3 Outs in different VRFs can communicate with each other.

• There is no associated L3Out required for the bridge domain. When an Inter-VRF shared L3Out is used, it is not necessary to associate the user tenant bridge domains with the L3Out in tenant common. If you had a tenant-specific L3Out, it would still be associated to your bridge domains in your respective tenants.

• Two Layer 3 Outs can be in two different VRFs, and they can successfully exchange routes.

• This enhancement is similar to the Application EPG to Layer 3 Out inter-VRF communications. The only difference is that instead of an Application EPG there is another Layer 3 Out. Therefore, in this case, the contract is between two Layer 3 Outs.

In the following figure, there are two Layer 3 Outs with a shared subnet. There is a contract between the Layer 3 external instance profile (I3extInstP) in both the VRFs. In this case, the Shared Layer 3 Out for VRF1 can communicate with the Shared Layer 3 Out for VRF2.

Figure 21: Shared Layer 3 Outs Communicating Between Two VRFs

Configuring Two Shared Layer 3 Outs in Two VRFs Using REST API

The following REST API configuration example that displays how two shared Layer 3 Outs in two VRFs communicate.

Procedure

Step 1 Configure the provider Layer 3 Out.

Example:
### Configuring Shared Layer 3 Out Inter-VRF Leaking Using the NX-OS Style CLI - Named Example

#### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>Enter the configure mode.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>\apic1# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td><strong>Configure the provider Layer 3 Out.</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>\apic1(config)# tenant t1_provider</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\apic1(config-tenant)# external-l3 epg</td>
<td></td>
</tr>
<tr>
<td></td>
<td>l3extInstP-1 l3out T0-o1-L3OUT-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\apic1(config-tenant-l3ext-epg)# vrf member VRF1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24 shared</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\apic1(config-tenant-l3ext-epg)# contract provider vzBrCP-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>\apic1(config-tenant-l3ext-epg)# exit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command or Action</td>
<td>Purpose</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td></td>
</tr>
<tr>
<td>apic1(config-tenant)# exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apic1(config)# leaf 101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# vrf context tenant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tl_provider vrf VRF1 l3out T0-o1-L3OUT-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf)# route-map</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T0-o1-L3OUT-1_shared</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# ip prefix-list l3extInstP-1 permit 192.168.2.0/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# match prefix-list l3extInstP-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map-match)# exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# exit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 3** Configure the consumer Layer 3 Out.

**Example:**

```plaintext
apic1(config)# tenant t1_consumer
apic1(config-tenant)# external-l3 epg l3extInstP-2 l3out T0-o1-L3OUT-1
apic1(config-tenant-l3ext-epg)# vrf member VRF2
apic1(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared
apic1(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported
apic1(config-tenant-l3ext-epg)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant tl_consumer vrf VRF2 l3out T0-o1-L3OUT-1
apic1(config-leaf-vrf)# route-map T0-o1-L3OUT-1_shared
apic1(config-leaf-vrf-route-map)# ip prefix-list l3extInstP-2 permit 199.16.2.0/24
apic1(config-leaf-vrf-route-map)# match prefix-list l3extInstP-2
apic1(config-leaf-vrf-route-map-match)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf)# exit
apic1(config)#
```
## Configuring Shared Layer 3 Out Inter-VRF Leaking Using the NX-OS Style CLI - Implicit Example

### Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** Enter the configure mode.  
Example:  
apic1# configure |         |
| **Step 2** Configure the provider tenant and VRF.  
Example:  
apic1(config)# tenant t1_provider  
apic1(config-tenant)# vrf context VRF1  
apic1(config-tenant-vrf)# exit  
apic1(config-tenant)# exit |         |
| **Step 3** Configure the consumer tenant and VRF.  
Example:  
apic1(config)# tenant t1_consumer  
apic1(config-tenant)# vrf context VRF2  
apic1(config-tenant-vrf)# exit  
apic1(config-tenant)# exit |         |
| **Step 4** Configure the contract.  
Example:  
apic1(config)# tenant t1_provider  
apic1(config-tenant)# contract vzBrCP-1  
type permit  
apic1(config-tenant-contract)# scope exportable  
apic1(config-tenant-contract)# export to tenant t1_consumer  
apic1(config-tenant-contract)# exit |         |
| **Step 5** Configure the provider External Layer 3 EPG.  
Example:  
apic1(config-tenant)# external-l3 epg l3extInstP-1  
apic1(config-tenant-l3ext-epg)# vrf member VRF1  
apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24 shared  
apic1(config-tenant-l3ext-epg)# contract provider vzBrCP-1  
apic1(config-tenant-l3ext-epg)# exit  
apic1(config-tenant)# exit |         |
| **Step 6** Configure the provider export map.  
Example: |         |
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>apic1(config)# leaf 101</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# vrf context tenant tl_provider vrf VRF1</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf)# route-map map1</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# ip prefix-list p1 permit 192.168.2.0/24</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# match prefix-list p1</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map-match)# exit</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# exit</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf)# export map map1</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 7</td>
<td>Configure the consumer external Layer 3 EPG.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1(config)# tenant tl_consumer</td>
<td></td>
</tr>
<tr>
<td>apic1(config-tenant)# external-l3 epg l3extInstP-2</td>
<td></td>
</tr>
<tr>
<td>apic1(config-tenant-l3ext-epg)# vrf member VRF2</td>
<td></td>
</tr>
<tr>
<td>apic1(config-tenant-l3ext-epg)# match ip 199.16.2.0/24 shared</td>
<td></td>
</tr>
<tr>
<td>apic1(config-tenant-l3ext-epg)# contract consumer vzBrCP-1 imported</td>
<td></td>
</tr>
<tr>
<td>apic1(config-tenant-l3ext-epg)# exit</td>
<td></td>
</tr>
<tr>
<td>apic1(config-tenant)# exit</td>
<td></td>
</tr>
<tr>
<td>Step 8</td>
<td>Configure the consumer export map.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1(config)# leaf 101</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# vrf context tenant tl_consumer vrf VRF2</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf)# route-map map2</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# ip prefix-list p2 permit 199.16.2.0/24</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# match prefix-list p2</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map-match)# exit</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf-route-map)# exit</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf-vrf)# export map map2</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# exit</td>
<td></td>
</tr>
<tr>
<td>apic1(config)#</td>
<td></td>
</tr>
</tbody>
</table>

Configuring Shared Layer 3 Out Inter-VRF Leaking Using the Advanced GUI

Before you begin
The contract label to be used by the consumer and provider is already created.
Procedure

Step 1  
On the menu bar, choose **Tenants > Add Tenant**.

Step 2  
In the **Create Tenant** dialog box, enter a tenant name for the provider.

Step 3  
In the **VRF Name** field, enter a VRF name for the provider.

Step 4  
In the **Navigation** pane, under the new tenant name, navigate to **External Routed Networks**.

Step 5  
In the **Work** pane canvas, drag the **L3 Out** icon to associate it with the new VRF that you created.

Step 6  
In the **Create Routed Outside** dialog box, perform the following actions:
   a) In the **Name** field, enter a name for the Layer 3 Routed Outside.
   b) Click **Next** to go to the **Step 2 > External EPG Networks** dialog box.
   c) Expand **External EPG networks**.

Step 7  
In the **Create External Network** dialog box, perform the following actions:
   a) In the **Name** field, enter the external network name.
   b) Expand **Subnet**, and in the **Create Subnet** dialog box, and in the **IP Address** field, enter the match IP address. Click **OK**.

Step 8  
In the **Navigation** pane, navigate to the **Layer 3 Outside_name > Networks > External_network_name** that you created.

Step 9  
In the **Work** pane, under **Properties** for the external network, verify that the resolved VRF is displayed in the **Resolved VRF** field.

Step 10  
Click the **Configured Subnet IP** address for external subnets to open the **Subnet** dialog box.

Step 11  
In the **Scope** field, check the desired check boxes, and then click **Submit**.
   In this scenario, check the check boxes for **Shared Route Control Subnet** and **Shared Security Import Subnet**.

Step 12  
Navigate to the **Layer 3 Outside** you created earlier.

Step 13  
In the **Provider Label** field, enter the provider name that was created as a pre-requisite to starting this task. Click **Submit**.

Step 14  
On the menu bar, click **Tenants > Add Tenant**.

Step 15  
In the **Create Tenant** dialog box, enter a tenant name for the Layer 3 Outside consumer.

Step 16  
In the **VRF name** field, enter a VRF name for the consumer.

Step 17  
In the **Navigation** pane, under the new tenant name, navigate to **External Routed Networks** for the consumer.

Step 18  
In the **Work** pane canvas, drag the **L3 Out** icon to associate it with the new VRF that you created.

Step 19  
In the **Create Routed Outside** dialog box, perform the following actions:
   a) In the **Name** field, from the drop-down menu, choose the VRF that was created for the consumer.
   b) In the **Consumer Label** field, enter the name for the consumer label.
   c) Click **Next** to go to the **Step 2 > External EPG Networks** dialog box.

Step 20  
Expand **EPG networks**, and in the **Create External Network** dialog box, perform the following actions:
   a) In the **Name** field, enter a name for the external network.
   b) Expand **Subnet**, and in the **Create Subnet** dialog box, and in the **IP Address** field, enter the match IP address. Click **OK**.
   c) In the **Scope** field, check the desired check boxes, and then click **OK**.
In this scenario, check the check boxes for Shared Route Control Subnet and Shared Security Import Subnet.

**Step 21**

In the Create External Network dialog box, click OK. In the Create Routed Outside dialog box, click Finish.

This completes the configuration of shared Layer 3 Outside Inter-VRF leaking.
Interleak of External Routes

This chapter contains the following sections:

• Overview, on page 179
• Configuring Interleak of External Routes Using the GUI, on page 179
• Configuring Interleak External Routes Using the NX-OS Style CLI, on page 180
• Configuring Interleak of External Routes Using the REST API, on page 181

Overview

This topic provides a typical example of how to configure an interleak of external routes such as OSPF when using Cisco APIC.

Interleak from OSPF has been available in earlier releases. The feature now enables the user to set attributes, such as community, preference, and metric for route leaking from OSPF to BGP.

Configuring Interleak of External Routes Using the GUI

The external routed network configured in the example can also be extended to support IPv4. Both IPv4 and IPv6 routes can be advertised to and learned from the external routed network.

Before you begin

• The tenant, VRF, and bridge domain are created.
• The external routed domain is created.

Procedure

**Step 1**

On the menu bar, click TENANTS.

**Step 2**

In the Navigation pane, expand the Tenant_name > Networking > External Routed Networks and perform the following actions:

a) Right-click External Routed Networks and click Create Routed Outside.

b) In the Create Routed Outside dialog box, in the Name field, enter a name for the routed outside.
c) In the VRF field, from the drop-down list, choose the appropriate VRF.
d) From the External Routed Domain drop-down list, choose the appropriate external routed domain.
e) Check the check box for the desired protocol.

The options for the purpose of this task are OSPF.
f) In the Route Profile for Interleak field, click Create Route Profile.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>In the Create Route Profile dialog box, in the Name field, enter a route profile name.</td>
</tr>
<tr>
<td>4</td>
<td>In the Type field, you must choose Match Routing Policy Only.</td>
</tr>
<tr>
<td>5</td>
<td>Click the + sign to open the Create Route Control Context dialog box and perform the following actions:</td>
</tr>
<tr>
<td>a)</td>
<td>Populate the Order and the Name fields as desired.</td>
</tr>
<tr>
<td>b)</td>
<td>In the Set Attribute field, click Create Action Rule Profile.</td>
</tr>
<tr>
<td>c)</td>
<td>In the Create Action Rule Profile dialog box, in the Name field, enter a name for the action rule profile.</td>
</tr>
<tr>
<td>d)</td>
<td>Choose the desired attributes, and related community, criteria, tags, and preferences. Click OK.</td>
</tr>
<tr>
<td>e)</td>
<td>In the Create Route Profile dialog box, click Submit.</td>
</tr>
</tbody>
</table>

Step 6 In the Create Routed Outside dialog box, expand the Nodes and Interfaces Protocol Profiles area.
Step 7 In the Create Node Profile dialog box, specify the node profile. Click OK.
Step 8 In the Create Routed Outside dialog box, click Next.
Step 9 In the External EPG Networks area, expand External EPG Networks.
Step 10 In the Create External Network dialog box, in the Name field, enter a name.
Step 11 Expand Subnet, and in the Create Subnet dialog box, in the IP address field, enter an IP address. Click OK.
Step 12 In the Create External Network dialog box, click OK.
Step 13 In the Create Routed Outside dialog box, click Finish.

The route profile for interleave is created and associated with the L3 Outside.

---

**Configuring Interleak External Routes Using the NX-OS Style CLI**

**Before you begin**

- The tenant, VRF, and bridge domain are created.
- The external routed domain is created.

**Procedure**

Configure the route redistribution route policy using the NX-OS CLI:

a) Create a route profile with tenant as the scope:

   **Example:**

   ```
   apic1(config-leaf)# template route-profile map_ospf tenant ExampleCorp
   ```
b) Configure the redistribute route profile under BGP for OSPF using one of the route profiles created in the previous step.

Example:

```apic
apic1(config-leaf)# set tag 100
apic1(config-leaf)# exit
```

Note: Note that the redistribute route map allows all routes and applies the route profile for the route-control actions. In the example, all OSPF learnt routes are redistributed into BGP with tag 100.

---

**Configuring Interleak of External Routes Using the REST API**

**Before you begin**

- The tenant, VRF, and bridge domain are created.
- The external routed domain is created.

**Procedure**

Configure an interleak of external routes:

Example:

```xml
<l3extOut descr="" enforceRtctrl="export" name="out1" ownerKey="" ownerTag=""
targetDscp="unspecified">
  <l3extLNodeP configIssues="" descr="" name="Lnodep1" ownerKey="" ownerTag=""
tag="yellow-green" targetDscp="unspecified">
    <l3extRsNodeL3OutAtt rtrId="1.2.3.4" rtrIdLoopBack="yes"
tDn="topology/pod-1/node-101"/>
  </l3extLNodeP>
</l3extOut>
```
<fvRsCustQosPol tnQosCustomPolName=""/>
<l3extSubnet aggregate="" descr="" ip="14.15.16.0/24" name=""
scope="export-rtctrl,import-security"/>
</l3extInstP>
<ospfExtP areaCost="1" areaCtrl="redistribute,summary" areaId="0.0.0.1" areaType="nssa"
descr=""/>
</l3extOut>
CHAPTER 15

Dataplane IP Learning per VRF

This chapter contains the following sections:

- Overview, on page 183
- Guidelines and Limitations for Dataplane IP Learning per VRF, on page 183
- Feature Interaction for Dataplane IP Learning per VRF, on page 184
- Configuring Dataplane IP Learning Using the GUI, on page 184
- Configuring Dataplane IP Learning Using the NX-OS-Style CLI, on page 185

Overview

Endpoint IP and MAC addresses are learned by the ACI fabric through common network methods such as ARP, GARP, and ND. ACI also uses an internal method that learns IP and MAC addresses through the dataplane.

Dataplane IP learning per VRF is unique to the ACI network much in the same way as endpoint learning. While endpoint learning is identified as both IP and MAC, dataplane IP learning is specific to IP addressing only in VRFs. In APIC, you can enable or disable dataplane IP learning at the VRF level.

Guidelines and Limitations for Dataplane IP Learning per VRF

Follow these guidelines and limitations when considering the effects of dataplane IP learning per VRF:

- When dataplane IP learning per VRF is disabled, all the remote IP address entries in the tenant VRF are removed. The local IP entries are aged out and, subsequently, will not be re-learned through the dataplane, but can still be learned from the control plane.
- When dataplane IP learning per VRF is disabled, already learned local IP endpoints are retained and require control plane refreshes to be kept alive (assuming IP aging is also enabled). Dataplane L3 traffic will not keep IP endpoints alive.
- For first-generation leaf switch-based ToRs, when dataplane IP learning per VRF is disabled, remote MAC addresses are not learned. Hardware Proxy mode on the corresponding BDs must be configured. Local inner MAC addresses from VXLAN packets on downlink are not learned whether data plane IP learning for the VRF is enabled or not.
- Remote MAC addresses are not learned in endpoint to endpoint ARP scenarios.
Feature Interaction for Dataplane IP Learning per VRF

This section provides information about the interaction of dataplane IP learning per VRF with other features.

- **Anycast**
  - Enabled: Local Anycast IP addresses can be learned from both the data and control planes.
  - Disabled: Local Anycast IP addresses are aged out but can be learned through the control plane and host tracking.
  - Remote IP addresses are not learned in Anycast regardless of how dataplane IP learning per VRF is configured.

- **Rogue Endpoint Detection**
  - Enabled: Rogue is generated and moves are detected as expected.
  - Disabled: Remote IP addresses are flushed and rogue IP addresses are aged out. Rogue IP address are not detected on local moves. The only moves that are detected are via control traffic. Bounce is learned via COOP but these are dropped once the bounce timer expires.

- **L4-L7 Virtual IP (VIP)**
  - Enabled: L4-L7 VIP functions as expected (endpoint IP learning for VIP is only through the control plane). Consider the following functional stream: (1) from client to load balancer (LB) (L3 traffic), (2) LB to server (L2 traffic), and (3) server to client (L3). Clients (IP endpoints) behind the EPG are learned through the data/control plane. The VIP is learned only through the control plane on the LB EPG. Even though it's through the control plane, the VIP is not learned on other EPGs.
  - Disabled:
    - Client to load balancer: No remote IP address learned for VIP. The remote IP address is cleared. It will use the spine-proxy. If the IP address of the VIP is learned, spine-proxy look-up will be successful, otherwise it will generate glean for the VIP and learn it through the control plane.
    - Load balancer to server: No effect. Only bridging between LB/Server is supported for DSR use case.
    - Server to client: The remote IP address for the client is cleared and the spine-proxy will be used. If the remote IP address for the client entry is deleted in the spine, it is re-learned through glean. For clients behind L3out, there is no L3 remote IP address.

Configuring Dataplane IP Learning Using the GUI

This section explains how to disable dataplane IP learning.

The following procedure assumes that you have already configured tenant and VRF.

**Procedure**

- **Step 1** Navigate to Tenants &gt; tenant_name &gt; Networking &gt; VRFs &gt; vrf_name.
- **Step 2** On the VRF - vrf_name work pane, click the Policy tab.
- **Step 3** Scroll to the bottom of the Policy work pane and locate IP Data-plane Learning.
Step 4  Click one of the following:
  - **Disabled** - Disables dataplane IP learning on the VRF.
  - **Enabled** - Enables dataplane IP learning on the VRF.

Step 5  Click Submit.

---

### Configuring Dataplane IP Learning Using the NX-OS-Style CLI

This section explains how to disable dataplane IP learning using the NX-OS-style CLI.

To disable dataplane IP learning for a specific VRF:

**Procedure**

**Step 1**  Enter the configuration mode.

**Example:**
```plaintext
apic1# config
```

**Step 2**  Enter the tenant mode for the specific tenant.

**Example:**
```plaintext
apic1(config)# tenant name
```

**Step 3**  Enter the VRF context mode.

**Example:**
```plaintext
apic1(config-tenant)# vrf context name
```

**Step 4**  Disable dataplane IP learning for the VRF.

**Example:**
```plaintext
apic1(config-tenant-vrf)# ipdataplanelearning disabled
```
Overview

The IP Aging policy tracks and ages unused IP addresses on an endpoint. Tracking is performed using the endpoint retention policy configured for the bridge domain to send ARP requests (for IPv4) and neighbor solicitations (for IPv6) at 75% of the local endpoint aging interval. When no response is received from an IP address, that IP address is aged out.

This document explains how to configure the IP Aging policy.

Configuring the IP Aging Policy Using the GUI

This section explains how to enable and disable the IP Aging policy.

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>From the menu bar, click the <strong>System</strong> tab.</td>
</tr>
<tr>
<td>Step 2</td>
<td>From the submenu bar, click <strong>System Settings</strong>.</td>
</tr>
<tr>
<td>Step 3</td>
<td>In the navigation pane, click <strong>Endpoint Controls</strong>.</td>
</tr>
<tr>
<td>Step 4</td>
<td>In the work pane, click <strong>Ip Aging</strong>. The <strong>IP Aging Policy</strong> appears with the <strong>Administrative State Disabled</strong> button selected.</td>
</tr>
<tr>
<td>Step 5</td>
<td>From the <strong>Administrative State</strong>, click one of the following options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Enabled</strong>—Enables IP aging</td>
</tr>
</tbody>
</table>
• **Disabled**—Disables IP aging

---

**What to do next**

To specify the interval used for tracking IP addresses on endpoints, create an End Point Retention policy by navigating to Tenants > tenant-name > Policies > Protocol, right-click End Point Retention, and choose Create End Point Retention Policy.

---

**Configuring the IP Aging Policy Using the NX-OS-Style CLI**

This section explains how to enable and disable the IP Aging policy using the CLI.

**Procedure**

**Step 1**  
To enable the IP aging policy:  
**Example:**  
```
ifc1(config)# endpoint ip aging
```

**Step 2**  
To disable the IP aging policy:  
**Example:**  
```
ifav9-ifc1(config)# no endpoint ip aging
```

---

**What to do next**

To specify the interval used for tracking IP addresses on endpoints, create an Endpoint Retention policy.

---

**Configuring IP Aging Using the REST API**

This section explains how to enable and disable the IP aging policy using the REST API.

**Procedure**

**Step 1**  
To enable the IP aging policy:  
**Example:**  
```
<epIpAgingP adminSt="enabled" descr="" dn="uni/infra/ipAgingP-default" name="default" ownerKey="" ownerTag=""/>
```

**Step 2**  
To disable the IP aging policy:  
**Example:**
What to do next

To specify the interval used for tracking IP addresses on endpoints, create an Endpoint Retention policy by sending a post with XML such as the following example:

```xml
<fvEpRetPol bounceAgeIntvl="630" bounceTrig="protocol" holdIntvl="350" lcOwn="local" localEpAgeIntvl="900" moveFreq="256" name="EndpointPoll" remoteEpAgeIntvl="350"/>
```
CHAPTER 17

IPv6 Neighbor Discovery

This chapter contains the following sections:

- Neighbor Discovery, on page 191
- Configuring IPv6 Neighbor Discovery on a Bridge Domain, on page 192
- Configuring IPv6 Neighbor Discovery on a Layer 3 Interface, on page 195
- Configuring IPv6 Neighbor Discovery Duplicate Address Detection, on page 200

Neighbor Discovery

The IPv6 Neighbor Discovery (ND) protocol is responsible for the address auto configuration of nodes, discovery of other nodes on the link, determining the link-layer addresses of other nodes, duplicate address detection, finding available routers and DNS servers, address prefix discovery, and maintaining reachability information about the paths to other active neighbor nodes.

ND-specific Neighbor Solicitation or Neighbor Advertisement (NS or NA) and Router Solicitation or Router Advertisement (RS or RA) packet types are supported on all ACI fabric Layer 3 interfaces, including physical, Layer 3 sub interface, and SVI (external and pervasive). Up to APIC release 3.1(1x), RS/RA packets are used for auto configuration for all Layer 3 interfaces but are only configurable for pervasive SVIs.

Starting with APIC release 3.1(2x), RS/RA packets are used for auto configuration and are configurable on Layer 3 interfaces including routed interface, Layer 3 sub interface, and SVI (external and pervasive).

ACI bridge domain ND always operates in flood mode; unicast mode is not supported.

The ACI fabric ND support includes the following:

- Interface policies (nd:IfPol) control ND timers and behavior for NS/NA messages.
- ND prefix policies (nd:PfxPol) control RA messages.
- Configuration of IPv6 subnets for ND (fv:Subnet).
- ND interface policies for external networks.
- Configurable ND subnets for external networks, and arbitrary subnet configurations for pervasive bridge domains are not supported.

Configuration options include the following:

- Adjacencies
• Configurable Static Adjacencies: (vrf, L3Iface, ipv6 address) --> mac address
• Dynamic Adjacencies: Learned via exchange of NS/NA packets

• Per Interface
  • Control of ND packets (NS/NA)
    • Neighbor Solicitation Interval
    • Neighbor Solicitation Retry count
  • Control of RA packets
    • Suppress RA
    • Suppress RA MTU
    • RA Interval, RA Interval minimum, Retransmit time

• Per Prefix (advertised in RAs) control
  • Lifetime, preferred lifetime
  • Prefix Control (auto configuration, on link)

• Neighbor Discovery Duplicate Address Detection (DAD)

Configuring IPv6 Neighbor Discovery on a Bridge Domain

Creating the Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the REST API

Procedure

Create a tenant, VRF, bridge domain with a neighbor discovery interface policy and a neighbor discovery prefix policy.

Example:

```xml
<vTenant descr="" dn="uni/tn-ExampleCorp" name="ExampleCorp" ownerKey="" ownerTag="">
  <ndIfPol name="NDPol001" ctrl="managed-cfg" descr="" hopLimit="64" mtu="1500"
nsIntvl="1000" nsRetries="3" ownerKey="" ownerTag="" raIntvl="600" raLifetime="1800"
reachableTime="0" retransTimer="0"/>
  <fvCtx descr="" knwMcastAct="permit" name="pvn1" ownerKey="" ownerTag=""
pcEnfPref="enforced">
    <fvBD descr="" mac="00:22:BD:F8:19:FF" multiDstPktAct="bd-flood" name="bd1"
ownerKey="" ownerTag="" unicastRoute="yes" unkMacUcastAct="proxy" unkMcastAct="flood">
      <fvRsBDToNdP tnNdIfPolName="NDPol001"/>
      <fvRsCtx tnFvCtxName="pvn1"/>
    </fvBD>
    <fvSubnet ctrl="nd" descr="" ip="34::1/64" name="" preferred="no" scope="private">

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
If you have a public subnet when you configure the routed outside, you must associate the bridge domain with the outside configuration.

---

**Configuring a Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the NX-OS Style CLI**

**Procedure**

**Step 1** Configure an IPv6 neighbor discovery interface policy and assign it to a bridge domain:

a) Create an IPv6 neighbor discovery interface policy:

**Example:**

```
apic1(config)# tenant ExampleCorp
apic1(config-tenant)# template ipv6 nd policy NDPol001
apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500
```

b) Create a VRF and bridge domain:

**Example:**

```
apic1(config-tenant)# vrf context pvn1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member pvn1
apic1(config-tenant-bd)# exit
```

c) Assign an IPv6 neighbor discovery policy to the bridge domain:

**Example:**

```
apic1(config-tenant)# interface bridge-domain bd1
apic1(config-tenant-interface)# ipv6 nd policy NDPol001
apic1(config-tenant-interface)# exit
```

**Step 2** Configure an IPv6 bridge domain subnet and neighbor discovery prefix policy on the subnet:

**Example:**

```
apic1(config-tenant)# interface bridge-domain bd1
```
Creating the Tenant, VRF, and Bridge Domain with IPv6 Neighbor Discovery on the Bridge Domain Using the GUI

This task shows how to create a tenant, a VRF, and a bridge domain (BD) within which two different types of Neighbor Discovery (ND) policies are created. They are ND interface policy and ND prefix policy. While ND interface policies are deployed under BDs, ND prefix policies are deployed for individual subnets. Each BD can have its own ND interface policy. The ND interface policy is deployed on all IPv6 interfaces by default. In Cisco APIC, there is already an ND interface default policy available to use. If desired, you can create a custom ND interface policy to use instead. The ND prefix policy is on a subnet level. Every BD can have multiple subnets, and each subnet can have a different ND prefix policy.

Procedure

Step 1
On the menu bar, click TENANT > Add Tenant.

Step 2
In the Create Tenant dialog box, perform the following tasks:
   a) in the Name field, enter a name.
   b) Click the Security Domains + icon to open the Create Security Domain dialog box.
   c) In the Name field, enter a name for the security domain. Click Submit.
   d) In the Create Tenant dialog box, check the check box for the security domain that you created, and click Submit.

Step 3
In the Navigation pane, expand Tenant-name > Networking. In the Work pane, drag the VRF icon to the canvas to open the Create VRF dialog box, and perform the following actions:
   a) In the Name field, enter a name.
   b) Click Submit to complete the VRF configuration.

Step 4
In the Networking area, drag the BD icon to the canvas while connecting it to the VRF icon. In the Create Bridge Domain dialog box that displays, perform the following actions:
   a) In the Name field, enter a name.
   b) Click the L3 Configurations tab, and expand Subnets to open the Create Subnet dialog box, enter the subnet mask in the Gateway IP field.

Step 5
In the Subnet Control field, ensure that the ND RA Prefix check box is checked.

Step 6
In the ND Prefix policy field drop-down list, click Create ND RA Prefix Policy.

Note
There is already a default policy available that will be deployed on all IPv6 interfaces. Alternatively, you can create an ND prefix policy to use as shown in this example. By default, the IPv6 gateway subnets are advertised as ND prefixes in the ND RA messages. A user can choose to not advertise the subnet in ND RA messages by un-checking the ND RA prefix check box.

Step 7
In the Create ND RA Prefix Policy dialog box, perform the following actions:
   a) In the Name field, enter the name for the prefix policy.
For a given subnet there can only be one prefix policy. It is possible for each subnet to have a different prefix policy, although subnets can use a common prefix policy.

b) In the Controller State field, check the desired check boxes.

c) In the Valid Prefix Lifetime field, choose the desired value for how long you want the prefix to be valid.

d) In the Preferred Prefix Lifetime field, choose a desired value. Click OK.

Note An ND prefix policy is created and attached to the specific subnet.

Step 8 In the ND policy field drop-down list, click Create ND Interface Policy and perform the following tasks:

a) In the Name field, enter a name for the policy.

b) Click Submit.

Step 9 Click OK to complete the bridge domain configuration.

Similarly you can create additional subnets with different prefix policies as required.

A subnet with an IPv6 address is created under the BD and an ND prefix policy has been associated with it.

---

**Configuring IPv6 Neighbor Discovery on a Layer 3 Interface**

**Guidelines and Limitations**

The following guidelines and limitations apply to Neighbor Discovery Router Advertisement (ND RA) Prefixes for Layer 3 Interfaces:

- An ND RA configuration applies only to IPv6 Prefixes. Any attempt to configure an ND policy on IPv4 Prefixes will fail to apply.

**Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the GUI**

Note The steps here show how to associate an IPv6 neighbor discovery interface policy with a Layer 3 interface. The specific example shows how to configure using the non-VPC interface.

**Before you begin**

- The tenant, VRF, BD are created.

- The L3Out is created under External Routed Networks.
Procedure

**Step 1** In the Navigation pane, navigate to the appropriate external routed network under the appropriate Tenant.

**Step 2** Under External Routed Networks, expand > Logical Node Profiles > Logical Node Profile_name > Logical Interface Profiles.

**Step 3** Double-click the appropriate Logical Interface Profile, and in the Work pane, click Policy > Routed Interfaces.

**Note** If you do not have a Logical Interface Profile created, you can create a profile here.

**Step 4** In the Routed Interface dialog box, perform the following actions:

a) In the ND RA Prefix field, check the check box to enable ND RA prefix for the interface. When enabled, the routed interface is available for auto configuration.

Also, the ND RA Prefix Policy field is displayed.

b) In the ND RA Prefix Policy field, from the drop-down list, choose the appropriate policy.

c) Choose other values on the screen as desired. Click Submit.

**Note** When you configure using a VPC interface, you must enable the ND RA prefix for both side A and side B as both are members in the VPC configuration. In the Work Pane, in the Logical Interface Profile screen, click the SVI tab. Under Properties, check the check boxes to enable the ND RA Prefix for both Side A and Side B. Choose the identical ND RA Prefix Policy for Side A and Side B.

---

### Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the REST API

**Procedure**

Configure an IPv6 neighbor discovery interface policy and associate it with a Layer 3 interface:

The following example displays the configuration in a non-VPC set up.

**Example:**

```xml
<vTenant dn="uni/tn-ExampleCorp" name="ExampleCorp">
<ndIfPol name="NDPol001" ctrl="managed-cfg" hopLimit="64" mtu="1500" nsIntvl="1000" nsRetries="3" raIntvl="600" raLifetime="1800" reachableTime="0" retransTimer="0"/>
<vCtx name="pvn1" pcEnfPref="enforced">
<extOut enforceRtctrl="export" name="13extOut001">
<extRsEctx tnFvCtxName="pvn1"/>
<extLNodeP name="1nodePO01">
<extRsNodeL3OutAtt rtrId="11.11.205.1" rtrIdLoopBack="yes" tDn="topology/pod-2/node-2011"/>
<extLIfP name="1IfP001">
<extRsPathL3OutAtt addr="2001:20:21:22::2/64" ifInstT="l3-port" llAddr="::"/>
</extLIfP>
</extLNodeP>
</extRsEctx>
</extOut>
</vCtx>
</vTenant>
```
For VPC ports, ndPfxP must be a child of l3extMember instead of l3extRsNodeL3OutAtt. The following code snippet shows the configuration in a VPC setup.

```
<l3extLNodeP name="lnodeP001">
  <l3extRsNodeL3OutAtt rtrId="11.11.205.1" rtrIdLoopBack="yes" tDn="topology/pod-2/node-2011"/>
  <l3extRsNodeL3OutAtt rtrId="12.12.205.1" rtrIdLoopBack="yes" tDn="topology/pod-2/node-2012"/>
  <l3extLIfP name="lifP002">
    <l3extRsPathL3OutAtt addr="0.0.0.0" encap="vlan-205" ifInstT="ext-svi" llAddr="::" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" tDn="topology/pod-2/protpaths-2011-2012/pathep-[vpc7]" >
      <l3extMember addr="2001:20:25:1::1/64" descr="" llAddr="::" name="" nameAlias="" side="A"><ndPfxP >
        <ndRsFfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </ndPfxP>
      </l3extMember>
      <l3extMember addr="2001:20:25:1::2/64" descr="" llAddr="::" name="" nameAlias="" side="B"> <ndPfxP >
        <ndRsFfxPToNdPfxPol tnNdPfxPolName="NDPfxPol001"/>
        </ndPfxP>
      </l3extMember>
    </l3extRsPathL3OutAtt>
  </l3extLIfP>
</l3extLNodeP>
```

**Note**

Configuring an IPv6 Neighbor Discovery Interface Policy with RA on a Layer 3 Interface Using the NX-OS Style CLI

This example configures an IPv6 neighbor discovery interface policy, and assigns it to a Layer 3 interface. Next, it configures an IPv6 Layer 3 Out interface, neighbor discovery prefix policy, and associates the neighbor discovery policy to the interface.
### IPv6 Neighbor Discovery

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code>  &lt;br&gt; Example: <code>apic1# configure</code></td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>tenant tenant_name</code>  &lt;br&gt; Example: <code>apic1(config)# tenant ExampleCorp</code>&lt;br&gt;<code>apic1(config-tenant)#</code></td>
<td>Creates a tenant and enters the tenant mode.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>template ipv6 nd policy policy_name</code>  &lt;br&gt; Example: <code>apic1(config-tenant)# template ipv6 nd policy NDPol001</code></td>
<td>Creates an IPv6 ND policy.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>ipv6 nd mtu mtu value</code>  &lt;br&gt; Example: <code>apic1(config-tenant-template-ipv6-nd)# ipv6 nd mtu 1500</code>&lt;br&gt;<code>apic1(config-tenant-template-ipv6)# exit</code>&lt;br&gt;<code>apic1(config-tenant-template)# exit</code>&lt;br&gt;<code>apic1(config-tenant)#</code></td>
<td>Assigns an MTU value to the IPv6 ND policy.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><code>vrf context VRF_name</code>  &lt;br&gt; Example: <code>apic1(config-tenant)# vrf context pvnl</code>&lt;br&gt;<code>apic1(config-tenant-vrf)# exit</code></td>
<td>Creates a VRF.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td><code>l3out VRF_name</code>  &lt;br&gt; Example: <code>apic1(config-tenant)# l3out l3extOut001</code></td>
<td>Creates a Layer 3 Out.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td><code>vrf member VRF_name</code>  &lt;br&gt; Example: <code>apic1(config-tenant-l3out)# vrf member pvnl</code></td>
<td>Associates the VRF with the Layer 3 Out.</td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>apic1(config-tenant-l3out)# exit</td>
<td><strong>Step 8</strong> Assigns the Layer 3 Out and the VRF to a Layer 3 interface.</td>
</tr>
<tr>
<td>apic1(config-tenant)# external-l3 epg instp l3out l3extOut001</td>
<td><strong>Step 9</strong> Enters the leaf switch mode.</td>
</tr>
<tr>
<td>apic1(config-tenant)# vrf member tenant ExampleCorp vrf pvn1 l3out l3extOut001</td>
<td><strong>Step 10</strong> Associates the VRF to the leaf switch.</td>
</tr>
<tr>
<td>apic1(config-leaf)# int eth 1/1</td>
<td><strong>Step 11</strong> Enters the interface mode.</td>
</tr>
<tr>
<td>apic1(config-leaf-if)# vrf member tenant ExampleCorp vrf pvn1 l3out l3extOut001</td>
<td><strong>Step 12</strong> Specifies the associated Tenant, VRF, Layer 3 Out in the interface.</td>
</tr>
<tr>
<td>apic1(config-leaf-if)# ipv6 address 2001:20:21:22::2/64 preferred</td>
<td><strong>Step 13</strong> Specifies the primary or preferred IPv6 address.</td>
</tr>
</tbody>
</table>
### Configuring IPv6 Neighbor Discovery Duplicate Address Detection

**About Neighbor Discovery Duplicate Address Detection**

Duplicate Address Detection (DAD) is a process that is used by Neighbor Discovery to detect the duplicated addresses in the network. By default, DAD is enabled for the link-local and global-subnet IPv6 addresses used on the ACI fabric leaf layer 3 interfaces. Optionally, you can disable the DAD process for a IPv6 global-subnet by configuring the knob through the REST API (using the `ipv6Dad="disabled"` setting) or through the GUI. Configure this knob when the same shared secondary address is required to be used across L3Outs on different border leaf switches to provide border leaf redundancy to the external connected devices. Disabling the DAD process in this case will avoid the situation where the DAD considers the same shared secondary address on multiple border leaf switches as duplicates. If you do not disable the DAD process in this case, the shared secondary address might enter into the DUPLICATE DAD state and become unusable.

**Configuring Neighbor Discovery Duplicate Address Detection Using the REST API**

**Procedure**

- **Step 1** Disable the Neighbor Discovery Duplicate Address Detection process for a subnet by changing the value of the `ipv6Dad` entry for that subnet to `disabled`. 

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 14</strong></td>
<td><strong>ipv6 nd prefix</strong> <code>2001:20:21:22::2/64 1000 1000</code>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>apic1(config-leaf-if)# ipv6 nd prefix 2001:20:21:22::2/64 1000 1000</code>&lt;br&gt;</td>
</tr>
<tr>
<td><strong>Step 15</strong></td>
<td><strong>inherit ipv6 nd NDPol001</strong>&lt;br&gt;<strong>Example:</strong>&lt;br&gt;<code>apic1(config-leaf-if)# inherit ipv6 nd NDPol001&lt;br&gt;apic1(config-leaf-if)# exit&lt;br&gt;apic1(config-leaf)# exit</code>&lt;br&gt;</td>
</tr>
</tbody>
</table>
The following example shows how to set the Neighbor Discovery Duplicate Address Detection entry for the 2001:DB8:A::11/64 subnet to disabled:

**Note** In the following REST API example, long single lines of text are broken up with the `\` character to improve readability.

**Example:**

```xml
<l3extRsPathL3OutAtt addr="2001:DB8:A::2/64" autostate="enabled" childAction="" descr="" encap="" encapScope="" ifInstT="" ipv6Dad="disabled" llAddr="" mac="" mtu="" rn="" status="" tDn=""/>
  <l3extIp addr="2001:DB8:A::11/64" childAction="" descr="" ipv6Dad="disabled" name="" nameAlias="" rn="" status=""/>
</l3extRsPathL3OutAtt>
</l3extLIfP>
</l3extLNodeP>
```

**Step 2** Enter the `show ipv6 int` command on the leaf switch to verify that the configuration was pushed out correctly to the leaf switch. For example:

```
swtb23-leaf5# show ipv6 int vrf icmpv6:v1
IPv6 Interface Status for VRF "icmpv6:v1":

vlan2, Interface status: protocol-up/link-up/admin-up, iod: 73
  if_mode: ext
  IPv6 address:
    2001:DB8:A::2/64 [VALID] [PREFERRED]
    2001:DB8:A::11/64 [VALID] [dad-disabled]
  IPv6 subnet: 2001:DB8:A::/64
  IPv6 link-local address: fe80::863d:c6ff:fe9f:eb8b/10 (Default) [VALID]
```

---

**Configuring Neighbor Discovery Duplicate Address Detection Using the GUI**

Use the procedures in this section to disable the Neighbor Discovery Duplicate Address Detection process for a subnet.

**Procedure**

**Step 1** Navigate to the appropriate page to access the DAD field for that interface. For example:

a) Navigate to Tenants > Tenant > Networking > External Routed Networks > L3Out > Logical Node Profiles > node > Logical Interface Profiles, then select the interface that you want to configure.

b) Click on Routed Sub-interfaces or SVI, then click on the Create (+) button to configure that interface.

**Step 2** For this interface, make the following settings for the DAD entries:
- For the primary address, set the value for the DAD entry to **enabled**.
- For the shared secondary address, set the value for the DAD entry to **disabled**. Note that if the secondary address is not shared across border leaf switches, then you do not need to disable the DAD for that address.

**Example:**
For example, if you were configuring this setting for the SVI interface, you would:
- Set the Side A IPv6 DAD to **enabled**.
- Set the Side B IPv6 DAD to **disabled**.

**Example:**
As another example, if you were configuring this setting for the routed sub-interface interface, you would:
- In the main Select Routed Sub-Interface page, set the value for IPv6 DAD for the routed sub-interface to **enabled**.
- Click on the Create (+) button on the IPv4 Secondary/IPv6 Additional Addresses area to access the Create Secondary IP Address page, then set the value for IPv6 DAD to **disabled**. Then click on the OK button to apply the changes in this screen.

**Step 3**
Click on the Submit button to apply your changes.

**Step 4**
Enter the **show ipv6 int** command on the leaf switch to verify that the configuration was pushed out correctly to the leaf switch. For example:

```
swtb23-leaf5# show ipv6 int vrf icmpv6:v1
IPv6 Interface Status for VRF "icmpv6:v1" (9)
vlan2, Interface status: protocol-up/link-up/admin-up, iod: 73
  if_mode: ext
  IPv6 address:
    2001:DB8:A::2/64 [VALID] [PREFERRED]
    2001:DB8:A::11/64 [VALID] [dad-disabled]
  IPv6 subnet: 2001:DB8:A::/64
  IPv6 link-local address: fe80::863d:c6ff:fe9f:eb8b/10 (Default) [VALID]
```
CHAPTER 18

IP Multicast

This chapter contains the following sections:

• Layer 3 Multicast, on page 203
• About the Fabric Interface, on page 204
• Enabling Multicast Routing, on page 205
• Allocating VRF GIPO, on page 205
• Multiple Border Leaf Switches as Designated Forwarder, on page 206
• PIM Designated Router Election, on page 207
• Non-Border Leaf Switch Behavior, on page 207
• Active Border Leaf Switch List, on page 208
• Overload Behavior On Bootup, on page 208
• First-Hop Functionality, on page 208
• The Last-Hop, on page 208
• Fast-Convergence Mode, on page 208
• About Rendezvous Points, on page 209
• About Inter-VRF Multicast, on page 209
• Guidelines and Restrictions for Configuring Layer 3 Multicast, on page 210
• Configuring Layer 3 Multicast Using the GUI, on page 212
• Configuring Layer 3 Multicast Using the NX-OS Style CLI, on page 214
• Configuring Layer 3 Multicast Using REST API, on page 216

Layer 3 Multicast

In the ACI fabric, most unicast and multicast routing operate together on the same border leaf switches, with the multicast protocol operating over the unicast routing protocols.

In this architecture, only the border leaf switches run the full Protocol Independent Multicast (PIM) protocol. Non-border leaf switches run PIM in a passive mode on the interfaces. They do not peer with any other PIM routers. The border leaf switches peer with other PIM routers connected to them over L3 Outs and also with each other.

The following figure shows the border leaf (BL) switches (BL1 and BL2) connecting to routers (R1 and R2) in the multicast cloud. Each virtual routing and forwarding (VRF) in the fabric that requires multicast routing will peer separately with external multicast routers.
About the Fabric Interface

The fabric interface is a virtual interface between software modules and represents the fabric for multicast routing. The interface takes the form of a tunnel interface with the tunnel destination being the VRF GIPo (Group IP outer address). For example, if a border leaf is the designated forwarder responsible for forwarding traffic for a group, then the fabric interface would be in the outgoing interface (OIF) list for the group. There is no equivalent for the interface in hardware. The operational state of the fabric interface should follow the aggFabState published by the intermediate system-to-intermediate system (IS-IS).

Note

The user must configure a unique loopback address on each border leaf on each VRF that is enables multicast routing.

Any loopback configured for unicast routing can be reused. This loopback address must be routed from the external network and will be injected into the fabric MPBGP (Multiprotocol Border Gateway Protocol) routes.

---

1 The GIPo (Group IP outer address) is the destination multicast address used in the outer IP header of the VXLAN packet for all multi-destination packets (Broadcast, Unknown unicast, and Multicast) packets forwarded within the fabric.
Enabling Multicast Routing

Multicast is enabled or disabled at three levels, the VRF, L3 Out, and the bridge domain (BD). At the top level, multicast routing must be enabled on the VRF that has any multicast-enabled BDs. On a multicast-enabled VRF, there can be a combination of multicast routing-enabled BDs and BDs where multicast routing is disabled. BD with multicast-routing disabled will not show on VRF multicast panel. L3 Out with multicast routing-enabled will show up on the panel as well, but any BD that has multicast routing-enabled will always be a part of a VRF that has multicast routing-enabled.

Multicast Routing is not supported on the leaf switches such as Cisco Nexus 93128TX, 9396PX, and 9396TX. All the multicast routing and any multicast-enabled VRF should be deployed only on the switches with -EX and -FX in their product IDs. For example:

- 93108TC-EX
- 93180YC-EX
- 93108TC-FX
- 93180YC-FX

Layer 3 Out ports and sub-interfaces are supported while external SVIs are not supported. Since external SVIs are not supported, PIM cannot be enabled in L3-VPC.

Allocating VRF GIPo

VRF GIPo is allocated implicitly based on configuration. There will be one GIPo for the VRF and one GIPo for every BD under that VRF. Additionally, any given GIPo might be shared between multiple BDs or multiple
VRFs, but not a combination of VRFs and BDs. APIC will be required to ascertain this. In order to handle the VRF GIPo in addition to the BD GIPos already handled and build GIPo trees for them, IS-IS is modified. All multicast traffic for PIM enabled BDs will be forwarded using the VRF GIPo. This includes both Layer 2 and Layer 3 IP multicast. Any broadcast or unicast flood traffic on the multicast enabled BDs will continue to use the BD GIPo. Non-IP multicast enabled BDs will use the BD GIPo for all multicast, broadcast, and unicast flood traffic.

The APIC GUI will display a GIPo multicast address for all BDs and VRFs. The address displayed is always a /28 network address (the last four bits are zero). When the VXLAN packet is sent in the fabric, the destination multicast GIPo address will be an address within this /28 block and is used to select one of 16 FTAG trees. This achieves load balancing of multicast traffic across the fabric.

**Table 6: GIPo Usage**

<table>
<thead>
<tr>
<th>Traffic</th>
<th>Non-MC Routing-enabled BD</th>
<th>MC Routing-enabled BD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast</td>
<td>BD GIPo</td>
<td>BD GIPo</td>
</tr>
<tr>
<td>Unknown Unicast Flood</td>
<td>BD GIPo</td>
<td>BD GIPo</td>
</tr>
<tr>
<td>Multicast</td>
<td>BD GIPo</td>
<td>VRF GIPo</td>
</tr>
</tbody>
</table>

**Multiple Border Leaf Switches as Designated Forwarder**

When there are multiple border leaf (BL) switches in the fabric doing multicast routing, only one of the border leaves is selected as the designated forwarder for attracting traffic from the external multicast network and forwarding it to the fabric. This prevents multiple copies of the traffic and it balances the load across the multiple BL switches.

This is done by striping ownership for groups across the available BL switches, as a function of the group address and the VRF virtual network ID (VNID). A BL that is responsible for a group sends PIM joins to the external network to attract traffic into the fabric on behalf of receivers in the fabric.

Each BL in the fabric has a view of all the other active BL switches in the fabric in that VRF. So each of the BL switches can independently stripe the groups consistently. Each BL monitors PIM neighbor relations on the fabric interface to derive the list of active BL switches. When a BL switch is removed or discovered, the groups are re-striped across the remaining active BL switches. The striping is similar to the method used for hashing the GIPos to external links in multi-pod deployment, so that the group-to-BL mapping is sticky and results in fewer changes on up or down.
PIM Designated Router Election

For Layer 3 multicast on ACI fabric, the PIM DR (designated router) mechanism for different interface types is as follows:

- PIM-enabled L3 Out interfaces: Follows standard PIM DR mechanism on these interface types.
- Fabric interface: DR election on this interface is not of much significance as the DR functionality is determined by the striping. PIM DR election continues unaltered on this interface.
- Multicast routing-enabled Pervasive BDs: The pervasive BDs in the fabric are all stubs as far as multicast routing is concerned. Hence, on all the leaf switches, the SVI interfaces for pervasive BDs including vPC, are considered DR on the segment.

Non-Border Leaf Switch Behavior

On the non-border leaf switches, PIM runs in passive mode on the fabric interface and on the pervasive BD SVIs. PIM is in a new passive-probe mode where it sends only hellos. PIM neighbors are not expected on these pervasive BD SVIs. It is desirable to raise a fault when a PIM hello is heard from a router on a pervasive BD. PIM, on the non-border leaf switches, does not send any PIM protocol packets except for hellos on pervasive BDs and source register packets on the fabric interface.

At the same time, PIM will receive and process the following PIM packets on the fabric interface:

- **PIM Hellos**: This is used to track the active BL list on the fabric interface and on the pervasive BDs, this is used to raise faults.

- **PIM BSR, Auto-RP advertisements**: This is received on the fabric interface and is processed to glean the RP to group-range mapping.
Active Border Leaf Switch List

On every leaf switch, PIM maintains a list of active border leaf switches that is used for striping and other purposes. On the border leaf switches themselves this active border leaf list is derived from the active PIM neighbor relations. On non-border leaf switches, the list is generated by PIM using the monitored PIM Hello messages on the fabric interface. The source IP on the hello messages is the loopback IP assigned to each border leaf switch.

Overload Behavior On Bootup

When a border leaf switch gains connectivity to the fabric for the first time after bootup or after losing connectivity, it is not desirable to cause the border leaf switch to be part of the active border leaf switch list till the border leaf switch has had a chance to pull the COOP repo² information and to bring up its southbound protocol adjacencies. This can be achieved by delaying the transmission of PIM hello messages for a non-configured period of time.

First-Hop Functionality

The directly connected leaf will handle the first-hop functionality needed for PIM sparse mode.

The Last-Hop

The last-hop router is connected to the receiver and is responsible for doing a shortest-path tree (SPT) switchover in case of PIM any-source multicast (ASM). The border leaf switches will handle this functionality. The non-border leaf switches do not participate in this function.

Fast-Convergence Mode

The fabric supports a configurable fast-convergence mode where every border leaf switch with external connectivity towards the root (RP for (*,G) and source for (S, G)) pulls traffic from the external network. To prevent duplicates, only one of the BL switches forwards the traffic to the fabric. The BL that forwards the traffic for the group into the fabric is called the designated forwarder (DF) for the group. The stripe winner for the group decides on the DF. If the stripe winner has reachability to the root, then the stripe winner is also the DF. If the stripe winner does not have external connectivity to the root, then that BL chooses a DF by sending a PIM join over the fabric interface. All non-stripe winner BL switches with external reachability to the root send out PIM joins to attract traffic but continue to have the fabric interface as the RPF interface for the route. This results in the traffic reaching the BL switch on the external link, but getting dropped.

The advantage of the fast-convergence mode is that when there is a stripe owner change due to a loss of a BL switch for example, the only action needed is on the new stripe winner of programming the right Reverse Path Forwarding (RPF) interface. There is no latency incurred by joining the PIM tree from the new stripe winner. This comes at the cost of the additional bandwidth usage on the non-stripe winners' external links.

² All multicast group membership information is stored in the COOP database on the spines. When a border leaf boots up it pulls this information from the spine.
Fast-convergence mode can be disabled in deployments where the cost of additional bandwidth outweighs the convergence time saving.

### About Rendezvous Points

A rendezvous point (RP) is an IP address that you choose in a multicast network domain that acts as a shared root for a multicast shared tree. You can configure as many RPs as you like, and you can configure them to cover different group ranges.

You can configure two types of RPs:

- **Static RP**—Enables you to statically configure an RP for a multicast group range. To do so, you must configure the address of the RP on every router in the domain.

- **Fabric RP**—Enables you to configure an RP on all leafs where PIM is enabled on the VRF, which is necessary for supporting inter-VRF multicast (See About Inter-VRF Multicast, on page 209). When configured, external routers can use the fabric RP. However, an anycast RP cannot exist between the fabric and the external router.

  **Note** Fabric RP has the following restrictions:

  - Fabric RP does not support fast-convergence mode.
  - The fabric IP:
    - Must be unique across all the static RP entries within the static RP and fabric RP.
    - Cannot be one of the Layer 3 out router IDs

For information about configuring an RP, see the following sections:

- Configuring Layer 3 Multicast Using the GUI, on page 212
- Configuring Layer 3 Multicast Using the NX-OS Style CLI, on page 214
- Configuring Layer 3 Multicast Using REST API, on page 216

### About Inter-VRF Multicast

In typical data center with multicast networks, the multicast sources and receivers are in the same VRF, and all multicast traffic is forwarded within that VRF. There are use cases where the multicast sources and receivers may be located in different VRFs:

- Surveillance cameras are in one VRF while the people viewing the camera feeds are on computers in a different VRF.
• A multicast content provider is in one VRF while different departments of an organization are receiving the multicast content in different VRFs.

ACI release 4.0 adds support for inter-VRF multicast, which enables sources and receivers to be in different VRFs. This allows the receiver VRF to perform the reverse path forwarding (RPF) lookup for the multicast route in the source VRF. When a valid RPF interface is formed in the source VRF, this enables an outgoing interface (OIF) in the receiver VRF. All inter-VRF multicast traffic will be forwarded within the fabric in the source VRF. The inter-VRF forwarding and translation is performed on the leaf switch where the receivers are connected.

Note
• For any-source multicast, the RP used must be in the same VRF as the source.
• Source and receiver VRFs can be in an EPG or connected behind an L3 Out.

For ACI, inter-VRF multicast is configured per receiver VRF. Every NBL/BL that has the receiver VRF will get the same inter-VRF configuration. Each NBL that may have directly connected receivers, and BLs that may have external receivers, need to have the source VRF deployed. Control plane signaling and data plane forwarding will do the necessary translation and forwarding between the VRFs inside the NBL/BL that has receivers. Any packets forwarded in the fabric will be in the source VRF.

**Inter-VRF Multicast Requirements**

This section explains the inter-vrf multicast requirements.

• All sources for a particular group must be in the same VRF (the source VRF).
• Source VRF and source EPGs need to be present on all leafs where there are receiver VRFs.
• For ASM:
  • The RP must be in the same VRF as the sources (the source VRF).
  • The source VRF must be using fabric RP.
  • The same RP address configuration must be applied under the source and all receiver VRFs for the given group-range.

**Guidelines and Restrictions for Configuring Layer 3 Multicast**

See the following guidelines and restrictions:

• Custom QoS policy is not supported for Layer 3 multicast traffic sourced from outside the ACI fabric (received from L3Out).
• Enabling PIMv4 (Protocol-Independent Multicast, version 4) and Advertise Host routes on a BD is not supported.
• If the border leaf switches in your ACI fabric are running multicast and you disable multicast on the L3Out while you still have unicast reachability, you will experience traffic loss if the external peer is a Cisco Nexus 9000 switch. This impacts cases where traffic is destined towards the fabric (where the
sources are outside the fabric but the receivers are inside the fabric) or transiting through the fabric (where the source and receivers are outside the fabric, but the fabric is transit).

• If the (s, g) entry is installed on a border leaf switch, you might see drops in unicast traffic that comes from the fabric to this source outside the fabric when the following conditions are met:
  • Preferred group is used on the L3Out EPG
  • Unicast routing table for the source is using the default route 0.0.0.0/0

This behavior is expected.

• The Layer 3 multicast configuration is done at the VRF level so protocols function within the VRF and multicast is enabled in a VRF, and each multicast VRF can be turned on or off independently.

• Once a VRF is enabled for multicast, the individual bridge domains (BDs) and L3 Outs under the enabled VRF can be enabled for multicast configuration. By default, multicast is disabled in all BDs and Layer 3 Outs.

• Layer 3 multicast is not currently supported on VRFs that are configured with a shared L3 Out.

• Any Source Multicast (ASM) and Source-Specific Multicast (SSM) are supported.

• Bidirectional PIM and PIM IPv6 are currently not supported.

• IGMP snooping cannot be disabled on pervasive bridge domains with multicast routing enabled.

• Multicast routers are not supported in pervasive bridge domains.

• The Layer 3 multicast feature is supported on the following leaf switches:
  • EX models:
    • N9K-93108TC-EX
    • N9K-93180LC-EX
    • N9K-93180YC-EX
  • FX models:
    • N9K-93108TC-FX
    • N9K-93180YC-FX
    • N9K-C9348GC-FXP
  • FX2 models:
    • N9K-93240YC-FX2
    • N9K-C9336C-FX2

• PIM is supported on Layer 3 Out routed interfaces and routed subinterfaces including Layer 3 port-channel interfaces. PIM is not supported on Layer 3 Out SVI interfaces.

• Enabling PIM on an L3Out causes an implicit external network to be configured. This action results in the L3Out being deployed and protocols potentially coming up even if you have not defined an external network.
• For Layer 3 multicast support, when the ingress leaf switch receives a packet from a source that is attached on a bridge domain, and the bridge domain is enabled for multicast routing, the ingress leaf switch sends only a routed VRF copy to the fabric (routed implies that the TTL is decremented by 1, and the source-mac is rewritten with a pervasive subnet MAC). The egress leaf switch also routes the packet into receivers in all the relevant bridge domains. Therefore, if a receiver is on the same bridge domain as the source, but on a different leaf switch than the source, that receiver continues to get a routed copy, although it is in the same bridge domain. This also applies if the source and receiver are on the same bridge domain and on the same leaf switch, if PIM is enabled on this bridge domain.

For more information, see details about Layer 3 multicast support for multipod that leverages existing Layer 2 design, at the following link Adding Pods.

• Starting with Release 3.1(1x), Layer 3 multicast is supported with FEX. Multicast sources or receivers that are connected to FEX ports are supported. For further details about how to add FEX in your testbed, see Configure a Fabric Extender with Application Centric Infrastructure at this URL: https://www.cisco.com/c/en/us/support/docs/cloud-systems-management/application-policy-infrastructure-controller-apic/200529-Configure-a-Fabric-Extender-with-Applica.html. For releases preceding Release 3.1(1x), Layer 3 multicast is not supported with FEX. Multicast sources or receivers that are connected to FEX ports are not supported.

Note Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link. On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface.

For the appropriate MTU values for each platform, see the relevant configuration guides.

We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1.

Configuring Layer 3 Multicast Using the GUI

This section explains how to configure Layer 3 multicast using the Cisco APIC GUI.

Note Click the help icon (?) located in the top-right corner of the Work pane and of each dialog box for information about a visible tab or a field.

Before you begin

• The desired VRF, bridge domains, Layer 3 Out interfaces with IP addresses must be configured to enable PIM and IGMP.

• Basic unicast network must be configured.
Procedure

Step 1  Navigate to Tenants > Tenant_name > Networking > VRFs > VRF_name > Multicast.
In the Work pane, a message is displayed as follows: PIM is not enabled on this VRF. Would you like to enable PIM?

Step 2  Click YES, ENABLE MULTICAST.

Step 3  Configure interfaces:
   a) From the Work pane, click the Interfaces tab.
   b) Expand the Bridge Domains table to display the Create Bridge Domain dialog and enter the appropriate value in each field.
   c) Click Select.
   d) Expand the Interfaces table to display the Select an L3 Out dialog.
   e) Click the L3 Out drop-down arrow to choose an L3 Out.
   f) Click Select.

Step 4  Configure a rendezvous point (RP):
   a) In the Work pane, click the Rendezvous Points tab and choose from the following rendezvous point (RP) options:
      • Static RP
         a. Expand the Static RP table.
         b. Enter the appropriate value in each field.
         c. Click Update.
      • Fabric RP
         a. Expand the Fabric RP table.
         b. Enter the appropriate value in each field.
         c. Click Update.
      • Auto-RP
         a. Enter the appropriate value in each field.
      • Bootstrap Router (BSR)
         a. Enter the appropriate value in each field.

Step 5  Configure the pattern policy:
   a) From the Work pane, click the Pattern Policy tab and choose the Any Source Multicast (ASM) or Source Specific Multicast (SSM) option.
   b) Enter the appropriate value in each field.

Step 6  Configure the PIM settings:
   a) Click the PIM Setting tab.
   b) Enter the appropriate value in each field.
Step 7 Configure the IGMP settings:
   a) Click the **IGMP Setting** tab.
   b) Expand the **IGMP Context SSM Translate Policy** table.
   c) Enter appropriate value in each field.
   d) Click **Update**.

Step 8 Configure inter-VRF multicast:
   a) In the **Work** pane, click the **Inter-VRF Multicast** tab.
   b) Expand the **Inter-VRF Multicast** table.
   c) Enter appropriate value in each field.
   d) Click **Update**.

Step 9 When finished, click **Submit**.

Step 10 To verify the configuration perform the following actions:
   a) In the **Work** pane, click **Interfaces** to display the associated *Bridge Domains*.
   b) Click **Interfaces** to display the associated *L3 Out* interfaces.
   c) In the **Navigation** pane, navigate to the BD.
   d) In the **Work** pane, the configured IGMP policy and PIM functionality are displayed as configured earlier.
   e) In the **Navigation** pane, the L3 Out interface is displayed.
   f) In the **Work** pane, the PIM functionality is displayed as configured earlier.
   g) In the **Work** pane, navigate to **Fabric > Inventory > Protocols > IGMP** to view the operational status of the configured IGMP interfaces.
   h) In the **Work** pane, navigate to **Fabric > Inventory > Pod name > Leaf_Node > Protocols > IGMP > IGMP Domains** to view the domain information for multicast enabled/disabled nodes.

---

**Configuring Layer 3 Multicast Using the NX-OS Style CLI**

**Procedure**

**Step 1** Enter the configure mode.

**Example:**
```
apic1# configure
```

**Step 2** Enter the configure mode for a tenant, the configure mode for the VRF, and configure PIM options.

**Example:**
```
apic1(config)# tenant tenant1
apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ip pim
apic1(config-tenant-vrf)# ip pim fast-convergence
apic1(config-tenant-vrf)# ip pim bsr forward
```

**Step 3** Configure IGMP and the desired IGMP options for the VRF.

**Example:**
Step 4
Enter the L3 Out mode for the tenant, enable PIM, and enter the leaf interface mode. Then configure PIM for this interface.

Example:
```
apic1(config-tenant)# l3out tenant1_l3out
apic1(config-tenant-l3out)# ip pim
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
```

Step 5
Configure IGMP for the interface using the IGMP commands.

Example:
```
apic1(config-leaf-if)# ip igmp fast-leave
apic1(config-leaf-if)# ip igmp inherit interface-policy igmp_intpoll
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Step 6
Configure a fabric RP.

Example:
```
apic1(config)# tenant tenant1
apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ip pim fabric-rp-address 20.1.15.1 route-map intervrf-ctx2
apic1(config-tenant-vrf)# ip pim fabric-rp-address 20.1.15.2 route-map intervrf-ctx1
apic1(config-tenant-vrf)# exit
```

Step 7
Configure a inter-VRF multicast.

Example:
```
apic1(config-tenant)# vrf context tenant1_vrf
apic1(config-tenant-vrf)# ip pim inter-vrf-src ctx2 route-map intervrf-ctx2
apic1(config-tenant-vrf)# route-map intervrf-ctx2 permit 1
apic1(config-tenant-vrf)# match ip multicast group 226.20.0.0/24
apic1(config-tenant-vrf)# exit
apic1(config)#
```

This completes the APIC Layer 3 multicast configuration.
Configuring Layer 3 Multicast Using REST API

Procedure

Step 1  Configure a tenant and VRF and enable multicast on a VRF.

Example:

```xml
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
  <fvCtx knmWcastAct="permit" name="ctx1">
    <pimCtxP mtu="1500">
      </pimCtxP>
  </fvCtx>
</fvTenant>
```

Step 2  Configure L3 Out and enable multicast (PIM, IGMP) on the L3 Out.

Example:

```xml
<l3extOut enforceRtctrl="export" name="l3out-pim_l3out1">
  <l3extRsEctx tnFvCtxName="ctx1"/>
  <l3extLNodeP configIssues="" name="bLeaf-CTX1-101">
    <l3extRsNodeL3OutAtt rtrId="200.0.0.1" rtrIdLoopBack="yes" tDn="topology/pod-1/node-101"/>
    <l3extLIfP name="if-PIM_Tenant-CTX1" tag="yellow-green">
      <igmpIfP/>
      <pimIfP>
        <pimRsIfPol tDn="uni/tn-PIM_Tenant/pimifpol-pim_poll1"/>
      </pimIfP>
    </l3extLIfP>
  </l3extLNodeP>
  <l3extRsL3DomAtt tDn="uni/l3dom-l3outDom"/>
  <l3extInstP name="l3out-PIM_Tenant-CTX1-1topo"/>
</l3extOut>
```

Step 3  Configure a BD under the tenant and enable multicast and IGMP on the BD.

Example:

```xml
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
  <fvBD arpFlood="yes" mcastAllow="yes" multiDstPktAct="bd-flood" name="bd2" type="regular" unicastRoute="yes" unkMacUcastAct="flood" unkMcastAct="flood">
    <igmpIfP/>
    <fvRsBDToOut tnL3extOutName="l3out-pim_l3out1"/>
    <fvRsCtx tnFvCtxName="ctx1"/>
    <fvRsIgmpsn/>
  </fvBD>
</fvTenant>
```

Step 4  Configure an IGMP policy and assign it to the BD.

Example:

```xml
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
  <igmpIfPol grpTimeout="260" lastMbrCnt="2" lastMbrRespTime="1" name="igmp_pol" querierTimeout="255" queryIntvl="125" robustFac="2" rspIntvl="10" startQueryCnt="2"/>
</fvTenant>
```
Step 5  Configure a route map, PIM, and RP policy on the VRF.

Note  When configuring a fabric RP using the REST API, first configure a static RP.

Example:

Configuring a static RP:

```xml
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
  <pimRouteMapPol name="rootMap">
    <pimRouteMapEntry action="permit" grp="224.0.0.0/4" order="10" rp="0.0.0.0" src="0.0.0.0/0"/>
  </pimRouteMapPol>
  <fvCtx knwMcastAct="permit" name="ctx1">
    <pimCtxP ctrl="" mtu="1500">
      <pimStaticRPPol>
        <pimStaticRPEntryPol rpIp="131.1.1.2">
          <pimRPGrpRangePol>
            <rtdmcRsFilterToRtMapPol tDn="uni/tn-PIM_Tenant/rtmap-rootMap"/>
          </pimRPGrpRangePol>
        </pimStaticRPEntryPol>
      </pimStaticRPPol>
    </pimCtxP>
  </fvCtx>
</fvTenant>
```

Configuring a fabric RP:

```xml
<fvTenant name="t0">
  <pimRouteMapPol name="fabricrp-rtmap">
    <pimRouteMapEntry grp="226.20.0.0/24" order="1"/>
  </pimRouteMapPol>
  <fvCtx name="ctx1">
    <pimCtxP ctrl="">
      <pimFabricRPPol status="">
        <pimStaticRPEntryPol rpIp="6.6.6.6">
          <pimRPGrpRangePol>
            <rtdmcRsFilterToRtMapPol tDn="uni/tn-t0/rtmap-fabricrp-rtmap"/>
          </pimRPGrpRangePol>
        </pimStaticRPEntryPol>
      </pimFabricRPPol>
    </pimCtxP>
  </fvCtx>
</fvTenant>
```

Step 6  Configure a PIM interface policy and apply it on the L3 Out.

Example:

```xml
<fvTenant dn="uni/tn-PIM_Tenant" name="PIM_Tenant">
  <l3extOut enforceRtcrl="export" name="l3out-pim_l3out1" targetDscp="unspecified">
    <l3extRsEctx tnFvCtxName="ctx1"/>
    <l3extLNodeP name="bLeaf-CTX1-101"/>
  </l3extOut>
</fvTenant>
```
Step 7 Configure inter-VRF multicast.

Example:
```
<fvTenant name="t0">
    <pimRouteMapPol name="intervrf" status=""/>
    <pimRouteMapEntry grp="225.0.0.0/24" order="1" status=""/>
    <pimRouteMapEntry grp="226.0.0.0/24" order="2" status=""/>
    <pimRouteMapEntry grp="228.0.0.0/24" order="3" status="deleted"/>
</pimRouteMapPol>
    <fvCtx name="ctx1">
        <pimCtxP ctrl="">
            <pimInterVRFPol status="">
                <pimInterVRFEntryPol srcVrfDn="uni/tn-t0/ctx-stig_r_ctx">
                    <rtmcRsFilterToRtMapPol tDn="uni/tn-t0/rtmap-intervrf"/>
                </pimInterVRFEntryPol>
            </pimInterVRFPol>
        </pimCtxP>
    </fvCtx>
</fvTenant>
```
IGMP Snooping

This chapter contains the following sections:

- About Cisco APIC and IGMP Snooping, on page 219
- Configuring and Assigning an IGMP Snooping Policy, on page 223
- Enabling IGMP Snooping Static Port Groups, on page 227
- Enabling IGMP Snoop Access Groups, on page 231

About Cisco APIC and IGMP Snooping

How IGMP Snooping is Implemented in the ACI Fabric

We recommend that you do not disable IGMP snooping on bridge domains. If you disable IGMP snooping, you may see reduced multicast performance because of excessive false flooding within the bridge domain.

IGMP snooping software examines IP multicast traffic within a bridge domain to discover the ports where interested receivers reside. Using the port information, IGMP snooping can reduce bandwidth consumption in a multi-access bridge domain environment to avoid flooding the entire bridge domain. By default, IGMP snooping is enabled on the bridge domain.

This figure shows the IGMP routing functions and IGMP snooping functions both contained on an ACI leaf switch with connectivity to a host. The IGMP snooping feature snoops the IGMP membership reports, and leaves messages and forwards them only when necessary to the IGMP router function.
IGMP snooping operates upon IGMPv1, IGMPv2, and IGMPv3 control plane packets where Layer 3 control plane packets are intercepted and influence the Layer 2 forwarding behavior.

IGMP snooping has the following proprietary features:

- Source filtering that allows forwarding of multicast packets based on destination and source IP addresses
- Multicast forwarding based on IP addresses rather than the MAC address
- Multicast forwarding alternately based on the MAC address

The ACI fabric supports IGMP snooping only in proxy-reporting mode, in accordance with the guidelines provided in Section 2.1.1, "IGMP Forwarding Rules," in RFC 4541:

IGMP networks may also include devices that implement "proxy-reporting", in which reports received from downstream hosts are summarized and used to build internal membership states. Such proxy-reporting devices may use the all-zeros IP Source-Address when forwarding any summarized reports upstream. For this reason, IGMP membership reports received by the snooping switch must not be rejected because the source IP address is set to 0.0.0.0.

As a result, the ACI fabric will send IGMP reports with the source IP address of 0.0.0.0.
For more information about IGMP snooping, see RFC 4541.

**Virtualization Support**

You can define multiple virtual routing and forwarding (VRF) instances for IGMP snooping.

On leaf switches, you can use the `show` commands with a VRF argument to provide a context for the information displayed. The default VRF is used if no VRF argument is supplied.

**The APIC IGMP Snooping Function, IGMPv1, IGMPv2, and the Fast Leave Feature**

Both IGMPv1 and IGMPv2 support membership report suppression, which means that if two hosts on the same subnet want to receive multicast data for the same group, the host that receives a member report from the other host suppresses sending its report. Membership report suppression occurs for hosts that share a port.

If no more than one host is attached to each switch port, you can configure the fast leave feature in IGMPv2. The fast leave feature does not send last member query messages to hosts. As soon as APIC receives an IGMP leave message, the software stops forwarding multicast data to that port.

IGMPv1 does not provide an explicit IGMP leave message, so the APIC IGMP snooping function must rely on the membership message timeout to indicate that no hosts remain that want to receive multicast data for a particular group.

---

**Note**

The IGMP snooping function ignores the configuration of the last member query interval when you enable the fast leave feature because it does not check for remaining hosts.

---

**The APIC IGMP Snooping Function and IGMPv3**

The IGMPv3 snooping function in APIC supports full IGMPv3 snooping, which provides constrained flooding based on the (S, G) information in the IGMPv3 reports. This source-based filtering enables the device to constrain multicast traffic to a set of ports based on the source that sends traffic to the multicast group.

By default, the IGMP snooping function tracks hosts on each VLAN port in the bridge domain. The explicit tracking feature provides a fast leave mechanism. Because every IGMPv3 host sends membership reports, report suppression limits the amount of traffic that the device sends to other multicast-capable routers. When report suppression is enabled, and no IGMPv1 or IGMPv2 hosts requested the same group, the IGMP snooping function provides proxy reporting. The proxy feature builds the group state from membership reports from the downstream hosts and generates membership reports in response to queries from upstream queriers.

Even though the IGMPv3 membership reports provide a full accounting of group members in a bridge domain, when the last host leaves, the software sends a membership query. You can configure the parameter last member query interval. If no host responds before the timeout, the IGMP snooping function removes the group state.
Cisco APIC and the IGMP Snooping Querier Function

When PIM is not enabled on an interface because the multicast traffic does not need to be routed, you must configure an IGMP snooping querier function to send membership queries. In APIC, within the IGMP Snoop policy, you define the querier in a bridge domain that contains multicast sources and receivers but no other active querier.

Cisco ACI has by default, IGMP snooping and IGMP snooping querier enabled. Additionally, if the Bridge Domain subnet control has “querier IP” selected, then the leaf switch behaves as a querier and starts sending query packets. Querier on the ACI leaf switch must be enabled when the segments do not have an explicit multicast router (PIM is not enabled). On the Bridge Domain where the querier is configured, the IP address used must be from the same subnet where the multicast hosts are configured.

A unique IP address must be configured so as to easily reference the querier function. You must use a unique IP address for IGMP snooping querier configuration, so that it does not overlap with any host IP address or with the IP addresses of routers that are on the same segment. The SVI IP address must not be used as the querier IP address or it will result in issues with querier election. As an example, if the IP address used for IGMP snooping querier is also used for another router on the segment, then there will be issues with the IGMP querier election protocol. The IP address used for querier functionality must also not be used for other functions, such as HSRP or VRRP.

The IP address for the querier should not be a broadcast IP address, multicast IP address, or 0 (0.0.0.0).

When an IGMP snooping querier is enabled, it sends out periodic IGMP queries that trigger IGMP report messages from hosts that want to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

The IGMP snooping querier performs querier election as described in RFC 2236. Querier election occurs in the following configurations:

- When there are multiple switch queriers configured with the same subnet on the same VLAN on different switches.
- When the configured switch querier is in the same subnet as with other Layer 3 SVI queriers.

Guidelines and Limitations for the APIC IGMP Snooping Function

The APIC IGMP snooping has the following guidelines and limitations:

- Layer 3 IPv6 multicast routing is not supported.
- Layer 2 IPv6 multicast packets will be flooded on the incoming bridge domain.
Configuring and Assigning an IGMP Snooping Policy

Configuring and Assigning an IGMP Snooping Policy to a Bridge Domain in the Advanced GUI

To implement IGMP snooping functionality, you configure an IGMP Snooping policy then assign that policy to one or more bridge domains.

Configuring an IGMP Snooping Policy Using the GUI

Create an IGMP Snooping policy whose IGMP settings can be assigned to one or multiple bridge domains.

Procedure

Step 1
Click the Tenants tab and the name of the tenant on whose bridge domain you intend to configure IGMP snooping support.

Step 2
In the Navigation pane, click Networking > Protocol Policies > IGMP Snoop.

Step 3
Right-click IGMP Snoop and select Create IGMP Snoop Policy.

Step 4
In the Create IGMP Snoop Policy dialog, configure a policy as follows:

a) In the Name and Description fields, enter a policy name and optional description.

b) In the Admin State field, select Enabled or Disabled to enable or disable this entire policy.

c) Select or unselect Fast Leave to enable or disable IGMPV2 immediate dropping of queries through this policy.

d) Select or unselect Enable querier to enable or disable the IGMP querier activity through this policy.

Note
For this option to be effectively enabled, the Subnet Control: Querier IP setting must also be enabled in the subnets assigned to the bridge domains to which this policy is applied. The navigation path to the properties page on which this setting is located is Tenants > tenant_name > Networking > Bridge Domains > bridge_domain_name > Subnets > subnet_name.

e) Specify in seconds the Last Member Query Interval value for this policy.

IGMP uses this value when it receives an IGMPv2 Leave report. This means that at least one host wants to leave the group. After it receives the Leave report, it checks that the interface is not configured for IGMP Fast Leave and if not, it sends out an out-of-sequence query.

f) Specify in seconds the Query Interval value for this policy.

This value is used to define the amount of time the IGMP function will store a particular IGMP state if it does not hear any reports on the group.

g) Specify in seconds Query Response Interval value for this policy.

When a host receives the query packet, it starts counting to a random value, less that the maximum response time. When this timer expires, host replies with a report.

h) Specify the Start query Count value for this policy.
Number of queries sent at startup that are separated by the startup query interval. Values range from 1 to 10. The default is 2.

i) Specify in seconds a Start Query Interval for this policy.

By default, this interval is shorter than the query interval so that the software can establish the group state as quickly as possible. Values range from 1 to 18,000 seconds. The default is 31 seconds.

**Step 5**  
Click **Submit**.

---

The new IGMP Snoop policy is listed in the Protocol Policies - IGMP Snoop summary page.

**What to do next**

To put this policy into effect, assign it to any bridge domain.

---

### Assigning an IGMP Snooping Policy to a Bridge Domain Using the GUI

Assigning an IGMP Snooping policy to a bridge domain configures that bridge domain to use the IGMP Snooping properties specified in that policy.

**Before you begin**

- Configure a bridge domain for a tenant.
- Configure the IGMP Snooping policy that will be attached to the bridge domain.

#### Note

For the Enable Querier option on the assigned policy to be effectively enabled, the Subnet Control: Querier IP setting must also be enabled in the subnets assigned to the bridge domains to which this policy is applied. The navigation path to the properties page on which this setting is located is **Tenants > tenant_name > Networking > Bridge Domains > bridge_domain_name > Subnets > subnet_name**.

**Procedure**

**Step 1**  
Click the APIC Tenants tab and select the name of the tenant whose bridge domains you intend to configure with an IGMP Snoop policy.

**Step 2**  
In the APIC navigation pane, click Networking > Bridge Domains, then select the bridge domain to which you intend to apply your policy-specified IGMP Snoop configuration.

**Step 3**  
On the main Policy tab, scroll down to the IGMP Snoop Policy field and select the appropriate IGMP policy from the drop-down menu.

**Step 4**  
Click **Submit**.

The target bridge domain is now associated with the specified IGMP Snooping policy.
# Configuring and Assigning an IGMP Snooping Policy to a Bridge Domain using the NX-OS Style CLI

## Before you begin
- Create the tenant that will consume the IGMP Snooping policy.
- Create the bridge domain for the tenant, where you will attach the IGMP Snooping policy.

## Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | Create a snooping policy based on default values.  
**Example:**  
apic1(config-tenant)# template ip igmp snooping policy cookieCut1  
apic1(config-tenant-template-ip-igmp-snooping)# show run all  
# Command: show running -config all  
tenant foo template ip igmp snooping policy cookieCut1  
# Time: Thu Oct 13 18:26:03 2016  
tenant t_10  
template ip igmp snooping policy cookieCut1  
  ip igmp snooping  
  no ip igmp snooping fast-leave  
  ip igmp snooping  
last-member-query-interval 1  
  no ip igmp snooping querier  
  ip igmp snooping query-interval 125  
  ip igmp snooping query-max-response-time 10  
  ip igmp snooping  
  start-up-query-count 2  
  ip igmp snooping  
  start-up-query-interval 31  
  no description  
  exit  
  exit  
apic1(config-tenant-template-ip-igmp-snooping)# | The example NX-OS style CLI sequence:  
- Creates an IGMP Snooping policy named cookieCut1 with default values.  
- Displays the default IGMP Snooping values for the policy cookieCut1. |

| **Step 2** | Modify the snooping policy as necessary.  
**Example:**  
apic1(config-tenant-template-ip-igmp-snooping)# ip igmp snooping query-interval 300  
apic1(config-tenant-template-ip-igmp-snooping)# show run all  
# Command: show running -config all | The example NX-OS style CLI sequence:  
- Specifies a custom value for the query-interval value in the IGMP Snooping policy named cookieCut1.  
- Confirms the modified IGMP Snooping value for the policy cookieCut1. |
Configuring and Assigning an IGMP Snooping Policy to a Bridge Domain using the REST API

### Command or Action

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>tenant foo template ip igmp snooping policy cookieCut1</td>
</tr>
</tbody>
</table>

#Time: Thu Oct 13 18:26:03 2016

| tenant foo |
| template ip igmp snooping policy |
| cookieCut1 |

| ip igmp snooping |
| no ip igmp snooping fast-leave |
| ip igmp snooping last-member-query-interval 1 |

| no ip igmp snooping querier |
| ip igmp snooping query-interval 300 |

| ip igmp snooping query-max-response-time 10 |
| ip igmp snooping startup-query-count 2 |

| startup-query-interval 31 |
| no description |
| exit |

| exit |
| apic1(config-tenant-template-ip-igmp-snooping)# |

| apic1(config--tenant)# |

---

### Step 3

Assign the policy to a bridge domain.

**Example:**

```
apic1(config-tenant)# int bridge-domain bd3
apic1(config-tenant-interface)# ip igmp snooping policy cookieCut1
```

### The example NX-OS style CLI sequence:

- Navigates to bridge domain, BD3 for the query-interval value in the IGMP Snooping policy named cookieCut1.
- Assigns the IGMP Snooping policy with a modified IGMP Snooping value for the policy cookieCut1.

---

### What to do next

You can assign the IGMP Snooping policy to multiple bridge domains.

### Configuring and Assigning an IGMP Snooping Policy to a Bridge Domain using the REST API

**Procedure**

To configure an IGMP Snooping policy and assign it to a bridge domain, send a post with XML such as the following example:

**Example:**

```
https://apic-ip-address/api/node/mo/uni/.xml
<fvTenant name="mcast_tenant1">
<!-- Create an IGMP snooping template, and provide the options -->
```

<!-- Create an IGMP snooping template, and provide the options -->
<igmpSnoopPol name="igmp_snp_bd_21"
    adminSt="enabled"
    lastMbrIntvl="1"
    queryIntvl="125"
    rspIntvl="10"
    startQueryCnt="2"
    startQueryIntvl="31"
/><fvCtx name="ip_video"/>

<fvBD name="bd_21">
    <fvRsCtx tnFvCtxName="ip_video"/>
    <!-- Bind IGMP snooping to a BD -->
    <fvRsIgmpsn tnIgmpSnoopPolName="igmp_snp_bd_21"/>
</fvBD></fvTenant>

This example creates and configures the IGMP Snooping policy, igmp_snp_bd_12 with the following properties, and binds the IGMP policy, igmp_snp_bd_21, to bridge domain, bd_21:

- Administrative state is enabled
- Last Member Query Interval is the default 1 second.
- Query Interval is the default 125.
- Query Response interval is the default 10 seconds
- The Start Query Count is the default 2 messages
- The Start Query interval is 35 seconds.

---

**Enabling IGMP Snooping Static Port Groups**

**Enabling IGMP Snooping Static Port Groups**

IGMP static port grouping enables you to pre-provision ports, that were previously statically-assigned to an application EPG, to enable the switch ports to receive and process IGMP multicast traffic. This pre-provisioning prevents the join latency which normally occurs when the IGMP snooping stack learns ports dynamically.

Static group membership can be pre-provisioned only on static ports assigned to an application EPG.

Static group membership can be configured through the APIC GUI, CLI, and REST API interfaces.

**Prerequisite: Deploy EPGs to Static Ports**

Enabling IGMP snoop processing on ports requires as a prerequisite that the target ports be statically-assigned to associated EPGs.

Static deployment of ports can be configured through the APIC GUI, CLI, or REST API interfaces. For information, see the following topics in the *Cisco APIC Layer 2 Networking Configuration Guide*:

- *Deploying an EPG on a Specific Node or Port Using the GUI*
Enabling IGMP Snooping and Multicast on Static Ports Using the GUI

You can enable IGMP snooping and multicast on ports that have been statically assigned to an EPG. Afterwards you can create and assign access groups of users that are permitted or denied access to the IGMP snooping and multicast traffic enabled on those ports.

**Before you begin**

Before you begin to enable IGMP snooping and multicast for an EPG, complete the following tasks:

- Identify the interfaces to enable this function and statically assign them to that EPG

**Note**

For details on static port assignment, see *Deploying an EPG on a Specific Node or Port Using the GUI* in the *Cisco APIC Layer 2 Networking Configuration Guide*.

- Identify the IP addresses that you want to be recipients of IGMP snooping and multicast traffic.

**Procedure**

**Step 1**

Click **Tenant** > **tenant_name** > **Application Profiles** > **application_name** > **Application EPGs** > **epg_name** > **Static Ports**.

Navigating to this spot displays all the ports you have statically assigned to the target EPG.

**Step 2**

Click the port to which you intend to statically assign group members for IGMP snooping. This action displays the **Static Path** page.

**Step 3**

On the IGMP Snoop Static Group table, click + to add an IGMP Snoop Address Group entry.

Adding an IGMP Snoop Address Group entry associates the target static port with a specified multicast IP address and enables it to process the IGMP snoop traffic received at that address.

a) In the **Group Address** field, enter the multicast IP address to associate with his interface and this EPG.
b) In the **Source Address** field enter the IP address of the source to the multicast stream, if applicable.
c) Click **Submit**.

When configuration is complete, the target interface is enabled to process IGMP Snooping protocol traffic sent to its associated multicast IP address.

**Note**

You can repeat this step to associate additional multicast addresses with the target static port.

**Step 4**

Click **Submit**.
Enabling IGMP Snooping and Multicast on Static Ports in the NX-OS Style CLI

You can enable IGMP snooping and multicast on ports that have been statically assigned to an EPG. Then you can create and assign access groups of users that are permitted or denied access to the IGMP snooping and multicast traffic enabled on those ports.

The steps described in this task assume the pre-configuration of the following entities:

- Tenant: tenant_A
- Application: application_A
- EPG: epg_A
- Bridge Domain: bridge_domain_A
- vrf: vrf_A -- a member of bridge_domain_A
- VLAN Domain: vd_A (configured with a range of 300-310)
- Leaf switch: 101 and interface 1/10
  - The target interface 1/10 on switch 101 is associated with VLAN 305 and statically linked with tenant_A, application_A, epg_A
- Leaf switch: 101 and interface 1/11
  - The target interface 1/11 on switch 101 is associated with VLAN 309 and statically linked with tenant_A, application_A, epg_A

Before you begin

Before you begin to enable IGMP snooping and multicasting for an EPG, complete the following tasks.

- Identify the interfaces to enable this function and statically assign them to that EPG

  **Note**
  For details on static port assignment, see *Deploying an EPG on a Specific Port with APIC Using the NX-OS Style CLI* in the *Cisco APIC Layer 2 Networking Configuration Guide*.

- Identify the IP addresses that you want to be recipients of IGMP snooping multicast traffic.

Procedure

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>On the target interfaces enable IGMP snooping and layer 2 multicasting</td>
<td>The example sequences enable:</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>apic1# conf t</td>
<td>• IGMP snooping on the statically-linked target interface 1/10 and associates it with a multicast IP address, 225.1.1.1</td>
</tr>
<tr>
<td></td>
<td>apic1(config)# tenant tenant_A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>apic1(config-tenant)# application application_A</td>
<td>• IGMP snooping on the statically-linked target interface 1/11 and associates it with a multicast IP address, 227.1.1.1</td>
</tr>
<tr>
<td></td>
<td>apic1(config-tenant-app)# epg epg_A</td>
<td></td>
</tr>
</tbody>
</table>
Enabling IGMP Snooping and Multicast on Static Ports Using the REST API

You can enable IGMP snooping and multicast processing on ports that have been statically assigned to an EPG. You can create and assign access groups of users that are permitted or denied access to the IGMP snoop and multicast traffic enabled on those ports.

Procedure

To configure application EPGs with static ports, enable those ports to receive and process IGMP snooping and multicast traffic, and assign groups to access or be denied access to that traffic, send a post with XML such as the following example.

In the following example, IGMP snooping is enabled on leaf 102 interface 1/10 on VLAN 202. Multicast IP addresses 224.1.1.1 and 225.1.1.1 are associated with this port.

Example:

https://apic-ip-address/api/node/mo/uni.xml

```xml
<fvTenant name="tenant_A">
  <fvAp name="application">
    <fvAEPg name="epg_A">
      <fvRsPathAtt encap="vlan-202" instrImedcy="immediate" mode="regular" tDn="topology/pod-1/paths-102/pathep-[eth1/10]">
        <!-- IGMP snooping static group case -->
        <igmpSnoopStaticGroup group="224.1.1.1" source="0.0.0.0"/>
        <igmpSnoopStaticGroup group="225.1.1.1" source="2.2.2.2"/>
      </fvRsPathAtt>
    </fvAEPg>
  </fvAp>
</fvTenant>
```
Enabling IGMP Snoop Access Groups

Enabling IGMP Snoop Access Groups

An “access-group” is used to control what streams can be joined behind a given port.

An access-group configuration can be applied on interfaces that are statically assigned to an application EPG in order to ensure that the configuration can be applied on ports that will actually belong to the that EPG.

Only Route-map-based access groups are allowed.

IGMP snoop access groups can be configured through the APIC GUI, CLI, and REST API interfaces.

Enabling Group Access to IGMP Snooping and Multicast Using the GUI

After you enable IGMP snooping and multicasting on ports that have been statically assigned to an EPG, you can then create and assign access groups of users that are permitted or denied access to the IGMP snooping and multicast traffic enabled on those ports.

Before you begin

Before you enable access to IGMP snooping and multicasting for an EPG, Identify the interfaces to enable this function and statically assign them to that EPG.

Note

For details on static port assignment, see Deploying an EPG on a Specific Node or Port Using the GUI in the Cisco APIC Layer 2 Networking Configuration Guide.

Procedure

Step 1
Click Tenant > tenant_name > Application Profiles > application_name > Application EPGs > epg_name > Static Ports.

Navigating to this spot displays all the ports you have statically assigned to the target EPG.

Step 2
Click the port to which you intend to assign multicast group access, to display the Static Port Configuration page.

Step 3
Click Actions > Create IGMP Snoop Access Group to display the IGMP Snoop Access Group table.

Step 4
Locate the IGMP Snoop Access Group table and click + to add an access group entry.

Adding an IGMP Snoop Access Group entry creates a user group with access to this port, associates it with a multicast IP address, and permits or denies that group access to the IGMP snoop traffic received at that address.

a) select Create Route Map Policy to display the Create Route Map Policy window.
b) In the Name field assign the name of the group that you want to allow or deny multicast traffic.
c) In the Route Maps table click + to display the route map dialog.
d) In the Order field, if multiple access groups are being configured for this interface, select a number that reflects the order in which this access group will be permitted or denied access to the multicast traffic on this interface. Lower-numbered access groups are ordered before higher-numbered access groups.

e) In the Group IP field enter the multicast IP address whose traffic is to be allowed or blocked for this access group.

f) In the Source IP field, enter the IP address of the source if applicable.

g) In the Action field, choose Deny to deny access for the target group or Permit to allow access for the target group.

h) Click OK.

i) Click Submit.

When the configuration is complete, the configured IGMP snoop access group is assigned a multicast IP address through the target static port and permitted or denied access to the multicast streams that are received at that address.

Note
• You can repeat this step to configure and associate additional access groups with multicast IP addresses through the target static port.

• To review the settings for the configured access groups, click to the following location: Tenant > tenant_name > Networking > Protocol Policies > Route Maps > route_map_access_group_name.

Step 5 Click Submit.

Enabling Group Access to IGMP Snooping and Multicast using the NX-OS Style CLI

After you have enabled IGMP snooping and multicast on ports that have been statically assigned to an EPG, you can then create and assign access groups of users that are permitted or denied access to the IGMP snooping and multicast traffic enabled on those ports.

The steps described in this task assume the pre-configuration of the following entities:

• Tenant: tenant_A

• Application: application_A

• EPG: epg_A

• Bridge Domain: bridge_domain_A

• vrf: vrf_A -- a member of bridge_domain_A

• VLAN Domain: vd_A (configured with a range of 300-310)

• Leaf switch: 101 and interface 1/10

The target interface 1/10 on switch 101 is associated with VLAN 305 and statically linked with tenant_A, application_A, epg_A

• Leaf switch: 101 and interface 1/11
The target interface 1/11 on switch 101 is associated with VLAN 309 and statically linked with tenant_A, application_A, epg_A

For details on static port assignment, see Deploying an EPG on a Specific Port with APIC Using the NX-OS Style CLI in the Cisco APIC Layer 2 Networking Configuration Guide.

### Procedure

<table>
<thead>
<tr>
<th></th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1** | Define the route-map "access groups." | The example sequences configure:
| Example: | • Route-map-access group "foobroker" linked to multicast group 225.1.1.1/24, access permitted  | • Route-map-access group "foobroker" linked to multicast group 227.1.1.1/24, access denied |
| | `apic1(config)# tenant tenant_A; application application_A; epg epg_A` | |
| | `apic1(config-tenant)# route-map foobroker permit` | |
| | `apic1(config-tenant-rtmap)# match ip multicast group 225.1.1.1/24` | |
| | `apic1(config-tenant-rtmap)# exit` | |
| | `apic1(config-tenant)# route-map foobroker deny` | |
| | `apic1(config-tenant-rtmap)# match ip multicast group 227.1.1.1/24` | |
| | `apic1(config-tenant-rtmap)# exit` | |
| **Step 2** | Verify route map configurations. | The example sequences configure:
| Example: | • Route-map-access group "foobroker" connected through leaf switch 101, interface 1/10, and VLAN 305. |
| | `apic1(config-tenant)# show running-config tenant test route-map foobroker` | • Route-map-access group "newbroker" connected through leaf switch 101, interface 1/10, and VLAN 305. |
| | `# Command: show running-config tenant test route-map foobroker` | |
| | `# Time: Mon Aug 29 14:34:30 2016` | |
| | `tenant test` | |
| | `route-map foobroker permit 10` | |
| | `match ip multicast group 225.1.1.1/24` | |
| | `exit` | |
| | `route-map foobroker deny 20` | |
| | `match ip multicast group 227.1.1.1/24` | |
| | `exit` | |
| | `exit` | |
| **Step 3** | Specify the access group connection path. | The example sequences configure:
<p>| Example: | • Route-map-access group &quot;foobroker&quot; connected through leaf switch 101, interface 1/10, and VLAN 305. |
| | <code>apic1(config-tenant)# application application_A</code> | • Route-map-access group &quot;newbroker&quot; connected through leaf switch 101, interface 1/10, and VLAN 305. |
| | <code>apic1(config-tenant-app)# epg epg_A</code> | |
| | <code>apic1(config-tenant-app-epg)# ip igmp snooping access-group route-map foobroker</code> | |
| | <code>leaf 101 interface ethernet 1/10 vlan 305</code> | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>apic1(config-tenant-app-epg)# ip igmp snooping access-group route-map newBroker leaf 101 interface ethernet 1/10 vlan 305</td>
<td></td>
</tr>
</tbody>
</table>

**Step 4** Verify the access group connections.

**Example:**

```
apic1(config-tenant-app-epg)# show run
# Command: show running-config tenant
tenant tenant_A application application_A epg epg_A
# Time: Mon Aug 29 14:43:02 2016
 tenant tenant_A
   application application_A
     epg epg_A
     bridge-domain member
     bridge_domain_A
     ip igmp snooping access-group
     route-map fooBroker leaf 101 interface ethernet 1/10 vlan 305
     ip igmp snooping access-group
     route-map fooBroker leaf 101 interface ethernet 1/11 vlan 309
     ip igmp snooping access-group
     route-map newBroker leaf 101 interface ethernet 1/10 vlan 305
     ip igmp snooping static-group
     225.1.1.1 leaf 101 interface ethernet 1/10 vlan 305
     ip igmp snooping static-group
     225.1.1.1 leaf 101 interface ethernet 1/11 vlan 309
     exit
     exit
```

---

**Enabling Group Access to IGMP Snooping and Multicast using the REST API**

After you have enabled IGMP snooping and multicast on ports that have been statically assigned to an EPG, you can then create and assign access groups of users that are permitted or denied access to the IGMP snooping and multicast traffic enabled on those ports.

**Procedure**

To define the access group, **F23broker**, send a post with XML such as in the following example.

The example configures access group **F23broker**, associated with tenant_A, Rmap_A, application_A, epg_A, on leaf 102, interface 1/10, VLAN 202. By association with Rmap_A, the access group **F23broker** has access to multicast traffic received at multicast address 226.1.1.1/24 and is denied access to traffic received at multicast address 227.1.1.1/24.

**Example:**
Enabling Group Access to IGMP Snooping and Multicast using the REST API

<api/node/mo/uni/.xml>

<fvTenant name="tenant_A">
  <pimRouteMapPol name="Rmap_A">
    <pimRouteMapEntry action="permit" grp="226.1.1.1/24" order="10"/>
    <pimRouteMapEntry action="deny" grp="227.1.1.1/24" order="20"/>
  </pimRouteMapPol>
  <fvAp name="application_A">
    <fvAEPg name="epg_A">
      <fvRsPathAtt encap="vlan-202" instrImedcy="immediate" mode="regular"
                   tDn="topology/pod-1/paths-102/pathep-[eth1/10]">
        <igmpSnoopAccessGroup name="F23broker">
          <igmpRsSnoopAccessGroupFilterRMap
              tnPimRouteMapPolName="Rmap_A"/>
        </igmpSnoopAccessGroup>
      </fvRsPathAtt>
    </fvAEPg>
  </fvAp>
</fvTenant>
Enabling Group Access to IGMP Snooping and Multicast using the REST API
HSRP

about HSRP

HSRP is a first-hop redundancy protocol (FHRP) that allows a transparent failover of the first-hop IP router. HSRP provides first-hop routing redundancy for IP hosts on Ethernet networks configured with a default router IP address. You use HSRP in a group of routers for selecting an active router and a standby router. In a group of routers, the active router is the router that routes packets, and the standby router is the router that takes over when the active router fails or when preset conditions are met.

Many host implementations do not support any dynamic router discovery mechanisms but can be configured with a default router. Running a dynamic router discovery mechanism on every host is not practical for many reasons, including administrative overhead, processing overhead, and security issues. HSRP provides failover services to such hosts.

When you use HSRP, you configure the HSRP virtual IP address as the default router of the host (instead of the IP address of the actual router). The virtual IP address is an IPv4 or IPv6 address that is shared among a group of routers that run HSRP.

When you configure HSRP on a network segment, you provide a virtual MAC address and a virtual IP address for the HSRP group. You configure the same virtual address on each HSRP-enabled interface in the group. You also configure a unique IP address and MAC address on each interface that acts as the real address. HSRP selects one of these interfaces to be the active router. The active router receives and routes packets destined for the virtual MAC address of the group.

HSRP detects when the designated active router fails. At that point, a selected standby router assumes control of the virtual MAC and IP addresses of the HSRP group. HSRP also selects a new standby router at that time.
HSRP uses a priority designator to determine which HSRP-configured interface becomes the default active router. To configure an interface as the active router, you assign it with a priority that is higher than the priority of all the other HSRP-configured interfaces in the group. The default priority is 100, so if you configure just one interface with a higher priority, that interface becomes the default active router.

Interfaces that run HSRP send and receive multicast User Datagram Protocol (UDP)-based hello messages to detect a failure and to designate active and standby routers. When the active router fails to send a hello message within a configurable period of time, the standby router with the highest priority becomes the active router. The transition of packet forwarding functions between the active and standby router is completely transparent to all hosts on the network.

You can configure multiple HSRP groups on an interface. The virtual router does not physically exist but represents the common default router for interfaces that are configured to provide backup to each other. You do not need to configure the hosts on the LAN with the IP address of the active router. Instead, you configure them with the IP address of the virtual router (virtual IP address) as their default router. If the active router fails to send a hello message within the configurable period of time, the standby router takes over, responds to the virtual addresses, and becomes the active router, assuming the active router duties. From the host perspective, the virtual router remains the same.

---

**Note**

Packets received on a routed port destined for the HSRP virtual IP address terminate on the local router, regardless of whether that router is the active HSRP router or the standby HSRP router. This process includes ping and Telnet traffic. Packets received on a Layer 2 (VLAN) interface destined for the HSRP virtual IP address terminate on the active router.

---

**About Cisco APIC and HSRP**

HSRP in Cisco ACI is supported only on routed-interface or sub-interface. Therefore HSRP can only be configured under Layer 3 Out. Also there must be Layer 2 connectivity provided by external device(s) such as a Layer 2 switch between ACI leaf switches running HSRP because HSRP operates on leaf switches by exchanging Hello messages over external Layer 2 connections. An HSRP hello message does not pass through the spine switch.

The following is an example topology of an HSRP deployment in Cisco APIC.
HSRP Versions

Cisco APIC supports HSRP version 1 by default. You can configure an interface to use HSRP version 2.

HSRP version 2 has the following enhancements to HSRP version 1:

- Expands the group number range. HSRP version 1 supports group numbers from 0 to 255. HSRP version 2 supports group numbers from 0 to 4095.
- For IPv4, uses the IPv4 multicast address 224.0.0.102 or the IPv6 multicast address FF02::66 to send hello packets instead of the multicast address of 224.0.0.2, which is used by HSRP version 1.
- Uses the MAC address range from 0000.0C9F.F000 to 0000.0C9F.FFFF for IPv4 and 0005.73A0.0000 through 0005.73A0.0FFF for IPv6 addresses. HSRP version 1 uses the MAC address range 0000.0C07.AC00 to 0000.0C07.ACFF.

Guidelines and Limitations

Follow these guidelines and limitations:

- The HSRP state must be the same for both HSRP IPv4 and IPv6. The priority and preemption must be configured to result in the same state after failovers.
• Currently, only one IPv4 and one IPv6 group is supported on the same sub-interface in Cisco ACI. Even when dual stack is configured, Virtual MAC must be the same in IPv4 and IPv6 HSRP configurations.

• BFD IPv4 and IPv6 is supported when the network connecting the HSRP peers is a pure layer 2 network. You must configure a different router MAC address on the leaf switches. The BFD sessions become active only if you configure different MAC addresses in the leaf interfaces.

• Users must configure the same MAC address for IPv4 and IPv6 HSRP groups for dual stack configurations.

• HSRP VIP must be in the same subnet as the interface IP.

• It is recommended that you configure interface delay for HSRP configurations.

• HSRP is only supported on routed-interface or sub-interface. HSRP is not supported on VLAN interfaces and switched virtual interface (SVI). Therefore, no VPC support for HSRP is available.

• Object tracking on HSRP is not supported.

• HSRP Management Information Base (MIB) for SNMP is not supported.

• Multiple group optimization (MGO) is not supported with HSRP.

• ICMP IPv4 and IPv6 redirects are not supported.

• Cold Standby and Non-Stop Forwarding (NSF) are not supported because HSRP cannot be restarted in the Cisco ACI environment.

• There is no extended hold-down timer support as HSRP is supported only on leaf switches. HSRP is not supported on spine switches.

• HSRP version change is not supported in APIC. You must remove the configuration and reconfigure with the new version.

• HSRP version 2 does not inter-operate with HSRP version 1. An interface cannot operate both version 1 and version 2 because both versions are mutually exclusive. However, the different versions can be run on different physical interfaces of the same router.

• Route Segmentation is programmed in Cisco Nexus 93128TX, Cisco Nexus 9396PX, and Cisco Nexus 9396TX leaf switches when HSRP is active on the interface. Therefore, there is no DMAC=router MAC check conducted for route packets on the interface. This limitation does not apply for Cisco Nexus 93180LC-EX, Cisco Nexus 93180YC-EX, and Cisco Nexus 93108TC-EX leaf switches.

• HSRP configurations are not supported in the Basic GUI mode. The Basic GUI mode has been deprecated starting with APIC release 3.0(1).

• Fabric to Layer 3 Out traffic will always load balance across all the HSRP leaf switches, irrespective of their state. If HSRP leaf switches span multiple pods, the fabric to out traffic will always use leaf switches in the same pod.

• This limitation applies to some of the earlier Cisco Nexus 93128TX, Cisco Nexus 9396PX, and Cisco Nexus 9396TX switches. When using HSRP, the MAC address for one of the routed interfaces or routed sub-interfaces must be modified to prevent MAC address flapping on the Layer 2 external device. This is because Cisco APIC assigns the same MAC address (00:22:BD:F8:19:FF) to every logical interface under the interface logical profiles.
Default HSRP Settings

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1</td>
</tr>
<tr>
<td>Delay</td>
<td>0</td>
</tr>
<tr>
<td>Reload Delay</td>
<td>0</td>
</tr>
<tr>
<td>Interface Control</td>
<td>No Use-Burned-in Address (BIA)</td>
</tr>
<tr>
<td>Group ID</td>
<td>0</td>
</tr>
<tr>
<td>Group Af</td>
<td>IPv4</td>
</tr>
<tr>
<td>IP Obtain Mode</td>
<td>admin</td>
</tr>
<tr>
<td>Priority</td>
<td>100</td>
</tr>
<tr>
<td>Hello Interval</td>
<td>3000 msec</td>
</tr>
<tr>
<td>Hold Interval</td>
<td>10000 msec</td>
</tr>
<tr>
<td>Group Control</td>
<td>Preemption disabled</td>
</tr>
<tr>
<td>Preempt Delay</td>
<td>0</td>
</tr>
<tr>
<td>Authentication Type</td>
<td>Plain Text</td>
</tr>
<tr>
<td>Authentication Key Timeout</td>
<td>0</td>
</tr>
<tr>
<td>VMAC</td>
<td>Derived (from HSRP groupID)</td>
</tr>
</tbody>
</table>

Configuring HSRP Using the GUI

HSRP is enabled when the leaf switch is configured.

Before you begin

- The tenant and VRF configured.
- VLAN pools must be configured with the appropriate VLAN range defined and the appropriate Layer 3 domain created and attached to the VLAN pool.
- The Attach Entity Profile must also be associated with the Layer 3 domain.
- The interface profile for the leaf switches must be configured as required.
Procedure

Step 1  On the menu bar, click > Tenants > Tenant_name. In the Navigation pane, click Networking > External Routed Networks > External Routed Network_name > Logical Node Profiles > Logical Interface Profile. An HSRP interface profile will be created here.

Step 2  Choose a logical interface profile, and click Create HSRP Interface Profile.

Step 3  In the Create HSRP Interface Profile dialog box, perform the following actions:
   a) In the Version field, choose the desired version.
   b) In the HSRP Interface Policy field, from the drop-down, choose Create HSRP Interface Policy.
   c) In the Create HSRP Interface Policy dialog box, in the Name field, enter a name for the policy.
   d) In the Control field, choose the desired control.
   e) In the Delay field and the Reload Delay field, set the desired values. Click Submit.

   The HSRP interface policy is created and associated with the interface profile.

Step 4  In the Create HSRP Interface Profile dialog box, expand HSRP Interface Groups.

Step 5  In the Create HSRP Group Profile dialog box, perform the following actions:
   a) In the Name field, enter an HSRP interface group name.
   b) In the Group ID field, choose the appropriate ID.

   The values available depend upon whether HSRP version 1 or version 2 was chosen in the interface profile.

   c) In the IP field, enter an IP address.

   The IP address must be in the same subnet as the interface.

   d) In the MAC address field, enter a Mac address.

   e) In the Group Name field, enter a group name.

   This is the name used in the protocol by HSRP for the HSRP MGO feature.

   f) In the Group Type field, choose the desired type.

   g) In the IP Obtain Mode field, choose the desired mode.

   h) In the HSRP Group Policy field, from the drop-down list, choose Create HSRP Group Policy.

Step 6  In the Create HSRP Group Policy dialog box, perform the following actions:
   a) In the Name field, enter an HSRP group policy name.

   b) The Key or Password field is automatically populated.

   The default value for authentication type is simple, and the key is "cisco." This is selected by default when a user creates a new policy.

   c) In the Type field, choose the level of security desired.

   d) In the Priority field choose the priority to define the active router and the standby router.

   e) In the remaining fields, choose the desired values, and click Submit.

   The HSRP group policy is created.

   f) Create secondary virtual IPs by populating the Secondary Virtual IPs field.

   This can be used to enable HSRP on each sub-interface with secondary virtual IPs. The IP address that you provide here also must be in the subnet of the interface.

   g) Click OK.
Configuring HSRP in Cisco APIC Using Inline Parameters in NX-OS Style CLI

HSRP is enabled when the leaf switch is configured.

Before you begin
- The tenant and VRF configured.
- VLAN pools must be configured with the appropriate VLAN range defined and the appropriate Layer 3 domain created and attached to the VLAN pool.
- The Attach Entity Profile must also be associated with the Layer 3 domain.
- The interface profile for the leaf switches must be configured as required.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters configuration mode.</td>
</tr>
</tbody>
</table>
| **Step 2** Configure HSRP by creating inline parameters. **Example:**
  apic1(config)# leaf 101
  apic1(config-leaf)# interface ethernet 1/17
  apic1(config-leaf-if)# hsrp version 1
  apic1(config-leaf-if)# hsrp use-bia
  apic1(config-leaf-if)# hsrp delay minimum 30
  apic1(config-leaf-if)# hsrp delay reload 30
  apic1(config-leaf-if)# hsrp 10 ipv4
  apic1(config-leaf-if-hsrp)# ip 182.16.1.2
  apic1(config-leaf-if-hsrp)# ip 182.16.1.3 secondary
  apic1(config-leaf-if-hsrp)# ip 182.16.1.4 secondary
  apic1(config-leaf-if-hsrp)# mac-address 5000.1000.1060
  apic1(config-leaf-if-hsrp)# timers 5 18
  apic1(config-leaf-if-hsrp)# priority 100 |
Configuring HSRP in Cisco APIC Using Template and Policy in NX-OS Style CLI

HSRP is enabled when the leaf switch is configured.

Before you begin

- The tenant and VRF configured.
- VLAN pools must be configured with the appropriate VLAN range defined and the appropriate Layer 3 domain created and attached to the VLAN pool.
- The Attach Entity Profile must also be associated with the Layer 3 domain.
- The interface profile for the leaf switches must be configured as required.

Procedure

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td>configure</td>
<td>Enters configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1(config)# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td>Configure HSRP policy templates.</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>apic1(config)# leaf 101</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# template hsrp</td>
<td></td>
</tr>
<tr>
<td>interface-policy hsrp-intfPol1 tenant t9</td>
<td></td>
</tr>
<tr>
<td>apic1(config-template-hsrp-if-pol)# hsrp-use-bia</td>
<td></td>
</tr>
<tr>
<td>apic1(config-template-hsrp-if-pol)# hsrp</td>
<td></td>
</tr>
</tbody>
</table>
### Command or Action
`delay minimum 30
apic1(config-template-hsrp-if-pol)# hsrp
delay reload 30`  

`apic1(config)# leaf 101
apic1(config-leaf)# template hsrp
group-policy hsrp-groupPol1 tenant t9
apic1(config-template-hsrp-group-pol)# timers 5 18
apic1(config-template-hsrp-group-pol)# priority 100
apic1(config-template-hsrp-group-pol)# preempt
apic1(config-template-hsrp-group-pol)# preempt delay minimum 60
apic1(config-template-hsrp-group-pol)# preempt delay reload 60
apic1(config-template-hsrp-group-pol)# preempt delay sync 60`  

### Step 3
Use the configured policy templates

**Example:**

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/17
apic1(config-leaf-if)# hsrp version 1
apic1(config-leaf-if)# inherit hsrp
interface-policy hsrp-intfPol1
apic1(config-leaf-if)# hsrp 10 ipv4
apic1(config-leaf-if)# ip 182.16.1.2
apic1(config-leaf-if)# ip 182.16.1.3
apic1(config-leaf-if)# ip 182.16.1.4
apic1(config-leaf-if)# secondary
apic1(config-leaf-if)# mac-address 5000.1000.1060
apic1(config-leaf-if)# inherit hsrp
group-policy hsrp-groupPol1
```

---

### Configuring HSRP in APIC Using REST API

HSRP is enabled when the leaf switch is configured.

**Before you begin**

- The tenant and VRF must be configured.
- VLAN pools must be configured with the appropriate VLAN range defined and the appropriate Layer 3 domain created and attached to the VLAN pool.
- The Attach Entity Profile must also be associated with the Layer 3 domain.
The interface profile for the leaf switches must be configured as required.

Procedure

Step 1
Create port selectors.

Example:

```xml
<polUni>
  <infraInfra dn="uni/infra">
    <infraLeafS name="leafselector" type="range">
      <infraNodeBlk name="nodeblk" from="101" to="101">
      </infraNodeBlk>
    </infraLeafS>
    <infraRsAccPortP tDn="uni/infra/accportprof-TenantPorts_101"/>
  </infraNodeP>
  <infraAccPortP name="TenantPorts_101">
    <infraHPortS name="portselector" type="range">
      <infraPortBlk name="portblk" fromCard="1" toCard="1" fromPort="41" toPort="41">
      </infraPortBlk>
    </infraHPortS>
    <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-TenantPortGrp_101"/>
  </infraAccPortP>
  <infraFuncP>
    <infraAccPortGrp name="TenantPortGrp_101">
      <infraRsAttEntP tDn="uni/infra/attentp-AttEntityProfTenant"/>
      <infraRsHIfPol tnFabricHIfPolName="default"/>
    </infraAccPortGrp>
  </infraFuncP>
  </infraInfra>
</polUni>
```

Step 2
Create a tenant policy.

Example:

```xml
<fvTenant name="t9" dn="uni/tn-t9" descr="">
  <fvCtx name="t9_ctx1" pcEnfPref="unenforced">
  </fvCtx>
  <fvBD name="t9_bd1" unkMacUcastAct="flood" arpFlood="yes">
    <fvRsCtx tnFvCtxName="t9_ctx1"/>
    <fvSubnet ip="101.9.1.1/24" scope="shared"/>
  </fvBD>
  <l3extOut dn="uni/tn-t9/out-l3extOut1" enforceRtctrl="export" name="l3extOut1">
    <l3extLNodeP name="Node101">
      <l3extRsNodeL3OutAtt rtrId="210.210.121.121" rtrIdLoopBack="no" tDn="topology/pod-1/node-101"/>
    </l3extLNodeP>
    <l3extRsEctx tnPvCtxName="t9_ctx1"/>
    <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
    <l3extInstP matchT="AtleastOne" name="extEpg" prio="unspecified" targetDscp="unspecified">
      <l3extSubnet aggregate="" descr="" ip="176.21.21.21/21" name="">
        <l3extInstP/>
        <l3extOut/>
      </l3extSubnet>
    </l3extInstP>
    <l3extOut/>
  </l3extOut>
</fvTenant>
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Step 3  Create an HSRP interface policy.

**Example:**

```xml
<polUni>
  <fvTenant name="t9" dn="uni/tn-t9" descr="">
    <hsrpIfPol name="hsrpIfPol" ctrl="bfd" delay="4" reloadDelay="11"/>
  </fvTenant>
</polUni>
```

Step 4  Create an HSRP group policy.

**Example:**

```xml
<polUni>
  <fvTenant name="t9" dn="uni/tn-t9" descr="">
    <hsrpIfPol name="hsrpIfPol" ctrl="bfd" delay="4" reloadDelay="11"/>
  </fvTenant>
</polUni>
```

Step 5  Create an HSRP interface profile and an HSRP group profile.

**Example:**

```xml
<polUni>
  <fvTenant name="t9" dn="uni/tn-t9" descr="">
    <l3extOut dn="uni/tn-t9/out-l3extOut1" enforceRtctrl="export" name="l3extOut1">
      <l3extLNodeP name="Node101">
        <l3extLIfP name="eth1-41-v6" ownerKey="" ownerTag="" tag="yellow-green">
          <hsrpIfP name="eth1-41-v6" version="v2">
            <hsrpRsIfPol tnHsrpIfPolName="hsrpIfPol"/>
            <hsrpGroupP descr="" name="HSRPV6-2" groupId="330" groupAf="ipv6" ip="fe80::3" mac="00:00:0C:18:AC:01" ipObtainMode="admin">
              <hsrpRsGroupPol tnHsrpGroupPolName="G1"/>
            </hsrpGroupP>
          </hsrpIfP>
        </l3extLIfP>
        <l3extLIfP name="eth1-41-v4" ownerKey="" ownerTag="" tag="yellow-green">
          <hsrpIfP name="eth1-41-v4" version="v1">
            <hsrpRsIfPol tnHsrpIfPolName="hsrpIfPol"/>
            <hsrpGroupP descr="" name="HSRPV4-2" groupId="51" groupAf="ipv4" ip="177.21.21.21" mac="00:00:0C:18:AC:01" ipObtainMode="admin">
              <hsrpRsGroupPol tnHsrpGroupPolName="G1"/>
            </hsrpGroupP>
          </hsrpIfP>
        </l3extLIfP>
        <l3extRsPathL3OutAtt addr="2002::100/64" descr="" encap="unknown" encapScope="local" ifInstT="l3-port" l1Addr="" mac="00:22:BD:F8:19:FF" mode="regular" mtu="inherit" tDn="topology/pod-1/paths-101/pathep-[eth1/41]" targetDscp="unspecified">
          <l3extIp addr="2004::100/64"/>
        </l3extRsPathL3OutAtt>
      </l3extLNodeP>
    </l3extOut>
  </fvTenant>
</polUni>
```
Cisco ACI GOLF

This chapter contains the following sections:

- Cisco ACI GOLF, on page 249
- Distributing BGP EVPN Type-2 Host Routes to a DCIG, on page 263

Cisco ACI GOLF

The Cisco ACI GOLF feature (also known as Layer 3 EVPN Services for Fabric WAN) enables much more efficient and scalable ACI fabric WAN connectivity. It uses the BGP EVPN protocol over OSPF for WAN routers that are connected to spine switches.
All tenant WAN connections use a single session on the spine switches where the WAN routers are connected. This aggregation of tenant BGP sessions towards the Data Center Interconnect Gateway (DCIG) improves control plane scale by reducing the number of tenant BGP sessions and the amount of configuration required for all of them. The network is extended out using Layer 3 subinterfaces configured on spine fabric ports. Transit routing with shared services using GOLF is not supported.

A Layer 3 external outside network (L3extOut) for GOLF physical connectivity for a spine switch is specified under the infra tenant, and includes the following:

- An l3ExtInstP (EPG) with subnets and contracts. The scope of the subnet is used to control import/export route control and security policies. The bridge domain subnet must be set to advertise externally and it must be in the same VRF as the application EPG and the GOLF L3Out EPG.

- Communication between the application EPG and the GOLF L3Out EPG is governed by explicit contracts (not Contract Preferred Groups).
• An `l3extConsLbl` consumer label that must be matched with the same provider label of an `L3Out` for GOLF in the `infra` tenant. Label matching enables application EPGs in other tenants to consume the LNodeP external L3Out EPG.

• The BGP EVPN session in the matching provider `L3extOut` in the `infra` tenant advertises the tenant routes defined in this `L3Out`.

Guidelines and Limitations

Observe the following GOLF guidelines and limitations:

• GOLF routers must advertise at least one route to Cisco ACI in order to accept traffic. No tunnel is created between leaf switches and the external routers until Cisco ACI receives a route from the external routers.

• All Cisco Nexus 9000 Series ACI-mode switches and all of the Cisco Nexus 9500 platform ACI-mode switch line cards and fabric modules support GOLF. With Cisco APIC, release 3.1(x) and higher, this includes the N9K-C9364C switch.

• At this time, only a single GOLF provider policy can be deployed on spine switch interfaces for the whole fabric.

• Up to APIC release 2.0(2), GOLF is not supported with multipod. In release 2.0 (2) the two features are supported in the same fabric only over Cisco Nexus N9000K switches without “EX” on the end of the switch name; for example, N9K-9312TX. Since the 2.1(1) release, the two features can be deployed together over all the switches used in the multipod and EVPN topologies.

• When configuring GOLF on a spine switch, wait for the control plane to converge before configuring GOLF on another spine switch.

• A spine switch can be added to multiple provider GOLF outside networks (GOLF L3Outs), but the provider labels have to be different for each GOLF L3Out. Also, in this case, the OSPF Area has to be different on each of the `L3extOuts` and use different loopback addresses.

• The BGP EVPN session in the matching provider `L3Out` in the `infra` tenant advertises the tenant routes defined in this `L3extOut`.

• When deploying three GOLF Outs, if only 1 has a provider/consumer label for GOLF, and 0/0 export aggregation, APIC will export all routes. This is the same as existing `L3extOut` on leaf switches for tenants.

• If there is direct peering between a spine switch and a data center interconnect (DCI) router, the transit routes from leaf switches to the ASR have the next hop as the PTEP of the leaf switch. In this case, define a static route on the ASR for the TEP range of that ACI pod. Also, if the DCI is dual-homed to the same pod, then the precedence (administrative distance) of the static route should be the same as the route received through the other link.

• The default `bgpPeerPfxPol` policy restricts routes to 20,000. For ACI WAN Interconnect peers, increase this as needed.

• In a deployment scenario where there are two `L3extOuts` on one spine switch, and one of them has the provider label `prov1` and peers with the DCI 1, the second `L3extOut` peers with DCI 2 with provider label `prov2`. If the tenant VRF has a consumer label pointing to any 1 of the provider labels (either `prov1` or `prov2`), the tenant route will be sent out both DCI 1 and DCI 2.
• When aggregating GOLFOpFlex VRFs, the leaking of routes cannot occur in the ACI fabric or on the GOLF device between the GOLFOpFlex VRF and any other VRF in the system. An external device (not the GOLF router) must be used for the VRF leaking.

Note
Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link. On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface.

For the appropriate MTU values for each platform, see the relevant configuration guides.

We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1.

Using Shared GOLFO Connections Between Multi-Site Sites

APIC GOLFO Connections Shared by Multi-Site Sites

For APIC Sites in a Multi-Site topology, if stretched VRFs share GOLFO connections, follow these guidelines to avoid the risk of cross-VRF traffic issues.

Route Target Configuration between the Spine Switches and the DCI

There are two ways to configure EVPN route targets (RTs) for the GOLFO VRFs: Manual RT and Auto RT. The route target is synchronized between ACI spines and DCIs through OpFlex. Auto RT for GOLFO VRFs has the Fabric ID embedded in the format: – ASN: [FabricID] VNID

If two sites have VRFs deployed as in the following diagram, traffic between the VRFs can be mixed.

<table>
<thead>
<tr>
<th>Site 1</th>
<th>Site 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASN: 100, Fabric ID: 1</td>
<td>ASN: 100, Fabric ID: 1</td>
</tr>
<tr>
<td>VRF A: VNID 1000</td>
<td>VRF A: VNID 2000</td>
</tr>
<tr>
<td>VRF B: VNID 2000</td>
<td>VRF B: VNID 1000</td>
</tr>
</tbody>
</table>

Route Maps Required on the DCI

Since tunnels are not created across sites when transit routes are leaked through the DCI, the churn in the control plane must be reduced as well. EVPN type-5 and type-2 routes sent from GOLFO spine in one site towards the DCI should not be sent to GOLFO spine in another site. This can happen when the DCI to spine switches have the following types of BGP sessions:
To avoid this happening on the DCI, route maps are used with different BGP communities on the inbound and outbound peer policies.

When routes are received from the GOLF spine at one site, the outbound peer policy towards the GOLF spine at another site filters the routes based on the community in the inbound peer policy. A different outbound peer policy strips off the community towards the WAN. All the route-maps are at peer level.

**Recommended Shared GOLF Configuration Using the NX-OS Style CLI**

Use the following steps to configure route maps and BGP to avoid cross-VRF traffic issues when sharing GOLF connections with a DCI between multiple APIC sites that are managed by Multi-Site.

**Procedure**

**Step 1** Configure the inbound route map

**Example:**

**Inbound peer policy to attach community:**

```
route-map multi-site-in permit 10
  set community 1:1 additive
```

**Step 2** Configure the outbound peer policy to filter routes based on the community in the inbound peer policy.

**Example:**

```
ip community-list standard test-com permit 1:1
route-map multi-site-out deny 10
  match community test-com exact-match
route-map multi-site-out permit 11
```

**Step 3** Configure the outbound peer policy to filter the community towards the WAN.

**Example:**

```
ip community-list standard test-com permit 1:1
route-map multi-site-wan-out permit 11
  set comm-list test-com delete
```

**Step 4** Configure BGP.

**Example:**

```
router bgp 1
  address-family l2vpn evpn
```
Configuring ACI GOLF Using the GUI

The following steps describe how to configure infra GOLF services that any tenant network can consume.

Procedure

**Step 1**  
On the menu bar, click **Tenants**, then click **infra** to select the infra tenant.

**Step 2**  
In the **Navigation** pane, expand the **Networking** option and perform the following actions:

a)  
Right-click **External Routed Networks** and click **Create Routed Outside for EVPN** to open the wizard.

b)  
In the **Name** field, enter a name for the policy.

c)  
In the **Route Target** field, choose whether to use automatic or explicit policy-governed BGP route target filtering policy:

   - **Automatic** - Implements automatic BGP route-target filtering on VRFs associated with this routed outside configuration.

   - **Explicit** - Implements route-target filtering through use of explicitly configured BGP route-target policies on VRFs associated with this routed outside configuration.

**Note**  
Explicit route target policies are configured in the **BGP Route Target Profiles** table on the **BGP Page** of the **Create VRF Wizard**. If you select the **Automatic** option the in **Route Target** field, configuring explicit route target policies in the **Create VRF Wizard** might cause BGP routing disruptions.

**Note**  
Explicit route target policies are configured in the **BGP Route Target Profiles** table on the **BGP Page** of the **Create VRF Wizard**. If you select the **Automatic** option the in **Route Target** field, configuring explicit route target policies in the **Create VRF Wizard** might cause BGP routing disruptions.


d)  
Complete the configuration options according to the requirements of your Layer 3 connection.
In the protocol check boxes area, assure that both BGP and OSPF are checked. GOLF requires both BGP and OSPF.

e) Click Next to display the Nodes and Interfaces Protocol Profile tab.
f) In the Define Routed Outside section, in the Name field, enter a name.
g) In the Spines table, click + to add a node entry.
h) In the Node ID drop-down list, choose a spine switch node ID.
i) In the Router ID field, enter the router ID.
j) In the Loopback Addresses, in the IP field, enter the IP address. Click Update.
k) In the OSPF Profile for Sub-interfaces/Routed Sub-Interfaces section in the Name field enter the name of the OSPF Profile for Sun-Interfaces.
l) Click OK.

Note The wizard creates a Logical Node Profile> Configured Nodes> Node Association profile, that set the Extend Control Peering field to enabled.

---

**Cisco ACI GOLF Configuration Example, Using the NX-OS Style CLI**

These examples show the CLI commands to configure GOLF Services, which uses the BGP EVPN protocol over OSPF for WAN routers that are connected to spine switches.

**Configuring the infra Tenant for BGP EVPN**

The following example shows how to configure the infra tenant for BGP EVPN, including the VLAN domain, VRF, Interface IP addressing, and OSPF:

```bash
configure
vlan-domain evpn-dom dynamic
exit
spine 111
  # Configure Tenant Infra VRF overlay-1 on the spine.
  vrf context tenant infra vrf overlay-1
  router-id 10.10.3.3
exit
```
interface ethernet 1/33
  vlan-domain member golf_dom
  exit
interface ethernet 1/33.4
  vrf member tenant infra vrf overlay-1
  mtu 1500
  ip address 5.0.0.1/24
  ip router ospf default area 0.0.0.150
  exit
interface ethernet 1/34
  vlan-domain member golf_dom
  exit
interface ethernet 1/34.4
  vrf member tenant infra vrf overlay-1
  mtu 1500
  ip address 2.0.0.1/24
  ip router ospf default area 0.0.0.200
  exit
router ospf default
  vrf member tenant infra vrf overlay-1
  area 0.0.0.150 loopback 10.10.5.3
  area 0.0.0.200 loopback 10.10.4.3
  exit
exit

Configuring BGP on the Spine Node

The following example shows how to configure BGP to support BGP EVPN:

Configure
spine 111
router bgp 100
  vrf member tenant infra vrf overlay-1
  neighbor 10.10.4.1 evpn
  label golf_aci
  update-source loopback 10.10.4.3
  remote-as 100
  exit
  neighbor 10.10.5.1 evpn
  label golf_aci2
  update-source loopback 10.10.5.3
  remote-as 100
  exit
  exit
exit

Configuring a Tenant for BGP EVPN

The following example shows how to configure a tenant for BGP EVPN, including a gateway subnet which will be advertised through a BGP EVPN session:

configure
tenant sky
  vrf context vrf_sky
  exit
bridge-domain bd_sky
  vrf member vrf_sky
  exit
interface bridge-domain bd_sky
  ip address 59.10.1.1/24
exit
bridge-domain bd_sky2
vrf member vrf_sky
exit
interface bridge-domain bd_sky2
   ip address 59.11.1.1/24
exit
exit

Configuring the BGP EVPN Route Target, Route Map, and Prefix EPG for the Tenant

The following example shows how to configure a route map to advertise bridge-domain subnets through BGP EVPN.

configure
spine 111
   vrf context tenant sky vrf vrf_sky
      address-family ipv4 unicast
         route-target export 100:1
         route-target import 100:1
      exit
   route-map rmap
      ip prefix-list p1 permit 11.10.10.0/24
      match bridge-domain bd_sky
      exit
      match prefix-list p1
      exit
   evpn export map rmap label golf_aci

   route-map rmap2
      match bridge-domain bd_sky
      exit
      match prefix-list p1
      exit
   evpn export map rmap label golf_aci2

external-l3 epg l3_sky
   vrf member vrf_sky
   match ip 80.10.1.0/24
   exit

Configuring GOLF Using the REST API

Procedure

Step 1

The following example shows how to deploy nodes and spine switch interfaces for GOLF, using the REST API:

Example:

POST
https://192.0.20.123/api/mo/uni/golf.xml
Step 2

The XML below configures the spine switch interfaces and infra tenant provider of the GOLF service. Include this XML structure in the body of the POST message.

Example:

```xml
<l3extOut descr="" dn="uni/tn-infra/out-golf" enforceRtctrl="export,import"
  name="golf"
  ownerKey="" ownerTag="" targetDscp="unspecified">
  <l3extRsEctx tnFvCtxName="overlay-1"/>
  <l3extProvLbl descr="" name="golf"
    ownerKey="" ownerTag="" tag="yellow-green"/>
  <l3extLNodeP configIssues="" descr=""
    name="bLeaf" ownerKey="" ownerTag=""
    tag="yellow-green" targetDscp="unspecified">
    <l3extRsNodeL3OutAtt rtrId="10.10.3.3" rtrIdLoopBack="no"
      tDn="topology/pod-1/node-111">
      <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
      <l3extLoopBackIfP addr="10.10.3.3" descr="" name=""/>
    </l3extRsNodeL3OutAtt>
    <l3extRsNodeL3OutAtt rtrId="10.10.3.4" rtrIdLoopBack="no"
      tDn="topology/pod-1/node-112">
      <l3extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
      <l3extLoopBackIfP addr="10.10.3.4" descr="" name=""/>
    </l3extRsNodeL3OutAtt>
  </l3extLNodeP>
  <l3extLIfP descr="" name="portIf-spine1-3"
    ownerKey="" ownerTag="" tag="yellow-green">
    <ospfIfP authKeyId="1" authType="none" descr="" name=""
      <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
      <l3extRsNdIfPol tnNdIfPolName=""/>
      <l3extRsIngressQosDppPol tnQosDppPolName=""/>
      <l3extRsEgressQosDppPol tnQosDppPolName=""/>
      <l3extRsPathL3OutAtt addr="7.2.1.1/24" descr=""
        encaps="vlan-4"
        encapsScope="local"
        ifInstT="sub-interface"
        llAddr="::"
        mac="00:22:BD:F8:19:FF"
        mode="regular"
        mtu="1500"
        tDn="topology/pod-1/paths-111/pathep-[eth1/12]" targetDscp="unspecified"/>
    </l3extLIfP>
  </l3extLIfP>
  <l3extLIfP descr="" name="portIf-spine2-1"
    ownerKey="" ownerTag=""
    tag="yellow-green">
    <ospfIfP authKeyId="1" authType="none" descr=""
      name="">
      <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
      <l3extRsNdIfPol tnNdIfPolName=""/>
      <l3extRsIngressQosDppPol tnQosDppPolName=""/>
      <l3extRsEgressQosDppPol tnQosDppPolName=""/>
      <l3extRsPathL3OutAtt addr="7.1.0.1/24" descr=""
        encaps="vlan-4"
        encapsScope="local"
        ifInstT="sub-interface"
        llAddr="::"
        mac="00:22:BD:F8:19:FF"
        mode="regular"
        mtu="9000"
        tDn="topology/pod-1/paths-112/pathep-[eth1/11]" targetDscp="unspecified"/>
    </l3extLIfP>
  </l3extLIfP>
</l3extOut>
```
<l3extLIfP descr="" name="portIf-spine2-2"
    ownerKey="" ownerTag=""
    tag="yellow-green">
  <ospfIf authKeyId="1" authType="none" descr=""
    name="">
    <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
  </ospfIfP>
</l3extLIfP>

<l3extRsNdIfPol tnNdIfPolName=""/>
<l3extRsIngressQosDppPol tnQosDppPolName=""/>
<l3extRsEgressQosDppPol tnQosDppPolName=""/>
<l3extRsPathL3OutAtt addr="7.2.2.1/24" descr=""
    encaps="vlan-4"
    encapsScope="local"
    ifInstT="sub-interface"
    llAddr="::" mac="00:22:BD:F8:19:FF"
    mode="regular"
    mtu="1500"
    tDn="topology/pod-1/paths-112/pathep-[eth1/12]
    targetDscp="unspecified"/>
</l3extLIfP>

<l3extLIfP descr="" name="portIf-spine1-2"
    ownerKey="" ownerTag=""
    tag="yellow-green">
  <ospfIf authKeyId="1" authType="none" descr=""
    name="">
    <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
  </ospfIfP>
</l3extLIfP>

<l3extRsNdIfPol tnNdIfPolName=""/>
<l3extRsIngressQosDppPol tnQosDppPolName=""/>
<l3extRsEgressQosDppPol tnQosDppPolName=""/>
<l3extRsPathL3OutAtt addr="9.0.0.1/24" descr=""
    encaps="vlan-4"
    encapsScope="local"
    ifInstT="sub-interface"
    llAddr="::" mac="00:22:BD:F8:19:FF"
    mode="regular"
    mtu="9000"
    tDn="topology/pod-1/paths-111/pathep-[eth1/11]
    targetDscp="unspecified"/>
</l3extLIfP>

<l3extLIfP descr="" name="portIf-spine1-1"
    ownerKey="" ownerTag=""
    tag="yellow-green">
  <ospfIf authKeyId="1" authType="none" descr=""
    name="">
    <ospfRsIfPol tnOspfIfPolName="ospfIfPol"/>
  </ospfIfP>
</l3extLIfP>

<l3extRsNdIfPol tnNdIfPolName=""/>
<l3extRsIngressQosDppPol tnQosDppPolName=""/>
<l3extRsEgressQosDppPol tnQosDppPolName=""/>
<l3extRsPathL3OutAtt addr="7.0.0.1/24" descr=""
    encaps="vlan-4"
    encapsScope="local"
    ifInstT="sub-interface"
    llAddr="::" mac="00:22:BD:F8:19:FF"
    mode="regular"
    mtu="1500"
    tDn="topology/pod-1/paths-111/pathep-[eth1/10]
    targetDscp="unspecified"/>
</l3extLIfP>

<bgpInfraPeerP addr="10.10.3.2"
    allowedSelfAsCnt="3"
    ctrl="send-com,send-ext-com"
    descr="" name="" peerCtrl=""
    peerT="wan"
Step 3  The XML below configures the tenant consumer of the infra part of the GOLF service. Include this XML structure in the body of the POST message.

Example:

```xml
<fvTenant descr="" dn="uni/tn-pep6" name="pep6" ownerKey="" ownerTag=""
  <vzBrCP descr="" name="webCtct"
    ownerKey="" ownerTag="" prio="unspecified"
    scope="global" targetDscp="unspecified">
    <vzSubj consMatchT="AtleastOne" descr=""
      name="http" prio="unspecified"
      provMatchT="AtleastOne"
      revFltPorts="yes" targetDscp="unspecified">
      <vzRsSubjFiltAtt directives="" tnVzFilterName="default"/>
    </vzSubj>
  </vzBrCP>
</fvTenant>
```
pcEnfDir="ingress" pcEnfPref="enforced">
<bgpRtTargetP af="ipv6-ucast"
    descr="" name="" ownerKey="" ownerTag="">
    <bgpRtTarget descr="" name="" ownerKey="" ownerTag=""
        rt="route-target:as4-nn2:100:1256"
        type="export"/>
    <bgpRtTarget descr="" name="" ownerKey="" ownerTag=""
        rt="route-target:as4-nn2:100:1256"
        type="import"/>
</bgpRtTargetP>
<bgpRtTargetP af="ipv4-ucast"
    descr="" name="" ownerKey="" ownerTag="">
    <bgpRtTarget descr="" name="" ownerKey="" ownerTag=""
        rt="route-target:as4-nn2:100:1256"
        type="export"/>
    <bgpRtTarget descr="" name="" ownerKey="" ownerTag=""
        rt="route-target:as4-nn2:100:1256"
        type="import"/>
</bgpRtTargetP>
</fvCtx>
</fvBD>
</fvBD>
</fvBD>
</fvBD>
</fvBD>
</fvCtx>
</fvBD>
</fvBD>
</fvBD>
Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Distributing BGP EVPN Type-2 Host Routes to a DCIG

In APIC up to release 2.0(1f), the fabric control plane did not send EVPN host routes directly, but advertised public bridge domain (BD) subnets in the form of BGP EVPN type-5 (IP Prefix) routes to a Data Center Interconnect Gateway (DCIG). This could result in suboptimal traffic forwarding. To improve forwarding, in APIC release 2.1x, you can enable fabric spines to also advertise host routes using EVPN type-2 (MAC-IP) host routes to the DCIG along with the public BD subnets.
To do so, you must perform the following steps:

1. When you configure the BGP Address Family Context Policy, enable Host Route Leak.

2. When you leak the host route to BGP EVPN in a GOLF setup:
   a. To enable host routes when GOLF is enabled, the BGP Address Family Context Policy must be configured under the application tenant (the application tenant is the consumer tenant that leaks the endpoint to BGP EVPN) rather than under the infrastructure tenant.
   b. For a single-pod fabric, the host route feature is not required. The host route feature is required to avoid sub-optimal forwarding in a multi-pod fabric setup. However, if a single-pod fabric is setup, then in order to leak the endpoint to BGP EVPN, a Fabric External Connection Policy must be configured to provide the ETEP IP address. Otherwise, the host route will not leak to BGP EVPN.

3. When you configure VRF properties:
   a. Add the BGP Address Family Context Policy to the BGP Context Per Address Families for IPv4 and IPv6.
   b. Configure BGP Route Target Profiles that identify routes that can be imported or exported from the VRF.

### Distributing BGP EVPN Type-2 Host Routes to a DCIG Using the GUI

Enable distributing BGP EVPN type-2 host routes with the following steps:

#### Before you begin

You must have already configured ACI WAN Interconnect services in the infra tenant, and configured the tenant that will consume the services.

#### Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>On the menu bar, click <strong>Tenants &gt; infra</strong>.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>In the Navigation pane, expand the <strong>External Routed Networks</strong>, then expand <strong>Protocol Policies</strong> and <strong>BGP</strong>.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Right-click <strong>BGP Address Family Context</strong>, select <strong>Create BGP Address Family Context Policy</strong> and perform the following steps:</td>
</tr>
<tr>
<td>a</td>
<td>Type a name for the policy and optionally add a description.</td>
</tr>
<tr>
<td>b</td>
<td>Click the <strong>Enable Host Route Leak</strong> check box.</td>
</tr>
<tr>
<td>c</td>
<td>Click <strong>Submit</strong>.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Click <strong>Tenants &gt; tenant-name</strong> (for a tenant that will consume the BGP Address Family Context Policy) and expand <strong>Networking</strong>.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Expand <strong>VRFs</strong> and click the VRF that will include the host routes you want to distribute.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>When you configure the VRF properties, add the <strong>BGP Address Family Context Policy</strong> to the <strong>BGP Context Per Address Families</strong> for IPv4 and IPv6.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Click <strong>Submit</strong>.</td>
</tr>
</tbody>
</table>
Enabling Distributing BGP EVPN Type-2 Host Routes to a DCIG Using the NX-OS Style CLI

**Procedure**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Configure distributing EVPN type-2 host routes to a DCIG with the following commands in the BGP address family configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td>This template will be available on all nodes where tenant bgp_t1 has a VRF deployment. To disable distributing EVPN type-2 host routes, enter the <strong>no host-rt-enable</strong> command.</td>
</tr>
<tr>
<td>apic1(config)# leaf 101</td>
<td></td>
</tr>
<tr>
<td>apic1(config-leaf)# template bgp address-family bgpAf1 tenant bgp_t1</td>
<td></td>
</tr>
<tr>
<td>apic1(config-bgp-af)# distance 250 240 230</td>
<td></td>
</tr>
<tr>
<td>apic1(config-bgp-af)# host-rt-enable</td>
<td></td>
</tr>
<tr>
<td>apic1(config-bgp-af)# exit</td>
<td></td>
</tr>
</tbody>
</table>

Enabling Distributing BGP EVPN Type-2 Host Routes to a DCIG Using the REST API

Enable distributing BGP EVPN type-2 host routes using the REST API, as follows:

**Before you begin**

EVPN services must be configured.

**Procedure**

**Step 1**

Configure the Host Route Leak policy, with a POST containing XML such as in the following example:

**Example:**

```xml
<bgpCtxAfPol descr="" ctrl="host-rt-leak" name="bgpCtxPol_0 status=""/>
```

**Step 2**

Apply the policy to the VRF BGP Address Family Context Policy for one or both of the address families using a POST containing XML such as in the following example:

**Example:**

```xml
<fvCtx name="vni-10001">
<fvRsCtxToBgpCtxAfPol af="ipv4-ucast" tnBgpCtxAfPolName="bgpCtxPol_0"/>
<fvRsCtxToBgpCtxAfPol af="ipv6-ucast" tnBgpCtxAfPolName="bgpCtxPol_0"/>
</fvCtx>
```
Enabling Distributing BGP EVPN Type-2 Host Routes to a DCIG Using the REST API
Multi-Pod

This chapter contains the following sections:

- About Multi-Pod, on page 267
- Multi-Pod Provisioning, on page 268
- Guidelines for Setting Up a Multi-Pod Fabric, on page 269
- Setting Up the Multi-Pod Fabric, on page 272
- Sample IPN Configuration for Multi-Pod For Cisco Nexus 9000 Series Switches, on page 281
- Moving an APIC from One Pod to Another Pod, on page 282

About Multi-Pod

Multi-Pod enables provisioning a more fault-tolerant fabric comprised of multiple pods with isolated control plane protocols. Also, Multi-Pod provides more flexibility with regard to the full mesh cabling between leaf and spine switches. For example, if leaf switches are spread across different floors or different buildings, Multi-Pod enables provisioning multiple pods per floor or building and providing connectivity between pods through spine switches.

Multi-Pod uses MP-BGP EVPN as the control-plane communication protocol between the ACI spines in different pods.

WAN routers can be provisioned in the Inter-Pod Network (IPN), directly connected to spine switches, or connected to border leaf switches. Spine switches connected to the IPN are connected to at least one leaf switch in the pod.

Multi-Pod uses a single APIC cluster for all the pods; all the pods act as a single fabric. Individual APIC controllers are placed across the pods but they are all part of a single APIC cluster.
Multi-Pod Provisioning

The IPN is not managed by the APIC. It must be preconfigured with the following information:

- Configure the interfaces connected to the spines of all pods. Use Layer 3 sub-interfaces tagging traffic with VLAN-4 and increase the MTU at least 50 bytes above the maximum MTU required for inter-site control plane and data plane traffic.

  If remote leaf switches are included in any pods, we strongly recommend that you deploy ACI software release 4.1(2) or later. A more complex configuration is required with earlier releases to connect the spines to the IPN, mandating the use of two sub-interfaces (with VLAN-4 and VLAN-5 tags) and a separate VRF on the IPN devices. For more information, see the Cisco ACI Remote Leaf Architecture White Paper.

- Enable OSPF on sub-interfaces with the correct area ID.

- Enable DHCP Relay on IPN interfaces connected to all spines.

- Enable PIM.

- Add bridge domain GIPO range as PIM Bidirectional (bidir) group range (default is 225.0.0.0/8).

  A group in bidir mode has only shared tree forwarding capabilities.

- Add 239.255.255.240/28 as PIM bidir group range.
- Enable PIM and IGMP on the interfaces connected to all spines.

**Note**

When deploying PIM *bidir*, at any given time it is only possible to have a single active RP (Rendezvous Point) for a given multicast group range. RP redundancy is hence achieved by leveraging a Phantom RP configuration. Because multicast source information is no longer available in Bidir, the Anycast or MSDP mechanism used to provide redundancy in sparse-mode is not an option for *bidir*.

---

**Figure 29: Multi-Pod Provisioning**

---

**Guidelines for Setting Up a Multi-Pod Fabric**

To configure a Multi-Pod fabric, follow these guidelines:

- All Cisco Nexus 9000 Series ACI-mode switches and all of the Cisco Nexus 9500 platform ACI-mode switch line cards and fabric modules support Multi-Pod. With Cisco APIC Release 3.1(x) and later, this includes the N9K-C9364C switch.

- Create the associated node group and Layer 3 Out policies.

- Before you make any changes to a spine switch, ensure that there is at least one operationally “up” external link that is participating in the Multi-Pod topology. Failure to do so could bring down the Multi-Pod connectivity.
• If you have to convert a Multi-Pod setup to a single pod (containing only Pod 1), the APIC controller(s) connected to the pod(s) that are decommissioned should be re-initialized and connected to the leaf switches in Pod 1, which will allow them to re-join the cluster after going through the initial setup script. See Moving an APIC from One Pod to Another Pod, on page 282 for those instructions. The TEP pool configuration should not be deleted.

• Support for Cisco ACI GOLF (also known as Layer 3 EVPN Services for Fabric WAN) and Multi-Pod used together varies, depending on the APIC release:
  - For releases prior to APIC release 2.0(2), GOLF was not supported with Multi-Pod.
  - For APIC release 2.0(2) to APIC release 2.1(1), GOLF and Multi-Pod were supported in the same fabric only over Generation 1 switches, which are switch models that can be identified by the lack of "EX" or "FX" at the end of the switch name (for example N9K-9312TX).
  - Since the 2.1(1) release, the two features can be deployed together over all the switches used in the Multi-Pod and EVPN topologies.

For more information on GOLF, see Cisco ACI GOLF, on page 249.

• In a Multi-Pod fabric, the Pod 1 configuration (with the associated TEP pool) must always exist on APIC, as the APIC nodes are always addressed from the Pod 1 TEP pool. This remains valid also in the scenario where the Pod 1 is physically decommissioned (which is a fully supported procedure) so that the original Pod 1 TEP pool is not re-assigned to other pods that may be added to the fabric.

• In a Multi-Pod fabric setup, if a new spine switch is added to a pod, it must first be connected to at least one leaf switch in the pod. This enables the APIC to discover the spine switch and join it to the fabric.

• After a pod is created and nodes are added in the pod, deleting the pod results in stale entries from the pod that are active in the fabric. This occurs because the APIC uses open source DHCP, which creates some resources that the APIC cannot delete when a pod is deleted.

• For APIC releases 2.2(2) and earlier, Forward Error Correction (FEC) is enabled for all 100G transceivers by default. Do not use QSFP-100G-LR4-S / QSFP-100G-LR4 transceivers for Multi-Pod configuration. ACI enables FEC mode by default for 100G-LR4 optics. Spine switches with these optics should not be used for Multi-Pod if the spines connect to IPN devices that cannot enable FEC mode.

• The following is required when deploying a pair of Active/Standby Firewalls (FWs) across pods:

  **Scenario 1:** Use of PBR to redirect traffic through the FW:
  - Mandates the use of Service Graphs and enables connecting the FW inside/outside interfaces to the ACI Fabric. This feature is fully supported from the 2.1(1) release.
  - Flows from all the compute leaf nodes are always sent to the leaf switches connected to the Active FW.

  **Scenario 2:** Use of separate L3Out connections in each pod between the border leaf switches and the FW:
  - Fully supported starting from 2.0(1) release.
  - Only supported with dynamic routing (no static routing) and with Cisco ASA (not with FWs using VRRP).
  - Active FW only peers with the BL nodes in the local pod. The leafs inject external routing information into the fabric.
Dynamic peering sessions must be re-established in the new pod, due to longer traffic outages after FW failover.

Scenario 3: Use of a single L3Out stretched across pods.

- Active and Standby FWs connected to a single leaf node with a physical link or (local port-channel) is supported in releases 2.1(2e) and 2.2(2e) on all ACI leaf nodes (E, EX, FX).
- Active and Standby FWs connected in vPC mode in each pod to a pair of leaf nodes is supported from release 2.3(1) and only for EX, FX or newer ACI leaf nodes.

- If you delete and recreate the Multi-Pod L3out, for example to change the name of a policy, a clean reload of some of the spine switches in the fabric must be performed. The deletion of the Multi-Pod L3Out causes one or more of the spine switches in the fabric to lose connectivity to the APICs and these spine switches are unable to download the updated policy from the APIC. Which spine switches get into such a state depends upon the deployed topology. To recover from this state, a clean reload must be performed on these spine switches. The reload is performed using the `setup-clean-config.sh` command, followed by the reload command on the spine switch.

Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or Multi-Pod connections through an Inter-Pod Network (IPN), it is recommended that the interface MTU is set appropriately on both ends of a link. On some platforms, such as Cisco ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value does not take into account the Ethernet headers (matching IP MTU, and excluding the 14-18 Ethernet header size), while other platforms, such as IOS-XR, include the Ethernet header in the configured MTU value. A configured value of 9000 results in a max IP packet size of 9000 bytes in Cisco ACI, Cisco NX-OS, and Cisco IOS, but results in a max IP packet size of 8986 bytes for an IOS-XR untagged interface.

For the appropriate MTU values for each platform, see the relevant configuration guides.

We highly recommend that you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as `ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1`.

You can set the global MTU for control plane (CP) packets sent by the nodes (APIC and the switches) in the fabric at **Fabric > Access Policies > Global Policies > CP MTU Policy**.

In a Multi-Pod topology, the MTU set for the fabric external ports must be greater than or equal to the CP MTU value set. Otherwise, the fabric external ports might drop the CP MTU packets.

If you change the IPN or CP MTU, we recommend changing the CP MTU value first, then changing the MTU value on the spine of the remote pod. This reduces the risk of losing connectivity between the pods due to MTU mismatch. This is to ensure that the MTU across all the interfaces of the IPN devices between the pods is large enough for both control plane and VXLAN data plane traffic at any given time. For data traffic, keep in mind the extra 50 bytes due to VXLAN.

To decommission a pod, decommission all the nodes in the pod. For instructions, see *Decommissioning and Recommissioning a Pod* in *Cisco APIC Troubleshooting Guide*.
Setting Up the Multi-Pod Fabric

In Cisco Application Policy Infrastructure Controller (APIC) 4.0(1) and later, a wizard was added to the GUI to simplify Multi-Pod configuration. To configure Multi-Pod using the GUI, follow the procedures in this section.

Setting up Multi-Pod between two physical pods involves preparing an existing physical pod to communicate over the interpod network (IPN) with the new pod. You then add the physical pod, and Cisco Cisco APIC creates the Multi-Pod fabric.

You can also configure Multi-Pod using the NX-OS style CLI and REST API. See the sections Setting Up Multi-Pod Fabric Using the NX-OS CLI, on page 276 and Setting Up Multi-Pod Fabric Using the REST API, on page 279 in this guide for instructions.

**Note**

You can also use the GUI wizard to add a Cisco Application Centric Infrastructure (ACI) Virtual Pod (vPod) as a remote extension of the Cisco ACI fabric. For information about Cisco ACI vPod, see the Cisco ACI vPod documentation.

Preparing the Pod for IPN Connectivity

Before you create a new pod, you first must ensure that the existing physical pod can communicate with it.

**Procedure**

**Step 1** Log in to the Cisco APIC.

**Step 2** Choose Fabric > Inventory.

**Step 3** Expand Quick Start and click Add Pod.

**Step 4** In the work pane, click Add Pod.

**Step 5** In the Configure Interpod Connectivity STEP 1 > Overview panel, review the tasks that are required to configure interpod network (IPN) connectivity, and then click Get Started.

**Step 6** In the Configure Interpod Connectivity STEP 2 > IP Connectivity dialog box, complete the following steps:

a) If you see a Name field in an L3 Outside Configuration area, choose an existing fabric external routing profile from the Name drop-down list.

b) Using the Spine ID selector, choose the spine.

   Click the + (plus) icon to add the IDs of more spines.

c) In the Interfaces area, in the Interface field, enter the spine switch interface (slot and port) used to connect to the IPN.

   Click the + (plus) icon to add more interfaces.

d) In the IPV4 Address field, enter the IPv4 gateway address and network mask for the interface.

e) From the MTU (bytes) drop-down list, choose a value for the maximum transmit unit of the external network.
The range is 1500 to 9216.

f) Click Next.

**Step 7** Configure Interpod Connectivity STEP 3 > Routing Protocols dialog box, in the OSPF area, complete the following steps:

a) Leave the Use Defaults checked or uncheck it.

   When the Use Defaults check box is checked, the GUI conceals the optional fields for configuring Open Shortest Path (OSPF). When it is unchecked, it displays all the fields. The check box is checked by default.

b) In the Area ID field, enter the OSPF area ID.

c) In the Area Type area, choose an OSPF area type.

   You can choose NSSA area, Regular area (the default), or Stub area.

d) (Optional) With the Area Cost selector, choose an appropriate OSPF area cost value.

e) From the Interface Policy drop-down list, choose or configure an OSPF interface policy.

   You can choose an existing policy, or you can create one with the Create OSPF Interface Policy dialog box.

**Step 8** In the Configure Interpod Connectivity STEP 3 > Routing Protocols dialog box, in the BGP area, complete the following steps:

a) Leave the Use Defaults checked or uncheck it.

   When the Use Defaults check box is checked, the GUI conceals the fields for configuring Border Gateway Protocol (BGP). When it is unchecked, it displays all the fields. The check box is checked by default.

b) In the Community field, enter the community name.

   We recommend that you use the default community name. If you use a different name, follow the same format as the default.

c) In the Peering Type field, choose either Full Mesh or Route Reflector for the route peering type.

   If you choose Route Reflector in the Peering Type field and you later want to remove the spine switch from the controller, you must first disable Route Reflector in the BGP Route Reflector page. Not doing so results in an error.

   To disable a route reflector, right-click on the appropriate route reflector in the Route Reflector Nodes area in the BGP Route Reflector page and select Delete. See the section "Configuring an MP-BGP Route Reflector Using the GUI" in the chapter "MP-BGP Route Reflector" in the Cisco APIC Layer 3 Networking Configuration Guide.

d) In the Peer Password field, enter the BGP peer password.

e) In the Confirm Password field, reenter the BGP peer password.

f) In the External Route Reflector Nodes area, click the + (plus) icon to add nodes.

   For redundancy purposes, more than one spine is configured as a route reflector node: one primary reflector and one secondary reflector. It is best practice to deploy at least one external route reflector per pod for redundancy purposes.

   The External Route Reflector Nodes fields appear only if you chose Route Reflector as the peering type.

g) Click Next.
Step 9

In the Configure Interpod Connectivity STEP 4 > External TEP dialog box, complete the following steps:

a) Leave the Use Defaults checked or uncheck it.

When the Use Defaults check box is checked, the GUI conceals the optional fields for configuring the external TEP pool. When it is unchecked, it displays all the fields. The check box is checked by default.

b) Note the nonconfigurable values in the Pod and Internal TEP Pool fields.

c) In the External TEP Pool field, enter the external TEP pool for the physical pod.

The external TEP pool must not overlap the internal TEP pool or external TEP pools belonging to other pods.

d) In the Dataplane TEP Pool field, accept the default, which is generated when you configure the External TEP Pool; if you enter another address, it must be outside of the external TEP pool.

e) (Optional) In the Router ID field, enter the IPN router IP address.

f) (Optional) In the Loopback Address field, enter the IPN router loopback IP address.

If you uncheck the Use Defaults, the Cisco APIC displays the nonconfigurable Unicast TEP IP and Spine ID fields.

g) Click Finish.

The Summary panel appears, displaying details of the IPN configuration. You can also click View JSON to view the REST API for the configuration. You can save the REST API for later use.

What to do next

Take one of the following actions:

• You can proceed directly with adding a pod, continuing with the procedure Adding a Pod to Create a Multi-Pod Fabric, on page 274 in this guide.

• Close the Configure Interpod Connectivity dialog box and add the pod later, returning to the procedure Adding a Pod to Create a Multi-Pod Fabric, on page 274 in this guide.

Adding a Pod to Create a Multi-Pod Fabric

The Add Physical Pod dialog enables you to set up a Multi-Pod environment. You define a new physical pod ID and tunnel endpoint (TEP) pool. You also configure the new pod network settings and the subinterfaces for the physical spines.

Before you begin

You have performed the following tasks:

• Created the node group and L3Out policies.

• Configured the interpod network (IPN). For a sample configuration, see Sample IPN Configuration for Multi-Pod For Cisco Nexus 9000 Series Switches, on page 281 in this guide.

• Prepared an existing pod to communicate with the new pod over the IPN. See the procedure Preparing the Pod for IPN Connectivity, on page 272 in this guide.

• Made sure that the spine switch that connects to the IPN also connects to at least one leaf switch in the pod.
• Created a tunnel endpoint (TEP) pool. See the procedure Preparing the Pod for IPN Connectivity, on page 272 in this guide.

Procedure

Step 1 Log into Cisco Application Policy Infrastructure Controller (APIC).

Step 2 Take one of the following actions:

• If you completed the procedure Preparing the Pod for IPN Connectivity, on page 272 and have not closed the Configure Interpod Connectivity dialog box, skip Step 3 through Step 5, and resume this procedure at Step 6.

• If you have completed the procedure Preparing the Pod for IPN Connectivity, on page 272 and have closed the Configure Interpod Connectivity dialog box, proceed to Step 3 in this procedure.

Step 3 Choose Fabric > Inventory.

Step 4 Click Quick Start and click Add Pod.

Step 5 In the work pane, click Add Pod.

Step 6 In the Add Physical Pod STEP 2 > Pod Fabric dialog box, complete the following steps:

a) In the Pod ID field, choose the pod ID.

   The pod ID can be any positive integer; however, it must be unique in the Cisco ACI fabric.

b) In the Pod TEP Pool field, enter the pool address and subnet.

   The pod TEP pool represents a range of traffic encapsulation identifiers and is a shared resource and can be consumed by multiple domains.

c) With the Spine ID selector, choose the spine ID.

   Choose more spine IDs by clicking the + (plus) icon.

d) In the Interfaces area, in the Interface field, enter the spine switch interface (slot and port) that is used to connect to the interpod network (IPN).

e) In the IPv4 Address field, enter the IPv4 gateway address and network mask for the interface.

f) In the MTU (bytes) field, choose a value for the maximum transmit unit (MTU) of the external network.

   You can configure another interface by clicking the + (plus) icon.

Step 7 In the Add Physical Pod STEP 3 > External TEP dialog box, complete the following steps:

a) Leave the Use Defaults check box checked or uncheck it to display the optional fields to configure an external TEP pool.

b) Note the values in the Pod and Internal TEP Pool fields, which are already configured.

c) In the External TEP Pool field, enter the external TEP pool for the physical pod.

   The external TEP pool must not overlap the internal TEP pool.

d) In the Dataplane TEP IP field, enter the address that is used to route traffic between pods.

e) (Optional) In the Unicast TEP IP field, enter the unicast TEP IP address.

   Cisco APIC automatically configures the unicast TEP IP address when you enter the data plane TEP IP address.

f) (Optional) Note the value in the nonconfigurable Node field.
g) (Optional) In the **Router ID** field, enter the IPN router IP address.

Cisco APIC automatically configures the router IP address when you enter the data plane TEP address.

h) In the **Loopback Address** field, enter the router loopback IP address.

Leave the **Loopback Address** blank if you use a router IP address.

i) Click **Finish**.

---

### Setting Up Multi-Pod Fabric Using the NX-OS CLI

#### Before you begin

- The node group and L3Out policies have already been created.

#### Procedure

---

**Step 1** Set up the multi-pod, as in the following example:

**Example:**

```bash
ifav4-ifc1# show run system
# Command: show running-config system
# Time: Mon Aug  1 21:32:03 2016
system cluster-size 3
system switch-id FOX2016G9DW 204 ifav4-spine4 pod 2
system switch-id SAL1748HS6D 201 ifav4-spine1 pod 1
system switch-id SAL1803L25H 102 ifav4-leaf2 pod 1
system switch-id SAL1819XPF4 101 ifav4-leaf1 pod 1
system switch-id SAL1931A3B 203 ifav4-spine2 pod 2
system switch-id SAL1934MNY0 103 ifav4-leaf3 pod 1
system switch-id SAL1934MNY3 104 ifav4-leaf4 pod 1
system switch-id SAL1938P7A6 202 ifav4-spine3 pod 1
system switch-id SAL1938PHBB 105 ifav4-leaf5 pod 2
system switch-id SAL1942R857 106 ifav4-leaf6 pod 2
system pod 1 tep-pool 10.0.0.0/16
system pod 2 tep-pool 10.1.0.0/16
ifav4-ifc1#
```

**Step 2** Configure a VLAN domain, as in the following example:

**Example:**

```bash
ifav4-ifc1# show running-config vlan-domain l3Dom
# Command: show running-config vlan-domain l3Dom
vlan-domain l3Dom
  vlan 4
  exit
ifav4-ifc1#
```

**Step 3** Configure the fabric external connectivity, as in the following example:

**Example:**

```bash
ifav4-ifc1# show running-config fabric-external
# Command: show running-config fabric-external
```
Step 4 Configure the spine switch interface and OSPF configuration as in the following example:

Example:

```bash
# Command: show running-config spine
# Time: Mon Aug  1 21:34:41 2016
spine 201
  vrf context tenant infra vrf overlay-1
  router-id 201.201.201.201
  exit
  interface ethernet 1/1
    vlan-domain member l3Dom
    exit
  interface ethernet 1/1.4
    vrf member tenant infra vrf overlay-1
    ip address 201.1.1.1/30
    ip router ospf default area 1.1.1.1
    ip ospf cost 1
    exit
  interface ethernet 1/2
    vlan-domain member l3Dom
    exit
  interface ethernet 1/2.4
    vrf member tenant infra vrf overlay-1
    ip address 201.2.1.1/30
    ip router ospf default area 1.1.1.1
    ip ospf cost 1
    exit
  router ospf default
    vrf member tenant infra vrf overlay-1
      area 1.1.1.1 loopback 201.201.201.201
    area 1.1.1.1 interpod peering
    exit
  exit
spine 202
  vrf context tenant infra vrf overlay-1
  exit
  interface ethernet 1/2
    vlan-domain member l3Dom
    exit
  interface ethernet 1/2.4
    vrf member tenant infra vrf overlay-1
    ip address 202.1.1.1/30
    ip router ospf default area 1.1.1.1
    exit
```
router ospf default
  vrf member tenant infra vrf overlay-1
    area 1.1.1.1 interpod peering
  exit
  exit
spine 203
  vrf context tenant infra vrf overlay-1
    router-id 203.203.203.203
  exit
  interface ethernet 1/1
    vlan-domain member l3Dom
  exit
  interface ethernet 1/1.4
    vrf member tenant infra vrf overlay-1
    ip address 203.1.1.1/30
    ip ospf default area 0.0.0.0
    ip ospf cost 1
  exit
  interface ethernet 1/2
    vlan-domain member l3Dom
  exit
  interface ethernet 1/2.4
    vrf member tenant infra vrf overlay-1
    ip address 203.2.1.1/30
    ip ospf default area 0.0.0.0
    ip ospf cost 1
  exit
router ospf default
  vrf member tenant infra vrf overlay-1
    area 0.0.0.0 loopback 203.203.203.203
    area 0.0.0.0 interpod peering
  exit
  exit
spine 204
  vrf context tenant infra vrf overlay-1
    router-id 204.204.204.204
  exit
  interface ethernet 1/31
    vlan-domain member l3Dom
  exit
  interface ethernet 1/31.4
    vrf member tenant infra vrf overlay-1
    ip address 204.1.1.1/30
    ip ospf default area 0.0.0.0
    ip ospf cost 1
  exit
router ospf default
  vrf member tenant infra vrf overlay-1
    area 0.0.0.0 loopback 204.204.204.204
    area 0.0.0.0 interpod peering
  exit
  exit
ifav4-ifc1#
Setting Up Multi-Pod Fabric Using the REST API

Procedure

Step 1  Login to Cisco APIC:
Example:
http://<apic-name/ip>:80/api/aaaLogin.xml
data: <aaaUser name="admin" pwd="ins3965!"/>

Step 2  Configure the TEP pool:
Example:
http://<apic-name/ip>:80/api/policymgr/mo/uni/controller.xml
<fabricSetupPol status=''>
  <fabricSetupP podId="1" tepPool="10.0.0.0/16" />
  <fabricSetupP podId="2" tepPool="10.1.0.0/16" status='' />
</fabricSetupPol>

Step 3  Configure the node ID policy:
Example:
http://<apic-name/ip>:80/api/node/mo/uni/controller.xml
<fabricNodeIdentPol>
  <fabricNodeIdentP serial="SAL1819RXP4" name="ifav4-leaf1" nodeId="101" podId="1"/>
  <fabricNodeIdentP serial="SAL1803L25H" name="ifav4-leaf2" nodeId="102" podId="1"/>
  <fabricNodeIdentP serial="SAL1934NNY0" name="ifav4-leaf3" nodeId="103" podId="1"/>
  <fabricNodeIdentP serial="SAL1934NNY3" name="ifav4-leaf4" nodeId="104" podId="1"/>
  <fabricNodeIdentP serial="SAL1748H56D" name="ifav4-spine1" nodeId="201" podId="1"/>
  <fabricNodeIdentP serial="SAL1938P7A6" name="ifav4-spine3" nodeId="202" podId="2"/>
  <fabricNodeIdentP serial="SAL1938PHBB" name="ifav4-spine5" nodeId="205" podId="2"/>
  <fabricNodeIdentP serial="SAL1931LA3B" name="ifav4-spine2" nodeId="203" podId="2"/>
  <fabricNodeIdentP serial="FGE173400A9" name="ifav4-spine4" nodeId="204" podId="2"/>
</fabricNodeIdentPol>

Step 4  Configure infra L3Out and external connectivity profile:
Example:
http://<apic-name/ip>:80/api/node/mo/uni.xml
<polUni>
  <fvTenant descr='' dn='uni/tn-infra' name='infra' ownerKey='' ownerTag=''>
    <l3extOut descr='' enforceRtctrl='export' name='multipod' ownerKey='' ownerTag='' targetDscp='unspecified' status=''>
      <ospfExtP areaId='0' areaType='regular' status='' />
      <l3extRsCtx tnFvCtxName='overlay-1' />
      <l3extProVlbl descr='' name='prov_mpl' ownerKey='' ownerTag='' tag='yellow-green' />
      <l3extLNodeP name='bSpine'>
        <l3extRsNodeL3OutAtt rtrId='201.201.201.201' rtrIdLoopBack='no'
          tDn='topology/pod-1/node-201'>
          <l3extInfraNodeP descr='' fabricExtCtrlPeering='yes' name='' />
          <l3extLoopBackIfP addr='201::201/128' descr='' />
        </l3extRsNodeL3OutAtt>
      </l3extLNodeP>
    </l3extOut>
  </fvTenant>
</polUni>
Multi-Pod

Setting Up Multi-Pod Fabric Using the REST API

<13extLoopBackIfP addr="201.201.201.201/32" descr="" name=""/>
</13extRsNodeL3OutAtt>

tDn="topology/pod-1/node-202">
<13extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
<13extLoopBackIfP addr="202::202/128" descr="" name=""/>
</13extRsNodeL3OutAtt>

<13extRsNodeL3OutAtt rtrId="203.203.203.203" rtrIdLoopBack="no"
tDn="topology/pod-2/node-203">
<13extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
<13extLoopBackIfP addr="203::203/128" descr="" name=""/>
<13extLoopBackIfP addr="203.203.203.203/32" descr="" name=""/>
</13extRsNodeL3OutAtt>

<13extRsNodeL3OutAtt rtrId="204.204.204.204" rtrIdLoopBack="no"
tDn="topology/pod-2/node-204">
<13extInfraNodeP descr="" fabricExtCtrlPeering="yes" name=""/>
<13extLoopBackIfP addr="204::204/128" descr="" name=""/>
<13extLoopBackIfP addr="204.204.204.204/32" descr="" name=""/>
</13extRsNodeL3OutAtt>

<l3extLIfP name='portIf'>
<l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-1/paths-201/pathep-[eth1/1]"
encap='vlan-4' ifInstT='sub-interface' addr="201.1.1.1/30" />
<l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-1/paths-201/pathep-[eth1/2]"
encap='vlan-4' ifInstT='sub-interface' addr="201.2.1.1/30" />
<l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-1/paths-202/pathep-[eth1/2]"
encap='vlan-4' ifInstT='sub-interface' addr="202.1.1.1/30" />
<l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-2/paths-203/pathep-[eth1/1]"
encap='vlan-4' ifInstT='sub-interface' addr="203.1.1.1/30" />
<l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-2/paths-203/pathep-[eth1/2]"
encap='vlan-4' ifInstT='sub-interface' addr="203.2.1.1/30" />
<l3extRsPathL3OutAtt descr='asr' tDn="topology/pod-2/paths-204/pathep-[eth4/31]"
encap='vlan-4' ifInstT='sub-interface' addr="204.1.1.1/30" />
</l3extLIfP>
</l3extLNodeP>

<l3extInstP descr="" matchT="AtleastOne" name="instp1" prio="unspecified"
targetDscp="unspecified">
<fvRsCustQosPol tnQosCustomPolName=""/>
</l3extInstP>
</l3extOut>

<fvFabricExtConnP descr="" id="1" name="Fabric_Ext_Conn_Pol1" rt="extended:as2-nn4:5:16"
status="">
Sample IPN Configuration for Multi-Pod For Cisco Nexus 9000 Series Switches

• The deployment of a dedicated VRF in the IPN for Inter-Pod connectivity is optional, but is a best practice recommendation. You can also use a global routing domain as an alternative.

• For the area of the sample configuration that shows `ip dhcp relay address 10.0.0.1`, this configuration is valid based on the assumption that the TEP pool of Pod 1 is 10.0.0.0/x.

---

Procedure

Sample configuration:

Example:

Sample IPN configuration for Cisco Nexus 9000 series switches:

```
{pod1-spine1}-----2/7[ IPN-N9K ]2/9-----{pod2-spine1}
```

```bash
feature dhcp
feature pim

service dhcp
ip dhcp relay
ip pim ssm range 232.0.0.0/8

# Create a new VRF for Multipod.
vrf context fabric-mpod
  ip pim rp-address 12.1.1.1 group-list 225.0.0.0/8 bidir
  ip pim rp-address 12.1.1.1 group-list 239.255.255.240/28 bidir
```
ip pim ssm range 232.0.0.0/8

interface Ethernet2/7
  no switchport
  mtu 9150
  no shutdown

interface Ethernet2/7.4
  description pod1-spine1
  mtu 9150
  encapsulation dot1q 4
  vrf member fabric-mpod
  ip address 201.1.2.2/30
  ip router ospf a1 area 0.0.0.0
  ip pim sparse-mode
  ip dhcp relay address 10.0.0.1
  ip dhcp relay address 10.0.0.2
  ip dhcp relay address 10.0.0.3
  no shutdown

interface Ethernet2/9
  no switchport
  mtu 9150
  no shutdown

interface Ethernet2/9.4
  description to pod2-spine1
  mtu 9150
  encapsulation dot1q 4
  vrf member fabric-mpod
  ip address 203.1.2.2/30
  ip router ospf a1 area 0.0.0.0
  ip pim sparse-mode
  ip dhcp relay address 10.0.0.1
  ip dhcp relay address 10.0.0.2
  ip dhcp relay address 10.0.0.3
  no shutdown

interface loopback29
  vrf member fabric-mpod
  ip address 12.1.1.1/32

router ospf a1
  vrf fabric-mpod
  router-id 29.29.29.29

---

Moving an APIC from One Pod to Another Pod

Use this procedure to move an APIC from one pod to another pod in an Multi-Pod setup.

**Procedure**

**Step 1**
Decommission the APIC in the cluster.
   a) On the menu bar, choose **System > Controllers**.
b) In the Navigation pane, expand Controllers > apic_controller_name > Cluster as Seen by Node.

c) In the Navigation pane, click an apic_controller_name that is within the cluster and not the controller that is being decommissioned.

d) In the Work pane, verify that the Health State in the Active Controllers summary table indicates the cluster is Fully Fit before continuing.

e) In the Work pane, click Actions > Decommission.

f) Click Yes.

   The decommissioned controller displays Unregistered in the Operational State column. The controller is then taken out of service and no longer visible in the Work pane.

Step 2  Move the decommissioned APIC to the desired pod.

Step 3  Enter the following commands to reboot the APIC.

apic1# acidiag touch setup
apic1# acidiag reboot

Step 4  In the APIC setup script, specify the pod ID where the APIC node has been moved.

   a) Log in to Cisco Integrated Management Controller (CIMC).
   b) In the pod ID prompt, enter the pod ID.

   Note  Do not modify the TEP Pool address information.

Step 5  Recommission the APIC.

   a) From the menu bar, choose SYSTEM > Controllers.
   b) In the Navigation pane, expand Controllers > apic_controller_name > Cluster as Seen by Node.
   c) From the Work pane, verify in the Active Controllers summary table that the cluster Health State is Fully Fit before continuing.
   d) From the Work pane, click the decommissioned controller that displaying Unregistered in the Operational State column.
   e) From the Work pane, click Actions > Commission.
   f) In the Confirmation dialog box, click Yes.
   g) Verify that the commissioned Cisco APIC controller is in the operational state and the health state is Fully Fit.
Remote Leaf Switches

This chapter contains the following sections:

- About Remote Leaf Switches in the ACI Fabric, on page 285
- Remote Leaf Switch Hardware Requirements, on page 287
- Restrictions and Limitations, on page 288
- WAN Router and Remote Leaf Switch Configuration Guidelines, on page 289
- Configure Remote Leaf Switches Using the REST API, on page 290
- Configure Remote Leaf Switches Using the NX-OS Style CLI, on page 293
- Configure the Pod and Fabric Membership for Remote Leaf Switches Using the GUI, on page 296
- Prerequisites Required Prior to Downgrading Remote Leaf Switches, on page 300

About Remote Leaf Switches in the ACI Fabric

With an ACI fabric deployed, you can extend ACI services and APIC management to remote datacenters with Cisco ACI leaf switches that have no local spine switch or APIC attached.
The remote leaf switches are added to an existing pod in the fabric. All policies deployed in the main datacenter are deployed in the remote switches, which behave like local leaf switches belonging to the pod. In this topology, all unicast traffic is through VXLAN over Layer 3. Layer 2 Broadcast, Unknown Unicast, and Multicast (BUM) messages are sent using Head End Replication (HER) tunnels without the use of Multicast. All local traffic on the remote site is switched directly between endpoints, whether physical or virtual. Any traffic that requires use of the spine switch proxy is forwarded to the main datacenter.

The APIC system discovers the remote leaf switches when they come up. From that time, they can be managed through APIC, as part of the fabric.

---

**Note**

- All inter-VRF traffic (pre-release 4.0(1)) goes to the spine switch before being forwarded.
- Before decommissioning a remote leaf, you must first delete the vPC.

Starting in release 4.0(1), Remote Leaf behavior takes on the following characteristics:

- Reduction of WAN bandwidth use by decoupling services from spine-proxy:
  - PBR: For local PBR devices or PBR devices behind a vPC, local switching is used without going to the spine proxy. For PBR devices on orphan ports on a peer remote leaf, a RL-vPC tunnel is used. This is true when the spine link to the main DC is functional or not functional.
  - ERSPAN: For peer destination EPGs, a RL-vPC tunnel is used. EPGs on local orphan or vPC ports use local switching to the destination EPG. This is true when the spine link to the main DC is functional or not functional.
• Shared Services: Packets do not use spine-proxy path reducing WAN bandwidth consumption.
• Inter-VRF traffic is forwarded via an upstream router and not placed on the spine.
• This enhancement is only applicable for a remote leaf vPC pair. For communication across remote leaf pairs, a spine proxy is still used.
• Resolution of unknown L3 endpoints (through ToR glean process) in a remote leaf site when spine-proxy is not reachable.

You can configure Remote Leaf in the APIC GUI, either with and without a wizard, or use the REST API or the NX-OS style CLI.

**Remote Leaf Switch Hardware Requirements**

The following switches are supported for the Remote Leaf Switch feature.

**Fabric Spine Switches**

For the spine switch at the ACI Main Datacenter that is connected to the WAN router, the following spine switches are supported:

• Fixed spine switches Cisco Nexus 9000 series N9K-C9364C and N9K-C9332C
• For modular spine switches, only Cisco Nexus 9000 series switches with names that end in EX, and later (for example, N9K-X9732C-EX) are supported.
• Older generation spine switches, such as the fixed spine switch N9K-C9336PQ or modular spine switches with the N9K-X9736PQ linecard are supported in the Main Datacenter, but only next generation spine switches are supported to connect to the WAN.

**Remote Leaf Switches**

• For the remote leaf switches, only Cisco Nexus 9000 series switches with names that end in EX, and later (for example, N9K-C93180LC-EX) are supported.
• The remote leaf switches must be running a switch image of 13.1.x or later (aci-n9000-dk9.13.1.x.x.bin) before they can be discovered. This may require manual upgrades on the leaf switches.
Restrictions and Limitations

Note

In Cisco APIC Release 4.0(x), the following features are supported that were not previously:

- Q-in-Q Encapsulation Mapping for EPGs
- PBR Tracking on remote leaf switches (with system-level global GiPo enabled)
- PBR Resilient Hashing
- Netflow
- MacSec Encryption
- Troubleshooting Wizard
- Atomic counters

Stretching of L3out SVI between local leaf switches (ACI main data center switches) and remote leaf switches is not supported.

The following deployments and configurations are not supported with the remote leaf switch feature:

- Spanning Tree Protocol across remote leaf site and main data center
- APIC controllers directly connected to remote leaf switches
- Orphan port-channel or physical ports on remote leaf switches, with a vPC domain (this restriction applies for releases 3.1 and earlier)
- With and without service node integration, local traffic forwarding within a remote location is only supported if the consumer, provider, and services nodes are all connected to Remote Leaf switches are in vPC mode

Full fabric and tenant policies are supported on remote leaf switches, in this release, except for the following features:

- ACI Multi-Site
- Layer 2 Outside Connections (except Static EPGs)
- 802.1Q Tunnels
- Copy services with vzAny contract
- FCoE connections on remote leaf switches
- Flood in encapsulation for bridge domains or EPGs
- Fast Link Failover policies
- Managed Service Graph-attached devices at remote locations
- Traffic Storm Control
- Cloud Sec Encryption
- First Hop Security
- PTP
- Layer 3 Multicast routing on remote leaf switches
- Maintenance mode
- TEP to TEP atomic counters
- Transit L3Out across remote locations, which is when the main Cisco ACI datacenter pod is a transit between two remote locations (the L3Out in RL `location-1` and L3Out in RL `location-2` are advertising prefixes for each other)
- Traffic forwarding directly across two remote leaf vPC pairs in the same remote datacenter or across datacenters

### WAN Router and Remote Leaf Switch Configuration Guidelines

Before a remote leaf is discovered and incorporated in APIC management, you must configure the WAN router and the remote leaf switches.

Configure the WAN routers that connect to the fabric spine switch external interfaces and the remote leaf switch ports, with the following requirements:

**WAN Routers**

- Enable OSPF on the interfaces, with the same details, such as area ID, type, and cost.
- Configure DHCP Relay on the interface leading to each APIC’s IP address in the main fabric.
- The interfaces on the WAN routers which connect to the VLAN-5 interfaces on the spine switches must be on different VRFs than the interfaces connecting to a regular multipod network.

**Remote Leaf Switches**

- Connect the remote leaf switches to an upstream router by a direct connection from one of the fabric ports. The following connections to the upstream router are supported:
  - 40 Gbps & higher connections
  - With a QSFP-to-SFP Adapter, supported 1G/10G SFPs

Bandwidth in the WAN must be a minimum of 100 Mbps and maximum supported latency is 300 msecs.

- It is recommended, but not required to connect the pair of remote leaf switches with a vPC. The switches on both ends of the vPC must be remote leaf switches at the same remote datacenter.
- Configure the northbound interfaces as Layer 3 sub-interfaces on VLAN-4, with unique IP addresses.

  If you connect more than one interface from the remote leaf switch to the router, configure each interface with a unique IP address.

- Enable OSPF on the interfaces, but do not set the OSPF area type as stub area.
- The IP addresses in the remote leaf switch TEP Pool subnet must not overlap with the pod TEP subnet pool. The subnet used must be /24 or lower.
• Multipod is supported, but not required, with the Remote Leaf feature.

• When connecting a pod in a single-pod fabric with remote leaf switches, configure an L3Out from a spine switch to the WAN router and an L3Out from a remote leaf switch to the WAN router, both using VLAN-4 on the switch interfaces.

• When connecting a pod in a multipod fabric with remote leaf switches, configure an L3Out from a spine switch to the WAN router and an L3Out from a remote leaf switch to the WAN router, both using VLAN-4 on the switch interfaces. Also configure a multipod-internal L3Out using VLAN-5 to support traffic that crosses pods destined to a remote leaf switch. The regular multipod and multipod-internal connections can be configured on the same physical interfaces, as long as they use VLAN-4 and VLAN-5.

• When configuring the Multipod-internal L3Out, use the same router ID as for the regular multipod L3Out, but deselect the **Use Router ID as Loopback Address** option for the router-id and configure a different loopback IP address. This enables ECMP to function.

## Configure Remote Leaf Switches Using the REST API

To enable Cisco APIC to discover and connect the IPN router and remote leaf switches, perform the steps in this topic.

This example assumes that the remote leaf switches are connected to a pod in a multipod topology. It includes two L3Outs configured in the infra tenant, with VRF overlay-1:

- One is configured on VLAN-4, that is required for both the remote leaf switches and the spine switch connected to the WAN router.
- One is the multipod-internal L3Out configured on VLAN-5, that is required for the multipod and Remote Leaf features, when they are deployed together.

### Procedure

#### Step 1
To define the TEP pool for two remote leaf switches to be connected to a pod, send a post with XML such as the following example:

**Example:**

```xml
<fabricSetupPol>
  <fabricSetupP tepPool="10.0.0.0/16" podId="1"/>
  <fabricExtSetupP tepPool="30.0.128.0/20" extPoolId="1"/>
</fabricSetupPol>

<fabricSetupPol>
  <fabricSetupP tepPool="10.1.0.0/16" podId="2"/>
  <fabricExtSetupP tepPool="30.1.128.0/20" extPoolId="1"/>
</fabricSetupPol>
```

#### Step 2
To define the node identity policy, send a post with XML, such as the following example:

**Example:**

```xml
<fabricNodeIdentPol>
  <fabricNodeIdentP serial="SAL17267Z7W" name="leaf1" nodeID="101" podID="1" extPoolId="1" nodeType="remote-leaf-wan"/>
  <fabricNodeIdentP serial="SAL17267Z7X" name="leaf2" nodeID="102" podID="1" extPoolId="1" nodeType="remote-leaf-wan"/>
</fabricNodeIdentPol>
```
Step 3
To configure the Fabric External Connection Profile, send a post with XML such as the following example:

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<imdata totalCount="1">
    <fvFabricExtConnP dn="uni/tn-infra/fabricExtConnP-1" id="1" name="Fabric_Ext_Conn_Pol1" rt="extended:as2-nn4:5:16" siteId="0">
        <l3extFabricExtRoutingP name="test">
            <l3extSubnet ip="150.1.0.0/16" scope="import-security"/>
        </l3extFabricExtRoutingP>
        <l3extFabricExtRoutingP name="ext_routing_prof_1">
            <l3extSubnet ip="204.1.0.0/16" scope="import-security"/>
            <l3extSubnet ip="209.2.0.0/16" scope="import-security"/>
            <l3extSubnet ip="202.1.0.0/16" scope="import-security"/>
            <l3extSubnet ip="207.1.0.0/16" scope="import-security"/>
            <l3extSubnet ip="200.0.0.0/8" scope="import-security"/>
            <l3extSubnet ip="201.2.0.0/16" scope="import-security"/>
            <l3extSubnet ip="210.2.0.0/16" scope="import-security"/>
            <l3extSubnet ip="201.1.0.0/16" scope="import-security"/>
            <l3extSubnet ip="203.2.0.0/16" scope="import-security"/>
            <l3extSubnet ip="208.1.0.0/16" scope="import-security"/>
            <l3extSubnet ip="207.2.0.0/16" scope="import-security"/>
            <l3extSubnet ip="100.0.0.0/8" scope="import-security"/>
            <l3extSubnet ip="201.1.0.0/16" scope="import-security"/>
            <l3extSubnet ip="203.1.0.0/16" scope="import-security"/>
            <l3extSubnet ip="208.2.0.0/16" scope="import-security"/>
        </l3extFabricExtRoutingP>
    </fvFabricExtConnP>
</imdata>
```

Step 4
To configure an L3Out on VLAN-4, that is required for both the remote leaf switches and the spine switch connected to the WAN router, enter XML such as the following example:

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<polUni>
    <fvTenant name="infra">
        <l3extOut name="rleaf-wan-test">
            <ospfExtP areaId="0.0.0.5"/>
            <bgpExtP/>
            <l3extRsEctx tnFvCtxName="overlay-1"/>
            <l3extRsL3DomAtt tDn="uni/l3dom-l3extDom1"/>
            <l3extProvLbl descr="" name="prov_mp1" ownerKey="" ownerTag="" tag="yellow-green"/>
            <l3extLNodeP name="rleaf-101">
            </l3extLNodeP>
            <l3extRsL3OutAtt>
                <l3extTlfP name="portIf">
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Step 5

To configure the multipod L3Out on VLAN-5, that is required for both multipod and the remote leaf topology, send a post such as the following example:

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<polUni>
  <fvTenant name="infra">
    <l3extOut name="ipn-multipodInternal">
      <ospfExtP areaCtrl="inherit-ipsec,redistribute,summary" areaId="0.0.0.5" multipodInternal="yes" />
      <l3extRsEctx tnFvCtxName="overlay-1" />
    </l3extOut>
    <l3extLNodeP name="bLeaf">
      </l3extRsNodeL3OutAtt>
      <l3extRsNodeL3OutAtt rtrId="102.102.102.102" rtrIdLoopBack="no" tDn="topology/pod-1/node-102">
        <l3extLoopBackIfP addr="102.102.102.112" />
      </l3extRsNodeL3OutAtt>
      <l3extLIfP name="portIf">
        <osspfIfP authKeyId="1" authType="none">
          <osspfRsIfPol tnOspfIfPolName='ospfIfPol'/>
        </ospfIfP>
        <l3extRsPathL3OutAtt addr="10.0.254.233/30" encap='vlan-5' />
      </l3extLIfP>
    </l3extLNodeP>
  </fvTenant>
</polUni>
```
Configure Remote Leaf Switches Using the NX-OS Style CLI

This example configures a spine switch and a remote leaf switch to enable the leaf switch to communicate with the main fabric pod.

**Before you begin**
- The IPN router and remote leaf switches are active and configured; see [WAN Router and Remote Leaf Switch Configuration Guidelines](#), on page 289.
- The remote leaf switches are running a switch image of 13.1.x or later (aci-n9000-dk9.13.1.x.x.bin).
- The pod in which you plan to add the remote leaf switches is created and configured.

**Procedure**

**Step 1**
Define the TEP pool for a remote location 5, in pod 2.
The network mask must be /24 or lower.
Use the following new command: `system remote-leaf-site site-id pod pod-id tep-pool ip-address-and-netmask`

**Example:**
apic1(config)# system remote-leaf-site 5 pod 2 tep-pool 192.0.0.0/16

**Step 2**
Add a remote leaf switch to pod 2, remote-leaf-site 5.
Use the following command: `system switch-id serial-number node-id leaf-switch-name pod pod-id remote-leaf-site remote-leaf-site-id node-type remote-leaf-wan`

**Example:**
apic1(config)# system switch-id FDO210805SKD 109 ifav4-leaf9 pod 2 remote-leaf-site 5 node-type remote-leaf-wan

**Step 3**
Configure a VLAN domain with a VLAN that includes VLAN 4.

**Example:**
apic1(config)# vlan-domain ospfDom
apic1(config-vlan)# vlan 4-5
apic1(config-vlan)# exit

**Step 4**
Configure two L3Outs for the infra tenant, one for the remote leaf connections and one for the multipod IPN.

**Example:**
apic1(config)# tenant infra
apic1(config-tenant)# l3out ri-wan
apic1(config-tenant-l3out)# vrf member overlay-1
Configure Remote Leaf Switches Using the NX-OS Style CLI

Remote Leaf Switches

Step 5 Configure the spine switch interfaces and sub-interfaces to be used by the L3Outs.

Example:

apic1(config)# spine 201
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-vrf)# exit
apic1(config-spine)# interface ethernet 8/36
apic1(config-spine-if)# vlan-domain member ospfDom
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#

Step 6 Configure the remote leaf switch interface and sub-interface used for communicating with the main fabric pod.

Example:

(config)# leaf 101
apic1(config-leaf)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-vrf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49
apic1(config-leaf-if)# vlan-domain member ospfDom
apic1(config-leaf-if)# exit
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-leaf-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49.4
apic1(config-leaf-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
Example

The following example provides a downloadable configuration:

```
apic1# configure
apic1(config)# system remote-leaf-site 5 pod 2 tep-pool 192.0.0.0/16
apic1(config)# system switch-id FDO210805SKD 109 ifav4-leaf9 pod 2
apic1(config)# vlan-domain ospfDom
apic1(config-vlan)# vlan 4-5
apic1(config-vlan)# exit
apic1(config)# tenant infra
apic1(config-tenant)# l3out rl-wan-test
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# l3out ipn-multipodInternal
apic1(config-tenant-l3out)# vrf member overlay-1
apic1(config-tenant-l3out)# exit
apic1(config-tenant)# exit
apic1(config)#
apic1(config)# spine 201
apic1(config-spine)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-vrf)# exit
apic1(config-spine)# interface ethernet 8/36
apic1(config-spine-if)# vlan-domain member ospfDom
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf default
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.4
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-spine-if)# ip router ospf default area 5
apic1(config-spine-if)# exit
apic1(config-spine)# router ospf multipod-internal
apic1(config-spine-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-spine-ospf-vrf)# area 5 l3out ipn-multipodInternal
apic1(config-spine-ospf-vrf)# exit
apic1(config-spine-ospf)# exit
apic1(config-spine)#
apic1(config-spine)# interface ethernet 8/36.5
apic1(config-spine-if)# vrf member tenant infra vrf overlay-1 l3out ipn-multipodInternal
apic1(config-spine-if)# ip router ospf multipod-internal area 5
apic1(config-spine-if)# exit
apic1(config-spine)# exit
apic1(config)#
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-vrf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49
apic1(config-leaf-if)# vlan-domain member ospfDom
apic1(config-leaf-if)# exit
```
apic1(config-leaf)# router ospf default
apic1(config-leaf-ospf)# vrf member tenant infra vrf overlay-1
apic1(config-leaf-ospf-vrf)# area 5 l3out rl-wan-test
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)#
apic1(config-leaf)# interface ethernet 1/49.4
apic1(config-leaf-if)# vrf member tenant infra vrf overlay-1 l3out rl-wan-test
apic1(config-leaf-if)# ip router ospf default area 5
apic1(config-leaf-if)# exit

Configure the Pod and Fabric Membership for Remote Leaf Switches Using the GUI

You can configure and enable Cisco APIC to discover and connect the IPN router and remote switches, either by using a wizard or by using the APIC GUI, without a wizard.

Configure the Pod and Fabric Membership for Remote Leaf Switches Using a Wizard

You can configure and enable Cisco APIC to discover and connect the IPN router and remote switches, using a wizard as in this topic, or in an alternative method using the APIC GUI. See Configure the Pod and Fabric Membership for Remote Leaf Switches Using the GUI (Without a Wizard), on page 297

Before you begin

• The IPN and WAN routers and remote leaf switches are active and configured; see WAN Router and Remote Leaf Switch Configuration Guidelines, on page 289.
• The remote leaf switch pair are connected with a vPC.
• The remote leaf switches are running a switch image of 13.1.x or later (aci-n9000-dk9.13.1.x.x.bin).
• The pod in which you plan to add the remote leaf switches is created and configured.
• The spine switch that will be used to connect the pod with the remote leaf switches is connected to the IPN router.

Procedure

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>On the menu bar click Fabric &gt; Inventory.</td>
</tr>
<tr>
<td>Step 2</td>
<td>In the Navigation pane, expand Quick Start and click Node or Pod Setup.</td>
</tr>
<tr>
<td>Step 3</td>
<td>In the Remote Leaf pane of the working pane, click Setup Remote Leaf or right-click Node or Pod Setup and click Setup Remote Leaf.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Follow the instructions to configure the following:</td>
</tr>
<tr>
<td></td>
<td>• Pod Fabric—Identify the pod and the TEP Pool subnet for the remote leaf switches.</td>
</tr>
<tr>
<td></td>
<td>Add the comma-separated subnets for the underlay routes leading to the remote leaf switches.</td>
</tr>
</tbody>
</table>
Repeat this for the other remote leaf switches to be added to the pod.

- **Fabric Membership**—Set up fabric membership for the remote leaf switches, including the node ID, Remote Leaf TEP Pool ID, and Remote Leaf Switch name.

- **Remote Leaf**—Configure Layer 3 details for the remote leaf switches, including the OSPF details (the same OSPF configuration as in the WAN router), the router IDs and loopback addresses, and routed sub-interfaces for nodes.

- **Connections**—Configure the Layer 3 details for the spine switch for the L3Out on the route to the remote leaf switches (only required if you are adding remote leaf switches to a single-pod fabric), including the OSPF details (same as configured in the IPN and WAN routers), the OSPF Profile, router IDs and routed sub-interfaces for the spine switches.

---

**Configure the Pod and Fabric Membership for Remote Leaf Switches Using the GUI (Without a Wizard)**

You can configure remote leaf switches using this GUI procedure, or use a wizard. For the wizard procedure, see Configure the Pod and Fabric Membership for Remote Leaf Switches Using a Wizard, on page 296

**Before you begin**

- The routers (IPN and WAN) and remote leaf switches are active and configured; see WAN Router and Remote Leaf Switch Configuration Guidelines, on page 289.

- The remote leaf switches are running a switch image of 13.1.x or later (aci-n9000-dk9.13.1.x.x.bin).

- The pod in which you plan to add the remote leaf switches is created and configured.

- The spine switch that will be used to connect the pod with the remote leaf switches is connected to the IPN router.

**Procedure**

**Step 1**

Configure the TEP pool for the remote leaf switches, with the following steps:

a) On the menu bar, click **Fabric > Inventory**.

b) In the Navigation pane, click **Pod Fabric Setup Policy**.

c) On the **Fabric Setup Policy** panel, double-click the pod where you want to add the pair of remote leaf switches.

d) Click the + on the **Remote Pools** table.

e) Enter the remote ID and a subnet for the remote TEP pool and click **Submit**.

f) On the **Fabric Setup Policy** panel, click **Submit**.

**Step 2**

Configure the L3Out for the spine switch connected to the IPN router, with the following steps:

a) On the menu bar, click **Tenants > infra**.

b) In the Navigation pane, expand **Networking**, right-click **External Routed Networks**, and choose **Create Routed Outside**.
c) Enter a name for the L3Out.
d) Click the OSPF checkbox to enable OSPF, and configure the OSPF details the same as on the IPN and WAN routers.
e) Only check the Enable Remote Leaf check box, if the pod where you are adding the remote leaf switches is part of a multipod fabric.

This option enables a second OSPF instance using VLAN-5 for multipod, which ensures that routes for remote leaf switches are only advertised within the pod they belong to.

f) Choose the overlay-1 VRF.

Step 3
Configure the details for the spine and the interfaces used in the L3Out, with the following steps:
a) Click the + on the Nodes and Interfaces Protocol Profiles table.
b) Enter the node profile name.
c) Click the + on the Nodes table, enter the following details.
   • Node ID—ID for the spine switch that is connected to the IPN router.
   • Router ID—IP address for the IPN router
   • External Control Peering—disable if the pod where you are adding the remote leaf switches is in a single-pod fabric
d) Click OK.
e) Click the + on the OSPF Interface Profiles table.
f) Enter the name of the interface profile and click Next.
g) Under OSPF Profile, click OSPF Policy and choose a previously created policy or click Create OSPF Interface Policy.
h) Click Next.
i) Click Routed Sub-Interface, click the + on the Routed Sub-Interfaces table, and enter the following details:
   • Node—Spine switch where the interface is located.
   • Path—Interface connected to the IPN router
   • Encap—Enter 4 for the VLAN
j) Click OK and click Next.
k) Click the + on the External EPG Networks table.
l) Enter the name of the external network, and click OK.
m) Click Finish.

Step 4
To complete the fabric membership configuration for the remote leaf switches, perform the following steps:
a) Navigate to Fabric > Inventory > Fabric Membership.

At this point, the new remote leaf switches should appear in the list of switches registered in the fabric. However, they are not recognized as remote leaf switches until you configure the Node Identity Policy, with the following steps.

b) For each remote leaf switch, double-click on the node in the list, configure the following details, and click Update:
   • Node ID—Remote leaf switch ID
Remote Leaf Switches

Configure the Pod and Fabric Membership for Remote Leaf Switches Using the GUI (Without a Wizard)

- RL TEP Pool—Identifier for the remote leaf TEP pool, that you previously configured
- Node Name—Name of the remote leaf switch

After you configure the Node Identity Policy for each remote leaf switch, it is listed in the Fabric Membership table with the role remote leaf.

Step 5
Configure the L3Out for the remote leaf location, with the following steps:
   a) Navigate to Tenants > infra > Networking.
   b) Right-click External Routed Networks, and choose Create Routed Outside.
   c) Enter a name for the L3Out.
   d) Click the OSPF checkbox to enable OSPF, and configure the OSPF details the same as on the IPN and WAN router.
   e) Only check the Enable Remote Leaf check box, if the pod where you are adding the remote leaf switches is part of a multipod fabric.
   f) Choose the overlay-1 VRF.

Step 6
Configure the nodes and interfaces leading from the remote leaf switches to the WAN router, with the following steps:
   a) In the Create Routed Outside panel, click the + on the Nodes and Interfaces Protocol Profiles table.
   b) Click the + on the Nodes table and enter the following details:
      - Node ID—ID for the remote leaf that is connected to the WAN router
      - Router ID—IP address for the WAN router
      - External Control Peering—only enable if the remote leaf switches are being added to a pod in a multipod fabric
   c) Click OK.
   d) Click on the + on OSPF Interface Profiles, and configure the following details for the routed sub-interface used to connect a remote leaf switch with the WAN router.
      - Identity—Name of the OSPF interface profile
      - Protocol Profiles—A previously configured OSPF profile or create one
      - Interfaces—On the Routed Sub-Interface tab, the path and IP address for the routed sub-interface leading to the WAN router

Step 7
Configure the Fabric External Connection Profile, with the following steps:
   a) Navigate to Tenants > infra > Policies > Protocol.
   b) Right-click Fabric Ext Connection Policies and choose Create Intrasite/Intersite Profile.
   c) Enter the mandatory Community value in the format provided in the example.
   d) Click the + on Fabric External Routing Profile.
   e) Enter the name of the profile and add uplink interface subnets for all of the remote leaf switches.
   f) Click Update and click Submit.

Step 8
To verify that the remote leaf switches are discovered by the APIC, navigate to Fabric > Inventory > Fabric Membership, or Fabric > Inventory > Pod > Topology.

Step 9
To view the status of the links between the fabric and the remote leaf switches, enter the show ip ospf neighbors vrf overlay-1 command on the spine switch that is connected to the IPN router.
Step 10
To view the status of the remote leaf switches in the fabric, enter the `acidiag fnvread` NX-OS style command on the APIC using the CLI.

---

Prerequisites Required Prior to Downgrading Remote Leaf Switches

**Note**
If you have remote leaf switches deployed, if you downgrade the APIC software from Release 3.1(1) or later, to an earlier release that does not support the Remote Leaf feature, you must decommission the remote nodes and remove the remote leaf-related policies (including the TEP Pool), before downgrading. For more information on decommissioning switches, see *Decommissioning and Recommissioning Switches* in the *Cisco APIC Troubleshooting Guide*.

Before you downgrade remote leaf switches, verify that the followings tasks are complete:

- Delete the vPC domain.
- Delete the vTEP - Virtual Network Adapter if using SCVMM.
- Decommission the remote leaf nodes, and wait 10 -15 minutes after the decommission for the task to complete.
- Delete the remote leaf to WAN L3out in the infra tenant.
- Delete the infra-L3out with VLAN 5 if using Multipod.
- Delete the remote TEP pools.
Transit Routing

This chapter contains the following sections:

- Transit Routing in the ACI Fabric, on page 301
- Transit Routing Use Cases, on page 302
- Supported Transit Combination Matrix, on page 307
- Transit Routing Guidelines, on page 309
- Configuring Transit Routing, on page 319

Transit Routing in the ACI Fabric

The Cisco APIC software supports external Layer 3 connectivity with OSPF (NSSA) and iBGP. The fabric advertises the tenant bridge domain subnets out to the external routers on the External Layer 3 Outside (L3Out) connections. The routes that are learned from the external routers are not advertised to other external routers. The fabric behaves like a stub network that can be used to carry the traffic between the external Layer 3 domains.
In transit routing, multiple L3Out connections within a single tenant and VRF are supported and the APIC advertises the routes that are learned from one L3Out connection to another L3Out connection. The external Layer 3 domains peer with the fabric on the border leaf switches. The fabric is a transit Multiprotocol-Border Gateway Protocol (MP-BGP) domain between the peers.

The configuration for external L3Out connections is done at the tenant and VRF level. The routes that are learned from the external peers are imported into MP-BGP at the ingress leaf per VRF. The prefixes that are learned from the L3Out connections are exported to the leaf switches only where the tenant VRF is present.

Note

For cautions and guidelines for configuring transit routing, see Guidelines for Transit Routing, on page 309

Transit Routing Use Cases

Transit Routing Between Layer 3 Domains

Multiple Layer 3 domains such as external pods, mainframes, service nodes, or WAN routers can peer with the ACI fabric to provide transit functionality between them.
Mainframe Traffic Transiting the ACI Fabric

Mainframes can function as IP servers running standard IP routing protocols that accommodate requirements from Logical Partitions (LPARs) and Virtual IP Addressing (VIPA).
In this topology, mainframes require the ACI fabric to be a transit domain for external connectivity through a WAN router and for east-west traffic within the fabric. They push host routes to the fabric to be redistributed within the fabric and out to external interfaces.

**Service Node Transit Connectivity**

Service nodes can peer with the ACI fabric to advertise a Virtual IP (VIP) route that is redistributed to an external WAN interface.
The VIP is the external facing IP address for a particular site or service. A VIP is tied to one or more servers or nodes behind a service node.

**Multipod in a Transit-Routed Configuration**

In a multipod topology, the fabric acts as a transit for external connectivity and interconnection between multiple pods. Cloud providers can deploy managed resource pods inside a customer datacenter. The demarcation point can be an L3Out with OSPF or BGP peering with the fabric.
In such scenarios, the policies are administered at the demarcation points and ACI policies need not be imposed.

Layer 4 to Layer 7 route peering is a special use case of the fabric as a transit where the fabric serves as a transit OSPF or BGP domain for multiple pods. You configure route peering to enable OSPF or BGP peering on the Layer 4 to Layer 7 service device so that it can exchange routes with the leaf node to which it is connected. A common use case for route peering is Route Health Injection where the SLB VIP is advertised over OSPF or iBGP to clients outside the fabric. See *L4-L7 Route Peering with Transit Fabric - Configuration Walkthrough* for a configuration walk-through of this scenario.

**GOLF in a Transit-Routed Configuration**

In APIC, release 2.0 and later, the Cisco ACI supports transit routing with GOLF L3Outs (with BGP and OSPF). For example, the following diagram shows traffic transiting the fabric with GOLF L3Outs and a border leaf L3Out.
### Supported Transit Combination Matrix

<table>
<thead>
<tr>
<th>Layer 3 Outside Connection Type</th>
<th>OSPF</th>
<th>iBGP</th>
<th>eBGP</th>
<th>EIGRP v4</th>
<th>EIGRP v6</th>
<th>Static Route</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layer 3 Outside Connection Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OSPF</td>
<td>Yes</td>
<td>Yes*</td>
<td>Yes</td>
<td>Yes* (tested in APIC release 1.3c)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Figure 36: GOLF L3Outs and a Border Leaf L3Out in a Transit-Routed Configuration*
<table>
<thead>
<tr>
<th>Layer 3 Outside Connection Type</th>
<th>OSPF</th>
<th>iBGP over OSPF</th>
<th>iBGP over Static Route</th>
<th>iBGP over Direct Connection</th>
<th>eBGP over OSPF</th>
<th>eBGP over Static Route</th>
<th>eBGP over Direct Connection</th>
<th>EIGRPv4</th>
<th>EIGRPv6</th>
<th>Static Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>iBGP</td>
<td>iBGP</td>
<td>Yes*</td>
<td>X</td>
<td>X</td>
<td>Yes*</td>
<td>X</td>
<td>(tested in APIC release 1.3c)</td>
<td>Yes</td>
<td>Yes</td>
<td>X</td>
</tr>
<tr>
<td>iBGP over Static Route</td>
<td></td>
<td>Yes</td>
<td>X</td>
<td>X</td>
<td>Yes*</td>
<td>X</td>
<td>(tested in APIC release 1.2g)</td>
<td>Yes</td>
<td>X</td>
<td>Yes</td>
</tr>
<tr>
<td>iBGP over Direct Connection</td>
<td>Yes*</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Yes</td>
<td>-</td>
<td>(tested in APIC release 1.2g)</td>
<td>Yes</td>
<td>Yes</td>
<td>X</td>
</tr>
<tr>
<td>eBGP</td>
<td>eBGP</td>
<td>Yes* (tested in APIC release 1.3c)</td>
<td>Yes* (tested in APIC release 1.3c)</td>
<td>Yes* (tested in APIC release 1.3c)</td>
<td>Yes</td>
<td>Yes* (tested in APIC release 1.3c)</td>
<td>(tested in APIC release 1.2g)</td>
<td>Yes</td>
<td>X</td>
<td>Yes</td>
</tr>
<tr>
<td>eBGP over Static Route</td>
<td>Yes*</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Yes</td>
<td>Yes (tested in APIC release 3.0)</td>
<td>(tested in APIC release 1.2g)</td>
<td>Yes</td>
<td>X</td>
<td>Yes</td>
</tr>
<tr>
<td>eBGP over Direct Connection</td>
<td>Yes</td>
<td>Yes (tested in APIC release 1.3c)</td>
<td>Yes (tested in APIC release 1.3c)</td>
<td>Yes (tested in APIC release 1.3c)</td>
<td>Yes*</td>
<td>Yes (tested in APIC release 1.3c)</td>
<td>(tested in APIC release 1.3c)</td>
<td>Yes</td>
<td>Yes</td>
<td>X</td>
</tr>
<tr>
<td>EIGRPv4</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (tested in APIC release 1.3c)</td>
<td>(tested in APIC release 1.3c)</td>
<td>Yes</td>
<td>X</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Transit Routing Guidelines

Guidelines for Transit Routing

Use the following guidelines when creating and maintaining transit routing connections:

- connec. = connection
- *= Not supported on the same leaf switch
- X = Unsupported/Untested combinations

---

<table>
<thead>
<tr>
<th>Layer 3 Outside Connection Type</th>
<th>OSPF</th>
<th>iBGP over OSPF</th>
<th>iBGP over Direct Connection</th>
<th>eBGP over OSPF</th>
<th>eBGP over Direct Connection</th>
<th>EIGRPv4</th>
<th>EIGRPv6</th>
<th>Static Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIGRPv6</td>
<td>Yes (tested in APIC release 1.2g)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>Yes (tested in APIC release 1.3c)</td>
</tr>
<tr>
<td>Static Route</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (tested in APIC release 1.3c)</td>
</tr>
</tbody>
</table>
Before APIC, release 2.3(1f), transit routing was not supported within a single L3Out profile. In APIC, release 2.3(1f) and later, you can configure transit routing with a single L3Out profile, with the following limitations:

- If the VRF is unenforced, an external subnet (l3extSubnet) of 0.0.0.0/0 can be used to allow traffic between the routers sharing the same L3EPG.

- If the VRF is enforced, an external default subnet (0.0.0.0/0) **cannot** be used to match both source and destination prefixes for traffic within the same Layer 3 EPG. To match all traffic within the same Layer 3 EPG, the following prefixes are supported:

  - **IPv4**
    - 0.0.0.0/1—with External Subnets for the External EPG
    - 128.0.0.0/1—with External Subnets for the External EPG
    - 0.0.0.0/0—with Import Route Control Subnet, Aggregate Import

  - **IPv6**
    - 0::0/1—with External Subnets for the External EPG
    - 8000::0/1—with External Subnets for the External EPG
    - 0:0/0—with Import Route Control Subnet, Aggregate Import

- Alternatively, a single default subnet (0.0.0.0/0) can be used when combined with a VzAny contract. For example:

  - Use a VzAny provided contract and an Layer 3 EPG consumed contract (matching 0.0.0.0/0), or a VzAny consumed contract and Layer 3 EPG provided contract (matching 0.0.0.0/0).

  - Use the subnet 0.0.0.0/0—with Import/Export Route Control Subnet, Aggregate Import, and Aggregate Export.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Caution or Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared Routes: Differences in Hardware Support</td>
<td>Routes shared between VRFs function correctly on Generation 2 switches (Cisco Nexus N9K switches with &quot;EX&quot; or &quot;FX&quot; on the end of the switch model name, or later; for example, N9K-93108TC-EX). On Generation 1 switches, however, there may be dropped packets with this configuration, because the physical ternary content-addressable memory (TCAM) tables that store routes do not have enough capacity to fully support route parsing.</td>
</tr>
</tbody>
</table>

| OSPF or EIGRP in Back to Back Configuration | Cisco APIC supports transit routing in export route control policies that are configured on the L3Out. These policies control which transit routes (prefixes) are redistributed into the routing protocols in the L3Out. When these transit routes are redistributed into OSPF or EIGRP, they are tagged 4294967295 to prevent routing loops. The Cisco ACI fabric does not accept routes matching this tag when learned on an OSPF or EIGRP L3Out. However, in the following cases, it is necessary to override this behavior:  
  - When connecting two Cisco ACI fabrics using OSPF or EIGRP.  
  - When connecting two different VRFs in the same Cisco ACI fabric using OSPF or EIGRP.  

Where an override is required, you must configure the VRF with a different tag policy at the following APIC GUI location:  
Tenant > Tenant_name > Networking > Protocol Policies > Route Tag. Apply a different tag.  

In addition to creating the new route-tag policy, update the VRF to use this policy at the following APIC GUI location:  
Tenant > Tenant_name > Networking > VRFs > Tenant_VRF. Apply the route tag policy that you created to the VRF.  

Note When multiple L3Outs or multiple interfaces in the same L3Out are deployed on the same leaf switch and used for transit routing, the routes are advertised within the IGP (not redistributed into the IGP). In this case the route-tag policy does not apply.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Caution or Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advertising BD Subnets Outside the Fabric</td>
<td>The import and export route control policies only apply to the transit routes (the routes that are learned from other external peers) and the static routes. The subnets internal to the fabric that are configured on the tenant BD subnets are not advertised out using the export policy subnets. The tenant subnets are still permitted using the IP prefix-lists and the route-maps but they are implemented using different configuration steps. See the following configuration steps to advertise the tenant subnets outside the fabric:&lt;br&gt;&lt;br&gt;1. Configure the tenant subnet scope as <strong>Public Subnet</strong> in the subnet properties window.&lt;br&gt;2. Optional. Set the Subnet Control as <strong>ND RA Prefix</strong> in the subnet properties window.&lt;br&gt;3. Associate the tenant bridge domain (BD) with the external Layer 3 Outside (L3Out).&lt;br&gt;4. Create contract (provider or consumer) association between the tenant EPG and the external EPG. Setting the BD subnet to Public scope and associating the BD to the L3Out creates an IP prefix-list and the route-map sequence entry on the border leaf for the BD subnet prefix.</td>
</tr>
<tr>
<td>Topic</td>
<td>Caution or Guideline</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Advertising a Default Route    | For external connections to the fabric that only require a default route, there is support for originating a default route for OSPF, EIGRP, and BGP L3Out connections. If a default route is received from an external peer, this route can be redistributed out to another peer following the transit export route control as described earlier in this article. A default route can also be advertised out using a Default Route Leak policy. This policy supports advertising a default route if it is present in the routing table or it always supports advertising a default route. The Default Route Leak policy is configured in the L3Out connection. When creating a Default Route Leak policy, follow these guidelines:  
  - For BGP, the Always property is not applicable.  
  - For BGP, when configuring the Scope property, choose Outside.  
  - For OSPF, the scope value Context creates a type-5 LSA while the Scope value Outside creates type-7 LSA. Your choice depends on the area type configured in the L3Out. If the area type is Regular, set the scope to Context. If the area type is NSSA, set the scope to Outside.  
  - For EIGRP, when choosing the Scope property, you must choose Context. |
| MTU                            | Cisco ACI does not support IP fragmentation. Therefore, when you configure Layer 3 Outside (L3Out) connections to external routers, or multipod connections through an Inter-Pod Network (IPN), it is critical that the MTU is set appropriately on both sides. On some platforms, such as ACI, Cisco NX-OS, and Cisco IOS, the configurable MTU value takes into account the IP headers (resulting in a max packet size to be set as 9216 bytes for ACI and 9000 for NX-OS and IOS). However, other platforms such as IOS-XR configure the MTU value exclusive of packet headers (resulting in a max packet size of 8986 bytes).  
  For the appropriate MTU values for each platform, see the relevant configuration guides.  
  Cisco highly recommends you test the MTU using CLI-based commands. For example, on the Cisco NX-OS CLI, use a command such as ping 1.1.1.1 df-bit packet-size 9000 source-interface ethernet 1/1. |
Transit Route Control

A route transit is defined to import traffic through a Layer 3 outside network `l3extOut` profile (`l3extInstP`), where it is to be imported. A different route transit is defined to export traffic through another `l3extInstP` where it is to be exported.

Since multiple `l3extOut` policies can be deployed on a single node or multiple nodes in the fabric, a variety of protocol combinations are supported. Every protocol combination can be deployed on a single node using multiple `l3extOut` policies or multiple nodes using multiple `l3extOut` policies. Deployments of more than two protocols in different `l3extOut` policies in the fabric are supported.

Export route-maps are made up of prefix-list matches. Each prefix-list consists of bridge domain (BD) public subnet prefixes in the VRF and the export prefixes that need to be advertised outside.

Route control policies are defined in an `l3extOut` policy and controlled by properties and relations associated with the `l3extOut`. APIC uses the `enforceRtctrl` property of the `l3extOut` to enforce route control directions. The default is to enforce control on export and allow all on import. Imported and exported routes (`l3extSubnets`), are defined in the `l3extInstP`. The default scope for every route is import. These are the routes and prefixes which form a prefix-based EPG.

All the import routes form the import route map and are used by BGP and OSPF to control import. All the export routes form the export route map used by OSPF and BGP to control export.

Import and export route control policies are defined at different levels. All IPv4 policy levels are supported for IPv6. Extra relations that are defined in the `l3extInstP` and `l3extSubnet` MOs control import.

Default route leak is enabled by defining the `l3extDefaultRouteLeakP` MO under the `l3extOut`. `l3extDefaultRouteLeakP` can have Virtual Routing and Forwarding (VRF) scope or `l3extOut` scope per area for OSPF and per peer for BGP.

The following set rules provide route control:

- `rtctrlSetPref`
- `rtctrlSetRtMetric`
- `rtctrlSetRtMetricType`

Additional syntax for the `rtctrlSetComm` MO includes the following:

- `no-advertise`
- `no-export`
- `no-peer`

BGP

The ACI fabric supports BGP peering with external routers. BGP peers are associated with an `l3extOut` policy and multiple BGP peers can be configured per `l3extOut`. BGP can be enabled at the `l3extOut` level by defining the `bgpExtP` MO under an `l3extOut`.

Note

Although the `l3extOut` policy contains the routing protocol (for example, BGP with its related VRF), the L3Out interface profile contains the necessary BGP interface configuration details. Both are needed to enable BGP.
BGP peer reachability can be through OSPF, EIGRP, a connected interface, static routes, or a loopback. iBGP or eBGP can be used for peering with external routers. The BGP route attributes from the external router are preserved since MP-BGP is used for distributing the external routes in the fabric. BGP enables IPv4 and/or IPv6 address families for the VRF associated with an l3extOut. The address family to enable on a switch is determined by the IP address type defined in bgpPeerP policies for the l3extOut. The policy is optional; if not defined, the default will be used. Policies can be defined for a tenant and used by a VRF that is referenced by name.

You must define at least one peer policy to enable the protocol on each border leaf (BL) switch. A peer policy can be defined in two places:

- Under l3extRsPathL3OutAtt—a physical interface is used as the source interface.
- Under l3extLNodeP—a loopback interface is used as the source interface.

**OSPF**

Various host types require OSPF to enable connectivity and provide redundancy. These include mainframe devices, external pods and service nodes that use the ACI fabric as a Layer 3 transit within the fabric and to the WAN. Such external devices peer with the fabric through a nonborder leaf switch running OSPF. Configure the OSPF area as an NSSA (stub) area to enable it to receive a default route and not participate in full-area routing. Typically, existing routing deployments avoid configuration changes, so a stub area configuration is not mandated.

You enable OSPF by configuring an ospfExtP managed object under an l3extOut. OSPF IP address family versions configured on the BL switch are determined by the address family that is configured in the OSPF interface IP address.

---

**Note**

Although the l3extOut policy contains the routing protocol (for example, OSPF with its related VRF and area ID), the Layer 3 external interface profile contains the necessary OSPF interface details. Both are needed to enable OSPF.

You configure OSPF policies at the VRF level by using the fVRSCtxToOspfCtxPol relation, which you can configure per address family. If you do not configure it, default parameters are used.

You configure the OSPF area in the ospfExtP managed object, which also exposes IPv6 the required area properties.

### Scope and Aggregate Controls for Subnets

The following section describes some scope and aggregate options available when creating a subnet:

**Export Route Control Subnet**—The control advertises specific transit routes out of the fabric. This is for transit routes only, and it does not control the internal routes or default gateways that are configured on a bridge domain (BD).

**Import Route Control Subnet**—This control allows routes to be advertised into the fabric with Border Gateway Protocol (BGP) when Import Route Control Enforcement is configured (only supported for BGP).

**External Subnets for the External EPG (also called Security Import Subnet)**—This option does not control the movement of routing information into or out of the fabric. If you want traffic to flow from one external EPG to another external EPG or to an internal EPG, the subnet must be marked with this control. If you do not mark the subnet with this control, then routes learned from one EPG are advertised to the other external
EPG, but packets are dropped in the fabric. The drops occur because the APIC operates in a whitelist model where the default behavior is to drop all data plane traffic between EPGs, unless it is explicitly permitted by a contract. The whitelist model applies to external EPGs and application EPGs. When using security policies that have this option configured, you must configure a contract and a security prefix.

Shared Route Control Subnet—Subnets that are learned from shared L3Outs in inter-VRF leaking must be marked with this control before being advertised to other VRFs. Starting with APIC release 2.2(2e), shared L3Outs in different VRFs can communicate with each other using a contract. For more about communication between shared L3Outs in different VRFs, see the Cisco APIC Layer 3 Networking Configuration Guide.

Shared Security Import Subnet—This control is the same as External Subnets for the External EPG for Shared L3Out learned routes. If you want traffic to flow from one external EPG to another external EPG or to another internal EPG, the subnet must be marked with this control. If you do not mark the subnet with this control, then routes learned from one EPG are advertised to the other external EPG, but packets are dropped in the fabric. When using security policies that have this option configured, you must configure a contract and a security prefix.

Aggregate Export, Aggregate Import, and Aggregate Shared Routes—This option adds 32 in front of the 0.0.0.0/0 prefix. Currently, you can only aggregate the 0.0.0.0/0 prefix for the import/export route control subnet. If the 0.0.0.0/0 prefix is aggregated, no route control profile can be applied to the 0.0.0.0/0 network.

Aggregate Shared Route—This option is available for any prefix that is marked as Shared Route Control Subnet.

Route Control Profile—The ACI fabric also supports the route-map set clauses for the routes that are advertised into and out of the fabric. The route-map set rules are configured with the Route Control Profile policies and the Action Rule Profiles.

<table>
<thead>
<tr>
<th>Property</th>
<th>OSPF</th>
<th>EIGRP</th>
<th>BGP</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Community</td>
<td>X</td>
<td>X</td>
<td>Yes</td>
<td>Supports regular and extended communities.</td>
</tr>
<tr>
<td>Route Tag</td>
<td>Yes</td>
<td>Yes</td>
<td>X</td>
<td>Supported only for BD subnets. Transit prefixes are always assigned the tag 4294967295.</td>
</tr>
<tr>
<td>Preference</td>
<td>X</td>
<td>X</td>
<td>Yes</td>
<td>Sets BGP local preference.</td>
</tr>
<tr>
<td>Metric</td>
<td>Yes</td>
<td>X</td>
<td>Yes</td>
<td>Sets MED for BGP and changes the metric for EIGRP, but you cannot specify the EIGRP composite metric.</td>
</tr>
<tr>
<td>Metric Type</td>
<td>Yes</td>
<td>X</td>
<td>X</td>
<td>OSPF Type-1 and OSPF Type-2.</td>
</tr>
</tbody>
</table>
Route Control Profile Policies

The ACI fabric also supports the route-map set clauses for the routes that are advertised into and out of the fabric. The route-map set rules are configured with the Route Control Profile policies and the Action Rule Profiles.

ACI supports the following set options:

Table 7: Action Rule Profile Properties (route-map set clauses)

<table>
<thead>
<tr>
<th>Property</th>
<th>OSPF</th>
<th>EIGRP</th>
<th>BGP</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Community</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Supports regular and extended communities.</td>
</tr>
<tr>
<td>Set Additional Community</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Supports regular and extended communities.</td>
</tr>
<tr>
<td>Route Tag</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Supported only for BD subnets. Transit prefixes are always assigned the tag 4294967295.</td>
</tr>
<tr>
<td>Preference</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Sets BGP local preference.</td>
</tr>
<tr>
<td>Metric</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Sets MED for BGP. Will change the metric for EIGRP but you cannot specify the EIGRP composite metric.</td>
</tr>
<tr>
<td>Metric Type</td>
<td>Yes</td>
<td></td>
<td></td>
<td>OSPF Type-1 and OSPF Type-2.</td>
</tr>
</tbody>
</table>

The Route Profile Policies are created under the Layer 3 Outside connection. A Route Control Policy can be referenced by the following objects:

- Tenant BD Subnet
- Tenant BD
- External EPG
- External EPG import/export subnet

Here is an example of using Import Route Control for BGP and setting the local preference for an external route learned from two different Layer 3 Outsides. The Layer 3 Outside connection for the external connection to AS300 is configured with the Import Route Control enforcement. An action rule profile is configured to set the local preference to 200 in the Action Rule Profile for Local Preference window.
The Layer 3 Outside connection External EPG is configured with a 0.0.0.0/0 import aggregate policy to allow all the routes. This is necessary because the import route control is enforced but any prefixes should not be blocked. The import route control is enforced to allow setting the local preference. Another import subnet 151.0.1.0/24 is added with a Route Profile that references the Action Rule Profile in the External EPG settings for Route Control Profile window.

Use the `show ip bgp vrf overlay-1` command to display the MP-BGP table. The MP-BGP table on the spine displays the prefix 151.0.1.0/24 with local preference 200 and a next hop of the border leaf for the BGP 300 Layer 3 Outside connection.

There are two special route control profiles—default-import and default-export. If the user configures using the names default-import and default-export, then the route control profile is automatically applied at the Layer3 outside level for both import and export. The default-import and default-export route control profiles cannot be configured using the 0.0.0.0/0 aggregate.

A route control profile is applied in the following sequential order for fabric routes:

1. Tenant BD subnet
2. Tenant BD
3. Layer3 outside

The route control profile is applied in the following sequential order for transit routes:

1. External EPG prefix
2. External EPG
3. Layer3 outside

### Security Import Policies

The policies discussed in the documentation have dealt with the exchange of the routing information into and out of the ACI fabric and the methods that are used to control and tag the routes. The fabric operates in a whitelist model in which the default behavior is to drop all dataplane traffic between the endpoint groups unless it is explicitly permitted by a contract. This whitelist model applies to the external EPGs and the tenant EPGs.

There are some differences in how the security policies are configured and how they are implemented for the transit traffic compared to the tenant traffic.

**Transit Security Policies**

- Uses prefix filtering.
- Starting with Release 2.0(1m), support for Ethertype, protocol, L4 port, and TCP flag filters is available.
- Implemented with the security import subnets (prefixes) and the contracts that are configured under the external EPG.

**Tenant EPG Security Policies**

- Do not use prefix filtering.
- Support Ethertype, protocol, L4 port, and TCP flag filters.
- Supported for tenant EPGs ➔ EPGs and tenant EPGs ➔ External EPGs.
If there are no contracts between the external prefix-based EPGs, the traffic is dropped. To allow traffic between two external EPGs, you must configure a contract and a security prefix. As only prefix filtering is supported, the default filter can be used in the contract.

**External L3Out Connection Contracts**

The union of prefixes for L3Out connections is programmed on all the leaf nodes where the L3Out connections are deployed. When more than two L3Out connections are deployed, the use of the aggregate rule 0.0.0.0/0 can allow traffic to flow between L3Out connections that do not have a contract.

You configure the provider and consumer contract associations and the security import subnets in the L3Out Instance Profile (instP).

When security import subnets are configured and the aggregate rule, 0.0.0.0/0, is supported, the security import subnets follow the ACL type rules. The security import subnet rule 10.0.0.0/8 matches all the addresses from 10.0.0.0 to 10.255.255.255. It is not required to configure an exact prefix match for the prefixes to be permitted by the route control subnets.

Be careful when configuring the security import subnets if more than two L3Out connections are configured in the same VRF, due to the union of the rules.

Transit traffic flowing into and out of the same L3Out is dropped by policies when configured with the 0.0.0.0/0 security import subnet. This behavior is true for dynamic or static routing. To prevent this behavior, define more specific subnets.

---

**Configuring Transit Routing**

**Transit Routing Overview**

This topic provides a typical example of how to configure Transit Routing when using Cisco APIC. The examples in this chapter use the following topology:
In the examples in this chapter, the Cisco ACI fabric has 2 leaf switches and two spine switches, that are controlled by an APIC cluster. The border leaf switches 101 and 102 have L3Outs on them providing connections to two routers and thus to the Internet. The goal of this example is to enable traffic to flow from EP 1 to EP 2 on the Internet into and out of the fabric through the two L3Outs.

In this example, the tenant that is associated with both L3Outs is \( t_1 \), with VRF \( v_1 \).

Before configuring the L3Outs, configure the nodes, ports, functional profiles, AEPs, and a Layer 3 domain. You must also configure the spine switches 104 and 105 as BGP route reflectors.

Configuring transit routing includes defining the following components:

1. Tenant and VRF
2. Node and interface on leaf 101 and leaf 102
3. Primary routing protocol on each L3Out (used to exchange routes between border leaf switch and external routers; in this example, BGP)
4. Connectivity routing protocol on each L3Out (provides reachability information for the primary protocol; in this example, OSPF)
5. Two external EPGs
6. One route map
7. At least one filter and one contract
8. Associate the contract with the external EPGs
For transit routing cautions and guidelines, see Guidelines for Transit Routing, on page 309.

The following table lists the names that are used in the examples in this chapter:

<table>
<thead>
<tr>
<th>Property</th>
<th>Names for L3Out1 on Node 101</th>
<th>Names for L3Out2 on Node 102</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenant</td>
<td>t1</td>
<td>t1</td>
</tr>
<tr>
<td>VRF</td>
<td>v1</td>
<td>v1</td>
</tr>
<tr>
<td>Node</td>
<td>nodep1 with router ID 11.11.11.103</td>
<td>nodep2 with router ID 22.22.22.203</td>
</tr>
<tr>
<td>OSPF Interface</td>
<td>ifp1 at eth/1/3</td>
<td>ifp2 at eth/1/3</td>
</tr>
<tr>
<td>BGP peer address</td>
<td>15.15.15.2/24</td>
<td>25.25.25.2/24</td>
</tr>
<tr>
<td>External EPG</td>
<td>extnw1 at 192.168.1.0/24</td>
<td>extnw2 at 192.168.2.0/24</td>
</tr>
<tr>
<td>Route map</td>
<td>rp1 with ctx1 and route destination 192.168.1.0/24</td>
<td>rp2 with ctx2 and route destination 192.168.2.0/24</td>
</tr>
<tr>
<td>Filter</td>
<td>http-filter</td>
<td>http-filter</td>
</tr>
<tr>
<td>Contract</td>
<td>httpCtrct provided by extnw1</td>
<td>httpCtrct consumed by extnw2</td>
</tr>
</tbody>
</table>

**Configuring Transit Routing Using the REST API**

These steps describe how to configure transit routing for a tenant. This example deploys two L3Outs, in one VRF, on two border leaf switches, that are each connected to a separate router.

**Before you begin**

- Configure the node, port, functional profile, AEP, and Layer 3 domain.
- Create the external routed domain and associate it to the interface for the L3Out.
- Configure a BGP route reflector policy to propagate the routes within the fabric.

For an example of the XML for these prerequisites, see REST API Example: L3Out Prerequisites, on page 31.

**Procedure**

**Step 1**

Configure the tenant and VRF.

This example configures tenant t1 and VRF v1. The VRF is not yet deployed.

**Example:**

```xml
<fvTenant name="t1">
  <fvCtx name="v1"/>
</fvTenant>
```
Step 2  Configure the nodes and interfaces.

This example configures two L3Outs for the tenant t1 and VRF v1, on two border leaf switches. The VRF has a Layer 3 domain, dom1.

- The first L3Out is on node 101, which is named nodep1. Node 101 is configured with router ID 11.11.11.103. It has a routed interface ifp1 at eth1/3, with the IP address 12.12.12.3/24.

- The second L3Out is on node 102, which is named nodep2. Node 102 is configured with router ID 22.22.22.203. It has a routed interface ifp2 at eth1/3, with the IP address, 23.23.23.1/24.

Example:

```xml
<l3extOut name="l3out1">
  <l3extRsEctx tnFvCtxName="v1"/>
  <l3extLNodeP name="nodep1">
    <l3extRsNodeL3OutAtt rtrId="11.11.11.103" tDn="topology/pod-1/node-101"/>
    <l3extLIfP name="ifp1"/>
    <l3extRsPathL3OutAtt addr="12.12.12.3/24" ifInstT="l3-port" tDn="topology/pod-1/paths-101/pathep-[eth1/3]"/>
  </l3extLNodeP>
  <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
</l3extOut>

<l3extOut name="l3out2">
  <l3extRsEctx tnFvCtxName="v1"/>
  <l3extLNodeP name="nodep2">
    <l3extRsNodeL3OutAtt rtrId="22.22.22.203" tDn="topology/pod-1/node-102"/>
    <l3extLIfP name="ifp2"/>
    <l3extRsPathL3OutAtt addr="23.23.23.1/24" ifInstT="l3-port" tDn="topology/pod-1/paths-102/pathep-[eth1/3]"/>
  </l3extLNodeP>
  <l3extRsL3DomAtt tDn="uni/l3dom-dom1"/>
</l3extOut>
```

Step 3  Configure the routing protocol for both border leaf switches.

This example configures BGP as the primary routing protocol for both the border leaf switches, both with ASN 100. It also configures Node 101 with BGP peer 15.15.15.2 and node 102 with BGP peer 25.25.25.2.

Example:

```xml
<l3extOut name="l3out1">
  <l3extLNodeP name="nodep1">
    <bgpPeerP addr="15.15.15.2/24" bgpAsP asn="100"/>
  </l3extLNodeP>
</l3extOut>

<l3extOut name="l3out2">
  <l3extLNodeP name="nodep2">
    <bgpPeerP addr="25.25.25.2/24" bgpAsP asn="100"/>
  </l3extLNodeP>
</l3extOut>
```

Step 4  Configure a connectivity routing protocol.

This example configures OSPF as the communication protocol, for both L3Outs, with regular area ID 0.0.0.0.
Example:

```xml
<l3extOut name="l3out1">
  <ospfExtP areaId="0.0.0.0" areaType="regular"/>
  <l3extLNodeP name="nodep1">
    <l3extLIfP name="ifp1">
      <ospfIfP/>
    </l3extLIfP>
  </l3extLNodeP>
</l3extOut>
<l3extOut name="l3out2">
  <ospfExtP areaId="0.0.0.0" areaType="regular"/>
  <l3extLNodeP name="nodep2">
    <l3extLIfP name="ifp2">
      <ospfIfP/>
    </l3extLIfP>
  </l3extLNodeP>
</l3extOut>
</l3extOut>
```

**Step 5** Configure the external EPGs.

This example configures the network 192.168.1.0/24 as external network `extnw1` on node 101 and 192.168.2.0/24 as external network `extnw2` on node 102. It also associates the external EPGs with the route control profiles `rp1` and `rp2`.

Example:

```xml
<l3extInstP name="extnw1">
  <l3extSubnet ip="192.168.1.0/24" scope="import-security"/>
  <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp1"/>
</l3extInstP>
<l3extInstP name="extnw2">
  <l3extSubnet ip="192.168.2.0/24" scope="import-security"/>
  <l3extRsInstPToProfile direction="export" tnRtctrlProfileName="rp2"/>
</l3extInstP>
</l3extOut>
```

**Step 6** Optional. Configure a route map.

This example configures a route map for each BGP peer in the inbound and outbound directions. For `l3out1`, the route map `rp1` is applied for routes that match an import destination of 192.168.1.0/24 and the route map `rp2` is applied for routes that match an export destination of 192.168.2.0/24. For `l3out2`, the direction of the route maps is reversed.

Example:

```xml
<fvTenant name="t1">
  <rtctrlSubjP name="match-rule1">
    <rtctrlMatchRtDest ip="192.168.1.0/24"/>
  </rtctrlSubjP>
  <rtctrlSubjP name="match-rule2">
    <rtctrlMatchRtDest ip="192.168.2.0/24"/>
  </rtctrlSubjP>
  <l3extOut name="l3out1">
    <rtctrlCtxP name="ctxp1" action="permit" order="0">
      <rtctrlRsCtxPToSubjP tnRtctrlSubjPName="match-rule1"/>
    </rtctrlCtxP>
    <rtctrlCtxP name="ctxp2" action="permit" order="0"/>
  </l3extOut>
  <l3extOut name="l3out2">
    <rtctrlCtxP name="ctxp1" action="permit" order="0"/>
    <rtctrlCtxP name="ctxp2" action="permit" order="0"/>
  </l3extOut>
</fvTenant>
```
Step 7  Create the filter and contract to enable the EPGs to communicate.

This example configures the filter `http-filter` and the contract `httpCtrct`. The external EPGs and the application EPGs are already associated with the contract `httpCtrct` as providers and consumers respectively.

**Example:**

```
<vzFilter name="http-filter">
  <vzEntry name="http-e" etherT="ip" prot="tcp"/>
</vzFilter>
<vzBrCP name="httpCtrct" scope="context">
  <vzSubj name="subj1">
    <vzRsSubjFiltAtt tnVzFilterName="http-filter"/>
  </vzSubj>
</vzBrCP>
```

Step 8  Associate the external EPGs with the contract.

This example associates the external EPG `extnw1` as provider and external EPG `extnw2` as consumer of the contract `httpCtrct`.

```
<l3extOut name="l3out1">
  <l3extInstP name="extnw1">
    <fvRsProv tnVzBrCPName="httpCtrct"/>
  </l3extInstP>
</l3extOut>
<l3extOut name="l3out2">
  <l3extInstP name="extnw2">
    <fvRsCons tnVzBrCPName="httpCtrct"/>
  </l3extInstP>
</l3extOut>
```

---

**REST API Example: Transit Routing**

The following example configures two L3Outs on two border leaf switches, using the REST API.
Configure Transit Routing Using the NX-OS Style CLI

These steps describe how to configure transit routing for a tenant. This example deploys two L3Outs, in one VRF, on two border leaf switches, that are each connected to separate routers.

Before you begin

- Configure the node, port, functional profile, AEP, and Layer 3 domain.
- Configure a VLAN domain using the `vlan-domain` domain and `vlan vlan-range` commands.
- Configure a BGP route reflector policy to propagate the routed within the fabric.

For an example of the commands for these prerequisites, see NX-OS Style CLI Example: L3Out Prerequisites, on page 37.

Procedure

Step 1  Configure the tenant and VRF.

This example configures tenant t1 with VRF v1. The VRF is not yet deployed.

Example:

```
apic1# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
```

Step 2  Configure the nodes and interfaces.

This example configures two L3Outs for the tenant t1, on two border leaf switches:
Configure Transit Routing Using the NX-OS Style CLI

- The first L3Out is on node 101, which is named `nodep1`. Node 101 is configured with router ID 11.11.11.103. It has a routed interface `ifp1` at `eth1/3`, with the IP address `12.12.12.3/24`.

- The second L3Out is on node 102, which is named `nodep2`. Node 102 is configured with router ID 22.22.22.203. It has a routed interface `ifp2` at `eth1/3`, with the IP address, `23.23.23.1/24`.

Example:

```
apic1(config)# leaf 101
napic1(config-leaf)# vrf context tenant t1 vrf v1
napic1(config-leaf-vrf)# router-id 11.11.11.103
napic1(config-leaf-vrf)# exit
napic1(config-leaf)# interface ethernet 1/3
napic1(config-leaf-if)# vlan-domain member dom1
napic1(config-leaf-if)# no switchport
napic1(config-leaf-if)# vrf member tenant t1 vrf v1
napic1(config-leaf-if)# ip address 12.12.12.3/24
napic1(config-leaf-if)# exit
napic1(config-leaf)# exit
napic1(config)# leaf 102
napic1(config-leaf)# vrf context tenant t1 vrf v1
napic1(config-leaf-vrf)# router-id 22.22.22.203
napic1(config-leaf-vrf)# exit
napic1(config-leaf)# interface ethernet 1/3
napic1(config-leaf-if)# vlan-domain member dom1
napic1(config-leaf-if)# no switchport
napic1(config-leaf-if)# vrf member tenant t1 vrf v1
napic1(config-leaf-if)# ip address 23.23.23.3/24
napic1(config-leaf-if)# exit
napic1(config-leaf)# exit
```

**Step 3** Configure the routing protocol for both leaf switches.

This example configures BGP as the primary routing protocol for both the border leaf switches, both with ASN 100. It also configures Node 101 with BGP peer `15.15.15.2` and node 102 with BGP peer `25.25.25.2`.

**Example:**

```
apic1(config)# leaf 101
napic1(config-leaf)# router bgp 100
napic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
napic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
napic1(config-leaf-bgp-vrf-neighbor)# exit
napic1(config-leaf-bgp-vrf)# exit
napic1(config-leaf-bgp)# exit
napic1(config-leaf)# exit
napic1(config)# leaf 102
napic1(config-leaf)# router bgp 100
napic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
napic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2
napic1(config-leaf-bgp-vrf-neighbor)# exit
napic1(config-leaf-bgp-vrf)# exit
napic1(config-leaf-bgp)# exit
napic1(config-leaf)# exit
```

**Step 4** Configure a connectivity routing protocol.

This example configures OSPF as the communication protocol, for both L3Outs, with regular area ID `0.0.0.0`.

**Example:**

```
apic1(config)# leaf 101
napic1(config-leaf)# router ospf default
napic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
```
Step 5 Configure the external EPGs.

This example configures the network 192.168.1.0/24 as external network extnw1 on node 101 and the network 192.168.2.0/24 as external network extnw2 on node 102.

Example:

```plaintext
apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.1.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw2
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# exit
```

Step 6 Optional. Configure the route maps.

This example configures a route map for each BGP peer in the inbound and outbound directions.

Example:

```plaintext
apic1(config)# leaf 101
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rpl1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rpl2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
```
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

Step 7 Create filters (access lists) and contracts to enable the EPGs to communicate.

Example:
apic1(config)# tenant t1
apic1(config-tenant)# access-list http-filter
apic1(config-tenant-acl)# match ip
apic1(config-tenant-acl)# match tcp dest 80
apic1(config-tenant-acl)# exit
apic1(config-tenant)# contract httpCtrct
apic1(config-tenant-contract)# scope vrf
apic1(config-tenant-contract)# subject subj1
apic1(config-tenant-contract-subj)# access-group http-filter both
apic1(config-tenant-contract-subj)# exit
apic1(config-tenant-contract)# exit
apic1(config-tenant)# exit

Step 8 Configure contracts and associate them with EPGs.

Example:
apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract provider httpCtrct
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# contract consumer httpCtrct
Example: Transit Routing

This example provides a merged configuration for transit routing. The configuration is for a single tenant and VRF, with two L3Outs, on two border leaf switches, that are each connected to separate routers.

```plaintext
apic1(config)# configure
apic1(config)# tenant t1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)#

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 11.11.11.103
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 12.12.12.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# route
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 40.40.40.1
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# router-id 22.22.22.203
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# interface ethernet 1/3
apic1(config-leaf-if)# vlan-domain member dom1
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vrf member tenant t1 vrf v1
apic1(config-leaf-if)# ip address 23.23.23.3/24
apic1(config-leaf-if)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 25.25.25.2/24
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# route
apic1(config-leaf-ospf)# vrf member tenant t1 vrf v1
apic1(config-leaf-ospf-vrf)# area 0.0.0.0 loopback 60.60.60.3
apic1(config-leaf-ospf-vrf)# exit
apic1(config-leaf-ospf)# exit
apic1(config-leaf)# exit
```

Cisco APIC Layer 3 Networking Configuration Guide, Release 4.0(1)
apic1(config)# tenant t1
apic1(config-tenant)# external-l3 epg extnw1
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.1.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# external-l3 epg extnw2
apic1(config-tenant-l3ext-epg)# vrf member v1
apic1(config-tenant-l3ext-epg)# match ip 192.168.2.0/24
apic1(config-tenant-l3ext-epg)# exit
apic1(config-tenant)# exit

apic1(config)# leaf 101
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw1
apic1(config-leaf-vrf)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# external-l3 epg extnw2
apic1(config-leaf-vrf)# exit

apic1(config)# leaf 101
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# exit
apic1(config-leaf)# router bgp 100
apic1(config-leaf-bgp)# vrf member tenant t1 vrf v1
apic1(config-leaf-bgp-vrf)# neighbor 15.15.15.2
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp1 in
apic1(config-leaf-bgp-vrf-neighbor)# route-map rp2 out
apic1(config-leaf-bgp-vrf-neighbor)# exit
apic1(config-leaf-bgp-vrf)# exit
apic1(config-leaf-bgp)# exit
apic1(config-leaf)# exit

apic1(config)# leaf 102
apic1(config-leaf)# template route group match-rule1 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.1.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# template route group match-rule2 tenant t1
apic1(config-route-group)# ip prefix permit 192.168.2.0/24
apic1(config-route-group)# exit
apic1(config-leaf)# vrf context tenant t1 vrf v1
apic1(config-leaf-vrf)# route-map rp1
apic1(config-leaf-vrf-route-map)# match route group match-rule1 order 0
apic1(config-leaf-vrf-route-map)# exit
apic1(config-leaf-vrf)# route-map rp2
apic1(config-leaf-vrf-route-map)# match route group match-rule2 order 0
apic1(config-leaf-vrf-route-map)# exit
Configure Transit Routing Using the GUI

These steps describe how to configure transit routing for a tenant. This example deploys two L3Outs, in one VRF, on two border leaf switches, that are connected to separate routers.

Except for the step to create the tenant and VRF, perform these steps twice, to create the two L3Outs under the same tenant and VRF.

For sample names, see Transit Routing in the ACI Fabric, on page 301.

Before you begin

• Configure the node, port, functional profile, AEP, and Layer 3 domain.

• Create the external routed domain and associate it to the interface for the L3Out.

• Configure a BGP Route Reflector policy to propagate the routes within the fabric.
Procedure

Step 1
To create the tenant and VRF, on the menu bar, choose Tenants > Add Tenant and in the Create Tenant dialog box, perform the following tasks:
   a) In the Name field, enter the tenant name.
   b) In the VRF Name field, enter the VRF name.
   c) Click Submit.

   Note After this step, perform the steps twice to create two L3Outs in the same tenant and VRF for transit routing.

Step 2
To start creating the L3Out, in the Navigation pane, expand Tenant and Networking and perform the following steps:
   a) Right-click External Routed Networks and choose Create Routed Outside.
   b) In the Name field, enter a name for the L3Out.
   c) From the VRF drop-down list, choose the VRF you previously created.
   d) From the External Routed Domain drop-down list, choose the external routed domain that you previously created.
   e) In the area with the routing protocol check boxes, check the desired protocols (BGP, OSPF, or EIGRP).

      For the example in this chapter, choose BGP and OSPF.

      Depending on the protocols you choose, enter the properties that must be set.
   f) Enter the OSPF details, if you enabled OSPF.

      For the example in this chapter, use the OSPF area 0 and type Regular area.
   g) Click the + icon to expand Nodes and Interfaces Protocol Profiles.
   h) In the Name field, enter a name.
   i) Click the + icon to expand Nodes.
   j) From the Node ID field drop-down list, choose the node for the L3Out.
   k) In the Router ID field, enter the router ID (IPv4 or IPv6 address for the router that is connected to the L3Out).
   l) (Optional) You can configure another IP address for a loopback address. Uncheck Use Router ID as Loopback Address, expand Loopback Addresses, enter an IP address, and click Update.
   m) In the Select Node dialog box, click OK.

Step 3
If you enabled BGP, click the + icon to expand BGP Peer Connectivity Profiles and perform the following steps:
   a) In the Peer Address field, enter the BGP peer address.
   b) In the Local-AS Number field, enter the BGP AS number.
   c) Click OK.

Step 4
Click the + icon to expand Interface Profiles (OSPF Interface Profiles if you enabled OSPF), and perform the following actions:
   a) In the Name field, enter a name for the interface profile.
   b) Click Next.
   c) In the Protocol Profiles dialog box, in the OSPF Policy field, choose an OSPF policy.
   d) Click Next.
e) Click the + icon to expand **Routed Interfaces**.

f) In the **Select Routed Interface** dialog box, from the **Node** drop-down list, choose the node.

g) From the **Path** drop-down list, choose the interface path.

h) In the **IPv4 Primary/IPv6 Preferred Address** field, enter the IP address and network mask for the interface.

   **Note** To configure IPv6, you must enter the link-local address in the **Link-local Address** field.

i) Click **OK** in the **Select Routed Interface** dialog box.

j) Click **OK** in the **Create Interface Profile** dialog box.

**Step 5** In the **Create Node Profile** dialog box, click **OK**.

**Step 6** In the **Create Routed Outside** dialog box, click **Next**.

**Step 7** In the **External EPG Networks** tab, click **Create Route Profiles**.

**Step 8** Click the + icon to expand **Route Profiles** and perform the following actions:

a) In the **Name** field, enter the route map name.

b) Choose the **Type**.

   For this example, leave the default, **Match Prefix AND Routing Policy**.

c) Click the + icon to expand **Contexts** and create a route context for the route map.

d) Enter the order and name of the profile context.

e) Choose the **Deny** or **Permit** action to be performed in this context.

f) (Optional) In the **Set Rule** field, choose **Create Set Rules for a Route Map**.

   Enter the name for the set rules, click the objects to be used in the rules, and click **Finish**.

g) In the **Associated Matched Rules** field, click the + icon to create a match rule for the route map.

h) Enter the name for the match rules and enter the **Match Regex Community Terms**, **Match Community Terms**, or **Match Prefix** to match in the rule.

i) Click the rule name and click **Update**.

j) In the **Create Match Rule** dialog box, click **Submit**, and then click **Update**.

k) In the **Create Route Control Context** dialog box, click **OK**.

l) In the **Create Route Map** dialog box, click **OK**.

**Step 9** Click the + icon to expand **External EPG Networks**.

**Step 10** In the **Name** field, enter a name for the external network.

**Step 11** Click the + icon to expand **Subnet**.

**Step 12** In the **Create Subnet** dialog box, perform the following actions:

a) In the **IP address** field, enter the IP address and network mask for the external network.

b) In the **Scope** field, check the appropriate check boxes to control the import and export of prefixes for the L3Out.

   **Note** For more information about the scope options, see the online Help for this **Create Subnet** panel.

c) (Optional) In the **Route Summarization Policy** field, from the drop-down list, choose an existing route summarization policy or create a new one as desired. Also click the check box for **Export Route Control Subnet**.

   The type of route summarization policy depends on the routing protocols that are enabled for the L3Out.

d) Click the + icon to expand **Route Control Profile**.
e) In the **Name** field, choose the route control profile that you previously created from the drop-down list.
f) In the **Direction** field, choose **Route Export Policy**.
g) Click **Update**.
h) In the **Create Subnet** dialog box, click **OK**.
i) (Optional) Repeat to add more subnets.
j) In the **Create External Network** dialog box, click **OK**.

**Step 13**
In the **Create Routed Outside** dialog box, click **Finish**.

**Step 14**
In the **Navigation** pane, under **External Routed Networks**, expand the previously created L3Out and right-click **Route Maps/Profiles**.

**Note**
To set attributes for BGP, OSPF, or EIGRP for received routes, create a default-import route control profile, with the appropriate set actions and no match actions.

**Step 15**
Choose **Create Route Map/Profile**, and in the **Create Route Map/Profile** dialog box, perform the following actions:

a) From the drop-down list on the **Name** field, choose **default-import**.
b) In the **Type** field, click **Match Routing Policy Only**. Click **Submit**.

**Step 16**
(Optional) To enable extra communities to use BGP, using the following steps:

a) Right-click **Set Rules for Route Maps**, and click **Create Set Rules for a Route Map**.
b) In the **Create Set Rules for a Route Map** dialog box, click the **Add Communities** field.
c) Follow the steps to assign multiple BGP communities per route prefix.

**Step 17**
To enable communications between the EPGs consuming the L3Out, create at least one filter and contract, using the following steps:

a) In the **Navigation** pane, under the tenant consuming the L3Out, expand **Contracts**.
b) Right-click **Filters** and choose **Create Filter**.
c) In the **Name** field, enter a filter name.

A filter is essentially an Access Control List (ACL).

d) Click the + icon to expand **Entries**, to add a filter entry.
e) Add the Entry details.

For example, for a simple web filter, set criteria such as the following:

- **EtherType**—IP
- **IP Protocol**—tcp
- **Destination Port Range From**—Unspecified
- **Destination Port Range To** to https

f) Click **Update**.
g) In the **Create Filter** dialog box, click **Submit**.

**Step 18**
To add a contract, use the following steps:

a) Under **Contracts**, right-click **Standard** and choose **Create Contract**.
b) Enter the name of the contract.
c) Click the + icon to expand **Subjects** and add a subject to the contract.
d) Enter a name for the subject.
e) Click the + icon to expand Filters and choose the filter that you previously created, from the drop-down list.
f) Click Update.
g) In the Create Contract Subject dialog box, click OK.
h) In the Create Contract dialog box, click Submit.

Step 19

Associate the EPGs for the L3Out with the contract, with the following steps:

The first L3 external EPG, extnw1, is the provider of the contract and the second L3 external EPG, extnw2, is the consumer.

a) To associate the contract to the L3 external EPG, as the provider, under the tenant, click Networking, expand External Routed Networks, and expand the L3Out.
b) Expand Networks, click the L3 external EPG, and click Contracts.
c) Click the the + icon to expand Provided Contracts.
   For the second L3 external EPG, click the + icon to expand Consumed Contracts.
d) In the Name field, choose the contract that you previously created from the list.
e) Click Update.
f) Click Submit.